École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur

de l'Ecole Nationale Supérieure des Télécommunications

Spécialité : Informatique et réseaux

Abdullatif SHIKFA

Sécurité des Communications Opportunistes

Soutenue le 28 juin 2010 devant le jury composé de

Jon CROWCROFT	Rapporteurs
Bruno MARTIN	
Ernst BIERSACK	Examinateurs
Roberto DI PIETRO	
Refik MOLVA	Directeurs de thèse
Melek ÖNEN	

TELECOM
ParisTech

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thesis

submitted in partial fulfilment of the requirements for the degree of Docteur (PhD) of

Ecole Nationale Supérieure des Télécommunications

Speciality : Computer Science and Networks

Abdullatif SHIKFA

Security in Opportunistic Communications

Defended on June 28, 2010 in front of the jury composed of

Jon CROWCROFT	Reviewers
Bruno MARTIN	
Ernst BIERSACK	Examiners
Roberto DI PIETRO	
Refik MOLVA	Advisors
Melek ÖNEN	

"Over every knowledgeable person is one more knowing."
The Holy Quran – Yusuf/Joseph 12, Verse: 76

"Knowledge is power."
Sir Francis Bacon – Religious Meditations, Of Heresies

"The fool doth think he is wise, but the wise man knows himself to be a fool."
William Shakespeare – As You Like It, Act 5 scene 1

"There are three classes of people: those who see, those who see when they are shown, those who do not see."
Leonardo da Vinci

Acknowledgements

As I do not believe that religion and science are adversaries, and that on the contrary, religion fosters scientific progress and knowledge strengthens faith, I first praise God the All-Knowledgeable for having bestowed on me a little knowledge that enabled me to finish this work.

This thesis represents four years of work, with ups and downs, and I would not have been able to finish it without the moral support of my parents first, who devoted much of their time to provide me with the best education and made me the man that I am today (I acknowledge that I will never be able to repay them this much), and of my wife and three kids who patiently bear with me in the periods of hard work and always fill my heart with joy when I come back home.

I also would like to thank my brother for his encouragements and kind words, and my friends Atef, Emmanuel, Guillaume, Hassan, Jihad, Karim, Mathias, Vivien and many others that I cannot cite and I hope that they will forgive me. It is always comforting to find people with whom you can talk about topics besides networks, computers or mathematics.

Concerning the work itself, I would first and foremost like to thank my advisor, Refik Molva, for his faith in my work and for giving me excellent advice on many issues concerning this work and beyond. Refik taught me to look beyond the technical details of a solution to see the bigger picture, he taught me the art of writing scientific articles, the art of explaining complex concepts with simple words, and many other skills. I still have a lot to learn from him, and even though I am a doctor now, I will always remain his student and he will always be my professor. It was an honor to work alongside him for a while, because he is not only a great scientist but he also has remarkable human qualities.

Then I would like to thank Melek Önen for her great availability and her patience in reading all the (many) versions of my articles and thesis: she always provided constructive criticism to improve my work. Furthermore, she provided me with innovative ideas for my work and I highly appreciated working with such a bright and kind researcher.

I thank the jury members for the precious time they have taken to read the manuscript and for making useful comments to improve it.

I would also like to thank my many office mates, Cédric, Adomas, Chi Anh, Matteo, Andrea, Antonio and very special thanks to Erik for the many interesting discussions both on scientific and more general topics, for trusting my appreciation with respect to some of his new ideas, and for providing useful remarks on my work. We also collaborated together with Refik, Guevara and Anil on an article [BKM$^+$09, BKM$^+$10] which is out of the scope of this thesis and enabled me to explore an interesting and different research topic. More generally, I would like to thank all the faculty members and students of the Network and Security department for the many interesting discussions around a coffee, and for making Eurecom such an exciting place to do research.

I would also like to thank all the partners of the Haggle project: it was really a unique experience for me, it was always a renewed pleasure to meet with all of them and I enjoyed it a lot.

Finally I would like to thank all the Eurecom staff and in particular the secretaries

Christine, Christine and Audrey as well as Gwenaelle, Carine, Laurence and Pascale for making the life of PhD students easier.

Abstract

In this thesis, we investigate security in opportunistic communications which are an exciting new communication paradigm aiming at enabling communication in challenged conditions. The specific characteristics of opportunistic communication require to revisit all security aspects of communication. For instance, nodes' high mobility implies that security solutions should be dynamic and local. The ad-hoc nature of opportunistic networks also calls for self-organized security solutions. Furthermore, delay tolerance, which is one of the main characteristics of opportunistic networks, has a strong impact from a security perspective as it amounts to the infeasibility for a node to contact at any time a centralized distant security server or the destination, hence interactive protocols are infeasible in opportunistic networks.

Moreover, radically new forwarding strategies have been proposed to enable communication in opportunistic networks: departing from traditional network addresses, these enriched forwarding strategies use information such as context of a node or content of a message to take forwarding decisions. These enriched forwarding strategies require a collapsed architecture which raises completely new security issues, as it implies that security solutions should take into consideration the requirements of application and networking at the same time, and cannot secure information at each layer separately. Furthermore, context or content are sensitive information that users might not want to reveal to others in order to preserve their privacy. Preserving user privacy requires a careful handling of these information by assuring the secrecy of private information. Hiding these information through classical encryption mechanisms is incompatible with the enriched strategies, which take forwarding decisions based on these very information and thus require access to them. The conflicting requirements between security and forwarding motivate the need for new security mechanisms that enable computation on encrypted data.

After analyzing the security challenges in opportunistic communications, we propose a complete security framework for context-based communication. This framework features not only data confidentiality and user privacy but also computation assurance, hence it enables privacy-preserving context-based forwarding while being resilient against malicious entities aiming at disrupting or subverting the communication. We also propose a privacy-preserving content-based protocol which relies on multiple encryption layers based on the neighborhood topology, and an associated local and topology-dependent key management solution.

Résumé

Dans cette thèse, nous étudions la sécurité des communications opportunistes. Dans ce nouveau type de communication, la mobilité des noeuds implique que les solutions de sécurité doivent être dynamiques et locales. Par ailleurs l'absence de connectivité bout-en-bout compromet toute solution de sécurité interactive.

En outre, contrairement au routage traditionnel basé sur des adresses, les nouvelles stratégies de transmission opportunistes utilisent des informations comme le contexte d'un

noeud ou le contenu d'un message pour prendre les décisions de transfert. Contexte et contenu sont des informations sensibles que les utilisateurs pourraient ne pas vouloir révéler aux autres afin de préserver leur vie privée, par conséquent, ces informations doivent être manipulées avec soin pour assurer leur confidentialité. Le conflit entre les exigences de sécurité et de routage justifie la recherche de nouveaux mécanismes de sécurité qui permettent certaines opérations sur des données chiffrées.

Après avoir analysé les problèmes de sécurité dans les communications opportunistes, nous proposons une solution de sécurité complète pour la communication basée sur le contexte. Cette solution garantit non seulement la confidentialité des données et le respect de la vie privée des utilisateurs, mais aussi la correction des opérations, ce qui procure une résistance face aux attaques visant à perturber ou à endiguer la communication. Nous proposons aussi un protocole de routage basé sur le contenu qui préserve la vie privée des utilisateurs via un système de chiffrement multicouches, et une solution associée de gestion de clés, qui est locale et dépendante de la topologie.

Contents

Résumé en Français **17**
0.1 Introduction . . . 17
0.2 Routage et Transmission dans les Réseaux Opportunistes . . . 18
0.3 Problématiques de Sécurité dans les Réseaux Opportunistes . . . 20
0.3.1 Coopération . . . 20
0.3.2 Authentification et intégrité . . . 21
0.3.3 Confidentialité et respect de la vie privée . . . 22
0.4 L'architecture Haggle . . . 24
0.4.1 Architecture Globale . . . 24
0.4.2 Le Security Manager . . . 24
0.4.3 Certificats d'Attributs . . . 25
0.5 Problèmes de Sécurité dans les Protocoles Basés sur le Contexte . . . 26
0.5.1 Transmission basée sur le contexte . . . 26
0.5.2 Problématiques de sécurité . . . 28
0.5.2.1 Confidentialité du contenu . . . 28
0.5.2.2 Vie privée des utilisateurs . . . 29
0.5.2.3 Assurance de calcul . . . 29
0.5.3 Bilan et vue d'ensemble . . . 30
0.6 Confidentialité du contenu dans les Protocoles Basés sur le Contexte . . . 31
0.6.1 Chiffrement basé sur des identités multiples . . . 32
0.6.2 Application au problème de confidentialité du contenu . . . 33
0.6.2.1 Phase d'organisation . . . 33
0.6.2.2 Phase d'exécution . . . 33
0.6.3 Evaluation de Sécurité . . . 33
0.7 Respect de la Vie Privée des Utilisateurs dans les Protocoles Basés sur le Contexte . . . 34
0.7.1 Chiffrement à clef publique avec recherche de mots-clefs (PEKS) . . 35
0.7.2 Nouvelle instanciation du protocole PEKS pour garantir le respect de la vie privée des utilisateurs . . . 36
0.7.2.1 Phase d'organisation . . . 36
0.7.2.2 Phase d'exécution . . . 36
0.7.3 Evaluation . . . 37

0.8 Exactitude des Calculs Chiffrés dans les Protocoles Basés sur le Contexte . 37
0.8.1 Première approche 38
0.8.2 Solution complète avec les filtres de Bloom 39
0.8.3 Evaluation de sécurité 40
0.8.3.1 Respect de la vie privée 40
0.8.3.2 Assurance de calcul 41
0.9 Evaluation Globale des Mécanismes de Sécurité Proposés pour les Protocoles Basés sur le Contexte 42
0.9.1 Etude de performance 42
0.9.1.1 Stockage 42
0.9.1.2 Impact sur la communication 42
0.9.1.3 Charge de calcul 43
0.9.1.4 Exemple numérique 43
0.9.2 Extensions 43
0.9.2.1 Révocation 44
0.9.2.2 Protection envers un TTP malveillant 44
0.9.2.3 Hiérarchiser les attributs 44
0.9.3 Conclusion 44
0.10 Confidentialité et Respect de la vie Privée dans les Protocoles Basés sur le Contenu 45
0.10.1 Modèle de communication 45
0.10.2 Problématique de sécurité 46
0.10.3 Protocole de routage basé sur le contenu sécurisé 47
0.10.3.1 Chiffrement commutatif multiple 47
0.10.3.2 Choix du système cryptographique commutatif 48
0.10.3.3 Description du protocole 48
0.10.3.4 Traitement des filtres de souscription 48
0.10.3.5 Diffusion des notifications d'évènements 49
0.10.4 Evaluation 49
0.11 Associations de Sécurité dans les Protocoles Basés sur le Contenu 50
0.11.1 Analyse des besoins de sécurité 50
0.11.2 Protocole d'établissement de clef et de découverte sécurisée du voisinage 51
0.11.2.1 Phase d'organisation 51
0.11.2.2 Phase d'exécution 51
0.11.3 Evaluation 52
0.11.4 Bilan 53
Conclusion 54

1 Introduction 55
1.1 Opportunistic Networks 55
1.2 Security in Opportunistic Networks 58
1.3 Contributions 60

I Opportunistic Communications: Forwarding Strategies and Security Challenges 63

2 Forwarding in Opportunistic Networks 65
2.1 Opportunistic Communication: A New Communication Paradigm 65
2.1.1 Characteristics 65
2.1.1.1 Mobility 66
2.1.1.2 Delay Tolerance 67
2.1.1.3 Disseminational Communication 68
2.1.2 Application Scenarios 70
2.1.2.1 Wildlife Monitoring 71
2.1.2.2 Opportunistic Networks for Developing Areas 72
2.1.2.3 Pocket-Switched Networks 73
2.2 Opportunistic Forwarding Protocols 73
2.2.1 Epidemic Forwarding 74
2.2.2 Protocols Based on a Simple Estimation of the Forwarding Likelihood 77
2.2.3 Enriched Forwarding Proposals 79
2.3 Classification of Opportunistic Forwarding Strategies 82
2.4 Summary 87

3 Security Issues in Opportunistic Networks 89
3.1 Cooperation Enforcement 90
3.2 Trust Establishment 93
3.3 Authentication and Integrity 96
3.4 Confidentiality and Privacy 99
3.5 Summary 103

4 Haggle Architecture 105
4.1 Overview of the Haggle Node Architecture 105
4.1.1 Architectural Invariants 105
4.1.2 Main Components of the Haggle Architecture 106
4.1.3 Data Objects 108
4.1.4 Haggle Managers 108
4.2 Security in the Haggle Node Architecture 109
4.2.1 Design Considerations 109
4.2.2 Attribute Certificates 111
4.2.2.1 Structure of Attribute Certificates in HAGGLE: HaggleCertificates 112
4.2.2.2 Key Management for HaggleCertificates 113
4.2.2.3 Storage of HaggleCertificates 113
4.2.3 Implementation of the Security Manager 114
4.2.3.1 Basic Functionalities 114
4.2.3.2 Management of Certificates 115

4.2.3.3 Securing Node Description 117
4.2.4 A practical Scenario 118
4.3 Conclusion 120

II Secure Context-Based Forwarding 121

5 Security Issues in Context-Based Forwarding 123
5.1 Introduction 123
5.2 Context-Based Forwarding Settings 124
5.2.1 Network Settings 125
5.2.2 Message Format 126
5.2.3 Forwarding Model 127
5.2.4 Examples 128
5.2.4.1 Scenario (I) 129
5.2.4.2 Scenario (II) 129
5.3 Security Requirements 130
5.3.1 Payload Confidentiality 132
5.3.2 User Privacy 133
5.3.3 Computation Assurance 135
5.3.4 Trusting the Attributes 136
5.4 Overview of the Proposed Security Solution 137

6 Solution for Payload Confidentiality 139
6.1 Related Work 139
6.2 Cryptographical Background 140
6.2.1 Cryptographic Hash Functions 140
6.2.2 Bilinear Pairings 141
6.2.3 Bilinear Diffie-Hellman Assumption 142
6.2.4 `IND-CPA` Security 142
6.3 Identity-Based Encryption 143
6.3.1 Definition 143
6.3.2 Construction 143
6.4 Multiple Identity-Based Encryption 145
6.4.1 Definition 145
6.4.2 Construction 146
6.4.3 The `IND-MID-CPA` Security Model 147
6.5 Application of the $\mathcal{MIBE}$ Scheme to Payload Confidentiality 149
6.5.1 Setup Phase 149
6.5.2 Runtime Phase 150
6.6 Security Evaluation 150
6.6.1 `BasicPub` 151
6.6.2 Reduction of `IND-MID-CPA` to `IND-CPA` Security 151

6.6.3 Semantic Security of the $\mathcal{MIBE}$ Scheme . . . 155
6.7 Summary . . . 155

7 **Proposal for User Privacy** **157**
7.1 Introduction . . . 157
7.2 Basic Approach Based on Hash Functions . . . 158
7.2.1 Solution Sketch . . . 158
7.2.2 Security Concerns . . . 158
7.2.3 Preventing Dictionary Attacks . . . 159
7.3 Public key Encryption with Keyword Search (PEKS) . . . 160
7.3.1 Definition . . . 160
7.3.2 Construction . . . 161
7.4 Searchable Encryption for User Privacy in MobiOpps . . . 163
7.4.1 Setup Phase . . . 164
7.4.2 Runtime Phase . . . 166
7.5 Evaluation . . . 166
7.6 Summary . . . 167

8 **Computation Assurance Proposal** **169**
8.1 Introduction . . . 169
8.2 Basic Idea . . . 170
8.2.1 Commitment Schemes . . . 170
8.2.2 A Fundamental Building Block for Computation Assurance . . . 171
8.2.3 Analysis . . . 172
8.3 Efficient Solution . . . 173
8.3.1 Counting Bloom Filters . . . 173
8.3.2 Combining PEKS and Counting Bloom Filters . . . 175
8.4 Security Assessment . . . 177
8.4.1 User Privacy . . . 178
8.4.2 Computation Assurance . . . 179
8.4.2.1 Attacker Model . . . 180
8.4.2.2 Probabilistic Modeling of Counting Bloom Filters . . . 181
8.4.2.3 Probability of Success of an Attacker . . . 191
8.4.3 Tuning the global Parameters . . . 194
8.5 Conclusion . . . 197

9 **Global Evaluation** **199**
9.1 Security Evaluation . . . 200
9.1.1 Key Management . . . 200
9.1.2 Operational Security . . . 201
9.2 Performance Evaluation . . . 203
9.2.1 Storage . . . 203
9.2.2 Communication Overhead . . . 203

9.2.3 Computation Overhead 204
9.2.4 Typical Figures 204
9.3 Extensions 206
9.3.1 Revocation 206
9.3.2 Protection against malicious TTP 208
9.3.3 Weighting the attributes 210
9.4 Conclusion 210

III Privacy-Preserving Content-Based Routing in Mobile Opportunistic Networks 213

10 Privacy in Content-Based Communication 215
10.1 Introduction 215
10.2 Reference Model 216
10.2.1 Content-Based Publish/Subscribe (CBPS) 217
10.2.2 Content-based Communication in Mobile Opportunistic Networks . . 217
10.2.3 Messages in Content-Based Communication 219
10.3 Privacy Issues in Content-Based Communication 220
10.3.1 Privacy, Confidentiality, and Routing 221
10.3.2 Threat Model and Security Assumptions 222
10.4 Privacy-Preserving Routing with Multiple Layer Encryption 223
10.4.1 Multiple layer commutative encryption (MLCE) 223
10.4.2 Security Primitives 224
10.5 Proposed solution 225
10.5.1 Choice of the Commutative Cryptosystem 225
10.5.2 The Pohlig-Hellman Cryptosystem 227
10.5.3 Propagation of Subscription Filters and Building of Routing Tables . 228
10.5.3.1 ENCRYPT_FILTER 228
10.5.3.2 SECURE_TABLE_BUILDING 228
10.5.4 Content Distribution and Secure Look-up 230
10.5.4.1 ENCRYPT_NOTIFICATION 230
10.5.4.2 SECURE_LOOK_UP 230
10.6 An Example 232
10.6.1 Propagation of Subscription Filters 232
10.6.2 Dissemination of Event Notifications 233
10.7 Analysis 234
10.7.1 Security Evaluation 235
10.7.2 Performance 236
10.7.3 Trade-off Between Performance and Security 236
10.8 Related Work 237
10.9 Conclusion 239

11 Bootstrapping Security Associations in Opportunistic Networks 243
11.1 Introduction . . . 243
11.2 Problem Statement . . . 244
11.2.1 Key Management Requirements . . . 244
11.2.1.1 Requirements akin to Opportunistic Networks . . . 244
11.2.1.2 Specific Requirements of the MLCE Solution . . . 244
11.2.2 Threats . . . 245
11.2.2.1 Generic Attacks . . . 245
11.2.2.2 Sybil Attack . . . 245
11.3 Proposed Solution . . . 246
11.3.1 Setup Phase . . . 246
11.3.2 Bootstrapping Local Security Associations . . . 248
11.3.2.1 Initiation of Security Association Request . . . 248
11.3.2.2 Processing of Security Association Requests . . . 250
11.3.2.3 Security Association Reply . . . 251
11.3.2.4 Key Computation . . . 251
11.3.3 Summary . . . 252
11.4 Evaluation . . . 252
11.4.1 Evaluation of the Setup Phase . . . 252
11.4.2 Analysis of the security association mechanism . . . 253
11.4.2.1 Protection against Passive Attackers . . . 253
11.4.2.2 Protection against Man-In-the-Middle Attack . . . 253
11.4.2.3 Incidence of Replay Attacks . . . 253
11.4.2.4 On the Modification of the STS Protocol . . . 253
11.4.3 Evaluation of the Neighborhood Discovery Mechanism . . . 254
11.4.4 Performance Evaluation . . . 255
11.5 Related work . . . 256
11.5.1 Key Management in Ad-Hoc Networks . . . 256
11.5.2 Secure Neighborhood Discovery . . . 257
11.6 Conclusion . . . 257

Conclusion 259

Bibliography 265

Résumé en Français

0.1 Introduction

Imaginons un nouveau type de communication, dans lequel l'émetteur n'aurait pas besoin de spécifier le destinataire du message explicitement (via une adresse par exemple). Le destinataire du message serait au contraire implicitement défini via le contenu du message, et il en irait de même pour le routage. L'architecture réseau classique divisée en couches devrait alors être remplacée par une architecture condensée.

Ce type de communication est le fondement des communications opportunistes, et ce n'est pas une utopie mais d'ors et déjà une réalité, un domaine de recherche important au sujet duquel des centaines d'articles scientifiques ont été publiés, et qui est le sujet principal de nombreux projets de recherche récents (dont le projet Haggle) et d'un groupe spécial de l'Internet Research Task Force appelé le Delay-Tolerant Networking Research Group. La raison pour laquelle ce nouveau paradigme est l'objet d'autant d'attentions est le nombre important d'applications qui peuvent en découler. Les communications opportunistes visent en effet à rendre possible la communication en présence de conditions hostiles dans lesquels les méthodes de communication classiques échouent (par exemple pour des communication ad hoc très dynamiques, en réponse à des catastrophe naturelles qui mettraient les infrastructures de communications traditionnelles hors-service, ou tout simplement dans des cas où le déploiement d'une infrastructure classique n'est pas rentable, comme dans les régions à faible densité de population), mais elles renouvellent également l'intérêt porté à des architectures de communication alternatives, comme les systèmes pub/sub basés sur le contenu, ou encore les réseaux bases sur le contenu promu récemment par Van Jacobson.

Du point de vue de la sécurité, le principal problème réside dans le fait que la protection de la vie privée ou de la confidentialité des données requiert un chiffrement du contenu de ces données, et dans le même temps ce contenu est à la base des décisions de transmission et de routage. Les exigences de sécurité entrent donc en conflit avec les besoins de transmission, et cette relation conflictuelle est à la base des problèmes que nous abordons dans cette thèse. Afin de dépasser cet apparent paradoxe et d'être en mesure de proposer des protocoles de communication opportuniste sécurisés, nous mettons donc en oeuvre des solutions permettant de réaliser certaines opérations utiles a la transmission sur des données chiffrées.

Nous avons repris dans ce résumé la structure de la thèse en anglais afin de faciliter les correspondances : chaque section du résumé correspond au chapitre de même numéro.

Cette thèse s'articule en trois grandes parties. Dans la première partie, nous présentons et classifions les protocoles opportunistes existants (section 0.2) puis nous nous analysons les nouveaux problèmes de sécurité qui s'en dégagent (section 0.3 [Shi10]), avant d'illustrer la démarche de communication au travers d'une plateforme expérimentale développée dans le cadre du projet européen Haggle (section 0.4). Dans une seconde partie nous nous intéressons tout particulièrement aux problèmes de sécurité dans le cadre des protocoles basés sur le contexte (section 0.5) et identifions trois problèmes principaux que nous traitons successivement : le problème de confidentialité des données (section 0.6 [SOM10b]), celui de respect de la vie privée (section 0.7 [SÖM09b]) et le problème nouveau d'assurance de calcul (section 0.8). Enfin nous évaluons la combinaison de ces solutions pour offrir un protocole de routage basé sur le contexte sécurisé en section 0.9. La dernière partie aborde le problème de routage sécurisé basé sur le contenu : nous proposons une solution originale à ce problème basée sur des couches de chiffrements multiples et commutatives (section 0.10 [SÖM09c, SÖM09a]) que nous complétons par un mécanisme d'établissement de clefs dépendant de la topologie (section 0.11 [SÖM10a]).

0.2 Routage et Transmission dans les Réseaux Opportunistes

Les principaux objets d'études de cette thèse sont les communications opportunistes sécurisées et, pour cerner ces problèmes efficacement, il est essentiel dans un premier temps de bien comprendre la définition des réseaux opportunistes et les contraintes qui en découlent.

Les réseaux opportunistes peuvent être vus comme une extension des réseaux ad hoc (MANET) et partagent donc leurs caractéristiques :

- ad hoc, qui implique essentiellement l'absence d'infrastructure et une organisation spontanée du réseau,
- ressources limitées car les noeuds du réseau sont souvent des appareils portables (des ordinateurs portables, des assistant personnels, des téléphones mobiles, ou même de simples senseurs),
- topologie dynamique du réseau due à la mobilité des noeuds.

Pour le dernier point, il est intéressant de noter que les MANET supportent la mobilité de façon très incomplète : le routage des messages se base en effet sur une route fixée entre l'émetteur et le récepteur, et toute modification de topologie durant une communication requiert de recalculer l'intégralité de la route. Cette démarche considère la mobilité comme un obstacle à surmonter et la communication n'est possible qu'en présence d'une topologie considérée comme stable pour un moment puis qui change vers une nouvelle configuration stable : cette approche n'est applicable que dans le cadre de réseaux à mobilité lente. Les réseaux opportunistes considèrent au contraire que la mobilité des noeuds peut être importante et exploite cette mobilité comme un avantage plutôt qu'un inconvénient : la mobilité physique est une possibilité supplémentaire de porter les messages pour palier a l'éventuelle l'absence de moyens de communications plus rapides et efficaces.

De là se dégage une autre caractéristique fondamentale des réseaux opportunistes, à savoir la tolérance au délai. Les réseaux opportunistes offrent en effet un support complet de la mobilité et, à ce titre, ne supposent pas l'existence d'un chemin de bout-en-bout. Le but de la communication opportuniste est en effet de porter le message à des relais de plus en plus près de la destination, et pour ce faire une stratégie dite *"store, carry and forward"* est adoptée en lieu et place du routage traditionnel :

- store (stoker) : les messages sont stockés en mémoire des noeuds mobiles en attendant une opportunité de communication,
- carry (porter) : les messages sont portés sur une certaine distance en exploitant la mobilité physique des utilisateurs,
- forward (transmettre) : lorsqu'une opportunité de communication avec un noeud plus proche de la destination se présente, le message est transmis au noeud en question.

Cette stratégie ne requiert donc pas d'établissement de route de bout-en-bout et s'accommode de la mobilité ou de la défaillance de certains noeuds : elle compense l'absence de connectivité bout-en-bout par la mobilité physique et la tolérance au délai.

Enfin, une troisième caractéristique majeure des communications opportunistes réside dans la structure condensée des messages. En effet, les messages sont transmis via des réseaux hétérogènes, et la structure des messages ne doit donc pas être dépendante d'un protocole en particulier. Ainsi toutes les informations concernant à la fois le contenu du message et la description de la destination (pour le routage) doivent être disponibles à un haut niveau d'abstraction, ce qui justifie une structure condensée. Cette structure est particulièrement adaptée aux communications multicast (multi-transmission) pour disséminer une information à plusieurs récepteurs, car elle permet d'envisager des protocoles de communication riches qui exploitent l'intégralité du contenu du message pour prendre les décisions de transmission et pas seulement un identifiant unique (une adresse par exemple) de la destination.

Prenant en compte ces différentes caractéristiques de nombreux protocoles adaptés aux communications opportunistes ont été proposés par la communauté scientifique ces dernières années. Nous les avons répertoriés puis classifier en fonction du cout de ces protocoles en terme d'utilisation du réseau et de la complexité de l'opération d'évaluation de la distance d'un noeud à une destination dans un premier temps (voir Figure 2.4 page 84), puis, faisant abstraction des problèmes de coût, en fonction de la quantité d'information utilisée pour prendre les décisions de transmission et de la précision avec laquelle est définie la destination (voir Figure 2.5 page 86). Nous avons alors dégagé trois grandes catégories de protocoles de communication opportunistes :

- Les protocoles de transmission aveugles, au sein desquels la destination du message est définie de façon très précise et explicite (via un identifiant unique). Ces protocoles adoptent principalement des stratégies de transmission épidémiques pour atteindre la destination, en se focalisant sur des heuristiques visant a réduire l'impact d'une telle approche sur la charge du réseau.
- Les protocoles de transmission basés sur le contexte, dans lesquels la destination du message est définie implicitement en fonction de son profil. Les décisions de transmission sont prises en comparant le contexte du message (correspondant au profil de la destination) avec le profil du noeud rencontré, et les messages sont transmis à des

noeuds avec taux de correspondance de plus en plus élevé. Au sein de cette catégorie il est possible de distinguer deux sous-ensembles : les protocoles complètement basés sur le contexte (qui prennent en compte l'intégralité du contexte dans le cadre des décisions de transmissions) et les protocoles partiellement basés sur le contexte qui ne prennent en compte qu'une information contextuelle bien précise (comme l'historique de rencontre ou les relations sociales des noeuds).

– Les protocoles de transmission basés sur le contenu, qui ne définissent pas de destination du tout : les noeuds expriment leurs intérêts pour un certain type de contenu, et les messages leur sont adressés en fonction de leur contenu.

L'intérêt de cette classification est de mettre en lumière les différences conceptuelles fondamentales entre ces trois catégories et l'évolution de la façon dont est considérée la destination (pour simplifier destination explicite, destination implicite et pas de destination). Ces trois catégories de transmission radicalement différentes posent des défis distincts, tant du point de vue de la transmission que du point de vue de la sécurité. Cette dernière constitue le sujet d'étude majeure de cette thèse et nous détaillons donc les problématiques de sécurité dans les réseaux opportunistes dans la section suivante.

0.3 Problématiques de Sécurité dans les Réseaux Opportunistes

Les problématiques de sécurité sont une composante essentielle de tout système de communication, et de nombreuses solutions ont été apportées pour résoudre les divers aspects de ce problème dans les réseaux de communication traditionnels. Ces solutions ne sont toutefois pas adaptées aux besoins et contraintes spécifiques des réseaux opportunistes. En effet, en plus des contraintes classiques des réseaux ad hoc mobiles MANET, qui requièrent des solutions de sécurité dynamiques, locales et auto-organisantes, la tolérance au délai et l'absence de connectivité bout-en-bout signifient que les protocoles interactifs entre émetteur et récepteur sont irréalisables dans les réseaux opportunistes et que l'accès à un serveur de sécurité ne peut être envisagé au cours de la communication. Enfin, l'architecture condensée soulève également de nouvelles difficultés, car elle implique que les problèmes de sécurité liés au contenu des applications et ceux liés aux informations de transmission doivent être traités de manière globale, contrairement au cas de l'architecture classique où la sécurité de chaque couche peut être assurée indépendamment des autres. Nous présentons donc dans cette section les différentes problématiques de sécurité dans les réseaux opportunistes en commençant par les problèmes généraux de coopération entre les noeuds puis ceux liés à l'intégrité, la confidentialité et le respect de la vie privée.

0.3.1 Coopération

La coopération entre les noeuds est essentielle au bon fonctionnement de tous les réseaux pairs-à-pairs, en particuliers les MANETs et réseaux opportunistes. Les solutions classiques pour éviter le développement de noeuds égoïstes qui ne participent pas aux opérations de transmission pour économiser leurs ressources énergétiques peuvent être classées en deux grandes catégories :

- mécanismes basés sur la réputation [BLB02a, BLB02b, MM02], où les noeuds acceptent de coopérer avec leurs voisins en fonction de l'historique de comportement de ces derniers, qui est mesuré par leur réputation (cette dernière croit lors d'un bon comportement et décroit pour les noeuds qui adoptent des comportements égoïstes),
- mécanismes basés sur les récompenses [RFJY03, HKLM03, GA04, BH03, ZCY03] dans lesquels les noeuds reçoivent une certaine récompense lorsqu'ils coopèrent, récompense qui peut ensuite être utilisée dans leur propre intérêt lorsqu'ils ont besoin du réseau à leur tour.

Les solutions existantes dans ces deux catégories ne sont en général pas adaptées aux réseaux opportunistes car elles font appel à une entité tierce de confiance qui doit être accessible à tout moment de la communication. Nous avons donc proposé une solution alternative [ÖSM07b] basée sur le principe de la patate chaude, dans laquelle les noeuds prennent la décision d'accepter ou non un message de façon aveugle (voir figure 3.1 page 92).

Lorsque un noeud N_1 a un message M à transmettre, il en informe ses voisins sans spécifier la destination du message M. Les voisins doivent alors décider à l'aveugle s'ils sont intéressés par ce message ou non. Supposons que le noeud N_2 est intéressé, N_2 envoie une récompense à N_1 qui lui transmet ensuite M. Si M intéresse effectivement N_2, alors N_2 a payé pour un message qui lui est destiné ce qui est équitable, mais si N_2 n'est pas intéressé par M, N_2 sera incité à transmettre le message à d'autres noeuds pour récupérer la récompense et ainsi la transmission du message et la coopération entre les noeuds sont assurées. Cette approche est optimiste dans le sens où une autorité de confiance n'est requise que pour :

- convertir les récompenses reçues par les noeuds en ressource utilisable dans le système,
- résoudre les conflits entre les noeuds qui se produisent si le noeud N_1 n'envoie pas le message M à N_2 après avoir reçu une récompense de la part de N_2.

Ce protocole assure donc une transmission équitable optimiste car l'autorité de confiance n'est requise qu'en cas de conflit entre les noeuds et même dans ce cas, l'accès à cette autorité n'a pas besoin d'être immédiat : cette autorité est dite hors-ligne (offline). Enfin ce protocole est flexible car il permet aux noeuds qui ne désirent pas coopérer (parce que leurs ressources restantes sont faibles) de ne pas le faire s'ils acceptent le risque de rater des messages qui leurs sont destinés, et il est indépendant du protocole de transmission et notamment du processus de choix du prochain noeud.

0.3.2 Authentification et intégrité

L'authentification des messages est un besoin de sécurité essentiel dans tout système de communication, et les solutions classiques consistent pour la source à signer le message avec une clef privée et un certificat associé. Cette solution peut être directement déployée dans les réseaux opportunistes car elle ne requiert pas de connectivite de bout-en-bout, mais simplement une phase antérieure à la communication, au cours de laquelle chaque noeud doit faire établir un certificat auprès d'une autorité de certification, qui est donc considérée hors-ligne et n'entre pas en conflit avec la communication opportuniste à proprement parler.

L'authentification bout-en-bout ne pose donc pas de souci particulier si les messages n'ont pas besoin d'être modifiés en cours de route. Cette dernière situation se produit toutefois dans deux cas intéressant dans le cadre des réseaux opportunistes :

- Lorsque les messages ont besoin d'être fragmentés pour avoir une taille de paquet conforme à une technologie réseau particulière. Une solution évidente est alors d'authentifier chaque fragment, mais des solutions plus intelligentes qui permettent l'authentification de l'ensemble des fragments de façon plus efficace existent (en se basant notamment sur des arbres de Merkle [AKK$^+$07]).
- Lorsque le codage réseau (network coding) est utilisé comme protocole de transmission. Le codage réseau est en effet très intéressant du point de vue performance pour disséminer une information dans une approche épidémique, mais il requiert de nombreuses modifications des paquets à chaque transmission. En effet chaque noeud doit transmettre une combinaison linéaire de l'ensemble des messages qu'il a reçu. En contrepartie de sa performance, le codage réseau est exposé à un risque très important de pollution : si un noeud envoie une mauvaise combinaison de messages, l'ensemble des noeuds du réseau peut se retrouver infecté par cette mauvaise combinaison et incapable de décoder le message final. Pour palier à ce type d'attaque il faut donc prévoir un mécanisme d'authentification des paquets qui permette aux noeuds intermédiaires de créer la signature d'une combinaison de messages à partir de la signature de chacun des messages reçus. Ce type de signature présente donc un caractère d'homomorphisme, puisque les signatures sont compatibles avec l'opération de combinaison linéaire. Les besoins de ces signatures sont donc très spécifiques et contraires aux besoins classiques des signatures (où la compatibilité avec l'opération de combinaison linéaire serait considérée comme un défaut, et nous avons proposé une solution à ce problème en modifiant un schéma de signature basé sur les couplages bilinéaires dans les courbes elliptiques [ÖSM07a, ÖSM07c].

0.3.3 Confidentialité et respect de la vie privée

Garantir la confidentialité des messages est un autre aspect fondamental de la sécurité des réseaux. Les solutions classiques pour ce problème passent par le chiffrement des données de bout-en-bout. On distingue deux grandes familles de chiffrements :

- les méthodes de chiffrements symétriques dans lesquels l'émetteur et le récepteur doivent partager une clef secrète,
- les méthodes de chiffrements asymétriques où chaque noeud possède généralement une clef publique (qui est accessible à tous) et une clef privée (que seul le noeud connaît).

Le cas des chiffrements symétriques ne peut être employé pour assurer la confidentialité bout-en-bout, car il suppose que l'émetteur et le récepteur entre dans une phase interactive d'établissement de la clef secrète, ce qui entre en conflit avec l'absence de connectivité bout-en-bout.

Les méthodes de chiffrement asymétriques sont plus adaptées a priori, mais soulèvent tout de même un problème : contrairement au cas de la signature où l'émetteur du message peut envoyer son certificat en même temps que le message qu'il a signé, le chiffrement requiert de l'émetteur la connaissance de la clef publique du récepteur avant l'envoi du message. Or ce certificat est en général disponible soit directement auprès du récepteur, soit auprès d'une entité tierce de confiance (l'autorité de certification par exemple). Bien que le chiffrement asymétrique ne soit donc pas interactif à proprement parler, la phase d'obtention du certificat du récepteur entre en conflit avec les contraintes des communications opportunistes. Pour contourner ce problème, il est possible d'utiliser le chiffrement à base d'identité [BF01] comme proposé par Asokan et al. [AKG^{+}07]. Le chiffrement basé sur l'identité permet en effet de dériver la clef publique de la destination à partir de l'identité de cette dernière, et permet donc de se passer de certificats (voir figure 3.3 page 3.3). Le chiffrement basé sur l'identité offre donc une solution crédible au problème de la confidentialité lorsque l'identité de la destination est connue comme c'est le cas pour les protocoles de transmission aveugles.

Le défi reste entier en revanche pour les protocoles plus riches comme par exemple les protocoles basés sur le contexte où l'identité de la destination n'est pas connue mais peut être déduite implicitement. Cette catégorie de protocoles requière donc de nouvelles méthodes de chiffrement qui permettent de dériver une clef de chiffrement à partir de la définition implicite de la destination et qui fassent en sorte que seule la destination puisse dériver la clef de déchiffrement associée.

De façon plus générale, le problème de respect de la vie privée couvre plusieurs aspects dont la confidentialité du contenu mais aussi la confidentialité de la communication, c'est-à-dire empêcher un noeud d'analyser le trafic pour savoir quel émetteur communique avec quel récepteur. Les besoins de protection de la vie privée peuvent être considérés à différents niveaux, et le niveau requis dépend de l'application, du point de vue adopté (émetteur, récepteur, noeud intermédiaire, noeud extérieur) and du niveau de confiance entre les entités. En nous basant sur nos travaux [SÖM09a], nous définissons un cadre général de modèles de respect de la vie privée, en considérant une donnée privée D_1 appartenant au noeud N_1 et qui doit être traitée par le noeud N_2 (pour prendre une décision de transmission par exemple) comme suit :

– **modèle 1, absence de secret** : ce modèle correspond au cas où N_1 ne requiert pas de protection pour D_1 du tout, N_2 (ou n'importe quel autre noeud) a accès à D_1 en clair dans le processus.
– **modèle, secret binaire** : dans ce modèle N_1 fait entièrement confiance à certains noeuds et pas du tout aux autres. Ainsi, si N_2 appartient au groupe de confiance de N_1, N_2 a accès à l'intégralité de D_1 en clair sinon N_2 n'a pas accès à D_1.
– **modèle 3, secret adaptable** : dans ce modèle, le niveau de protection dépend de la relation entre les noeuds : N_1 fait partiellement confiance à N_2. Le niveau de confiance peut être basé sur l'appartenance à une communauté. N_2 doit alors être en mesure d'accéder à une partie de D_1 variable selon le niveau de confiance.
– **modèle 4, secret complet** : contrairement aux modèles précédents, ce modèle fait

référence au cas où les noeuds n'ont aucune confiance les uns envers les autres, et dans ce cas N_2 doit être en mesure de manipuler les données D_1 sans y avoir accès en clair.

La nature de la donnée secrète D_1 (qui peut correspondre notamment au contexte ou au contenu) ainsi que le niveau de confiance et de protection requis dépendent du scenario considéré et nous analyserons plus en détails ces différents modèles pour chaque scenario dans la suite du manuscrit. De façon générale, le défi est de permettre la prise de décision de transmission dans les différents modèles (voir figure 3.4 page 103). En particulier, les modèles de respect de la vie privée les plus exigeants (3 et 4) impliquent des méthodes de calcul sur des données chiffrées, ce qui constitue le coeur de notre travail de recherche.

0.4 L'architecture Haggle

Haggle est un projet européen qui est l'une des initiatives les plus avancées en matière de recherche dans les réseaux opportunistes. L'un des objectifs principaux de Haggle était d'établir une nouvelle architecture de communication opportuniste que nous présentons dans cette section, en nous concentrant sur l'aspect sécurité auquel nous avons principalement contribué.

0.4.1 Architecture Globale

Afin de respecter les nombreuses caractéristiques originales des communications opportunistes, l'architecture Haggle est une architecture condensée basée sur quatre composants essentiels (voir Figure 4.1 page 107) :

1. **HaggleKernel :** Le HaggleKernel est le noyau de l'architecture et est implémenté sous forme d'un ordonnanceur d'événements qui coordonne la communication entre les Managers.
2. **Datastore :** Un répertoire central contenant des données (d'applications) et des informations à propos des noeuds et de leurs interfaces. Le Datastore est accessible en lecture par tous les Managers.
3. **Managers :** Les Managers sont responsables de maintenir certaines parties du Datastore ainsi que de tâches spécifiques comme gérer les interfaces de communications ou gérer les fonctions relatives à la sécurité.
4. **Modules :** Les Managers peuvent faire appel à des modules qui implémentent des fonctionnalités bien spécifiques à un algorithme donné (par exemple un algorithme de transmission particulier).

0.4.2 Le Security Manager

Nous nous intéressons maintenant de façon spécifique au Security Manager dont nous avons implémenté une première version en Java [ÖSTB08] puis que nous avons porté et étendu à la version adulte de Haggle en C++ [ÖS09].

L'architecture du Security Manager prend en compte la forte interaction avec les autres Managers (voir Figure 4.2 page 110) :

- Les autres Managers font en effet appel au Security Manager pour effectuer des opérations de sécurité, par exemple le chiffrement d'une donnée,
- Le Security Manager doit également faire appel aux autres Managers pour communiquer avec le Security Manager d'un noeud voisin, dans le but par exemple d'établir des clefs communes.

Ces interactions nombreuses et complexes sont exigeantes du point de vue de l'ordonnancement des évènements et c'est pourquoi l'architecture basée sur un ordonnanceur est plus adaptée qu'une architecture séquentielle.

Nous avons ainsi implémenté des briques de bases de chiffrements et de signature pour la confidentialité et l'intégrité des données et avons pris en compte deux niveaux de sécurité :

- Le niveau applicatif pour la protection des données de bout-en-bout entre l'émetteur et le récepteur d'un message,
- Le niveau protocolaire pour la protection des données lors de leur transmission entre deux noeuds, c'est-à-dire une protection contre des attaquants indiscrets qui écoutent les transmissions.

Le second cas requiert un chiffrement et une signature d'un noeud au suivant et ne pose pas de problème particulier dans les réseaux opportunistes car il y a toujours la possibilité d'établir des clefs entre noeuds voisins. Nous nous intéressons donc dans la suite de cette section à la sécurité au niveau applicatif.

0.4.3 Certificats d'Attributs

Un concept central pour établir les communications dans Haggle est le concept de communauté afin de faciliter l'établissement de confiance. Les personnes appartenant à une même communauté ont en effet plus naturellement tendance à se rendre service et à s'entraider.

Dans Haggle, chaque noeud a alors une liste d'attributs qui le caractérisent et indiquent les communautés auxquelles il appartient. Lorsque deux noeuds se rencontrent ils s'échangent leurs attributs pour découvrir les communautés qu'ils ont en commun.

Du point de vue de la sécurité cette approche présente deux défis majeurs :

- L'appartenance à une communauté est un élément de la vie privée de chaque noeud et les noeuds pourraient donc souhaiter ne pas divulguer l'intégralité de leurs caractéristiques à leurs voisins, mais simplement découvrir les attributs en communs.
- Nous avons mentionné que l'appartenance à une communauté impliquait un certain niveau de confiance, il est donc important de pouvoir prouver l'appartenance a une communauté et d'empêcher un noeud de prétendre avoir des caractéristiques qu'il n'a pas.

Afin de palier à ces problèmes nous avons proposé et implémenté des certificats d'attributs appelés HaggleCertificates qui prouvent la possession d'une certaine caractéristique au lieu de prouver une identité comme c'est le cas avec les certificats classiques. Ces certificats ont donc une structure similaire à celle des certificats classiques (type X509) sauf qu'ils ne sont pas nominatifs mais prouvent simplement une caractéristique. Ces certificats sont délivrés soit par une autorité de confiance dans le cas des communautés organisées, soit par un noeud Haggle qui décide d'établir une nouvelle communauté de manière spontanée.

Par ailleurs, la caractéristique prouvée par le certificat peut être chiffrée de sorte que les noeuds qui ne font pas partie de la communauté ne puissent pas savoir de quelle communauté il s'agit.

Ces certificats ont été implémentés en nous basant sur les primitives de bases (chiffrement, déchiffrement, signature et vérification) offerte par OpenSSL et déclenchent des évènements specifique reconnus par le HaggleKernel (l'ordonnanceur). De plus les certificats sont enregistrés dans un registre spécifique reprenant la structure du Datastore et qui fait le distinguo entre les certificats du noeud, ceux des voisins du noeud, et ceux des autorités de confiance. Les clefs privées de certificats sont chiffres avec une clef connue du Security Manager pour éviter l'accès à ces clefs de certificats par les autres Managers. Le code correspondant à la librairie HaggleCetificate ainsi qu'à l'ensemble de l'architecture Haggle est disponible à l'adresse suivante : http ://code.google.com/p/haggle/.

En nous basant sur ces certificats, nous avons réalisé un scenario pratique dans lequel les noeuds ne transmettent que les messages de membres de leur communauté et un scenario où tous les noeuds participent à la transmission mais seuls les membres de la communauté ont accès à la donnée. Ces scenarii permettent de préserver la confidentialité et la vie privée dans le modèle 2 et sont surtout utilisés comme preuve de l'intérêt du concept de communautés pour la transmission opportuniste. Pour atteindre des modèles plus exigeants il faut envisager des méthodes de calcul sur données chiffrées qui vont au-delà de l'adaptation de concepts classiques comme les certificats et qui sont l'objet des études présentées dans la suite de cette thèse.

0.5 Problèmes de Sécurité dans les Protocoles Basés sur le Contexte

Comme nous l'avions évoqué en section 0.2, les protocoles basés sur le contexte forment une catégorie de protocoles de transmission très différents des approches classiques de routage et qui présentent des problèmes de sécurité nouveaux. Nous présentons donc plus en détail le fonctionnement des protocoles basés sur le contexte dans cette section puis nous analysons les problèmes de sécurité associés.

0.5.1 Transmission basée sur le contexte

Dans les grandes lignes, la transmission basée sur le contexte consiste en une comparaison du contexte d'un noeud avec celui de la destination et la transmission du message aux noeuds avec un degré de correspondance croissant jusqu'à atteindre la destination. L'idée sous-jacente est que plus les noeuds partagent un contexte important plus la probabilité qu'ils se rencontrent est grande. Pour préciser cela nous utilisons les paramètres suivants pour les noeuds :

- Nous considérons un réseau composé de n noeuds $\{N_i\}_{1\leq i\leq n}$.
- Le contexte d'un noeud N_i, également appelé profil et dénoté par $Prof(i)$, est défini comme un ensemble d'attributs $\{A_{i,j}\}_{1\leq j\leq m}$: $A_{i,j}$ est le j-ième attribut du noeud N_i.
- Un attribut est un couple (nom de l'attribut, valeur de l'attribut). Le j-ième attribut du noeud N_i, est ainsi dénoté par $A_{i,j} = (E_j, V_{i,j})$. Le nom de l'attribut ne dépend pas de N_i, seule la valeur change d'un noeud a l'autre.

En ce qui concerne les messages, chaque message M est divisé en deux parties :

- L'en-tête $\mathcal{H}(M)$, qui contient des informations à propos de la destination du message,
- Le contenu $\mathcal{PLD}(M)$, qui correspond aux données utiles du message.

L'en-tête $\mathcal{H}(M)$ se présente sous la forme d'une concaténation d'attributs :

$$\mathcal{H}(M) = ||_{j\in L_M} A_{M,j},$$

où L_M est un sous-ensemble d'indices dans $[1, m]$.

$\mathcal{H}(M)$ est également appelé le contexte du message et il défini implicitement la destination comme étant tout noeud N_i qui partage tous les attributs définis dans $\mathcal{H}(M)$, autrement dit tout noeud N_i tel que $\forall j \in L_M$, $A_{M,j} = A_{i,j}$.

Pour transmettre un message M, un noeud N_i sélectionne le voisin le plus apte à rencontrer la destination. Pour ce faire, il diffuse l'en-tête $\mathcal{H}(M)$ à ses voisins. Un voisin N_k peut alors extraire le sous-ensemble de correspondance $L_{M,k} \subset L_M$ qui contient les indexes des attributs partagés entre le contexte de M et celui de N_k, de telle sorte que $\forall j \in L_{M,k}, A_{M,j} = A_{k,j}$. Il est alors possible de calculer le ratio de correspondance entre un message M et un noeud N_k qui est défini comme étant le nombre d'attributs partagés dans les contextes de M et N_k divisé par le nombre d'attributs dans $\mathcal{H}(M)$, soit

$$p_k(M) = \frac{|L_{M,k}|}{|L_M|}.$$

N_i transmet ensuite l'intégralité du message au voisin qui présente le ratio de correspondance le plus élevé. On remarque que la destination du message est par définition celle qui a le ratio le plus élevé, à savoir 1.

Pour illustrer ce protocole de transmission nous nous appuyons sur le scenario (II) présenté Figure 5.2 page 128. Les profils des noeuds de cette figure sont présentés en Figure 5.1 page 126.

Dans ce scenario nous supposons donc que N_4 souhaite déclarer sa flamme à N_1. N_4 construit le message M_2 dont le contenu est :

$$\mathcal{PLD}(M_2) = \text{"}\textit{I love you}\text{"}$$

Et dont l'en-tête comprend trois attributs :

$$\mathcal{H}(M_2) = (Mail, alice@inria.fr)||(Workplace, INRIA)||(Status, student).$$

Dans ce cas, nous avons $L_{M_2} = \{1, 2, 3\}$.

L'en-tête est diffusée à N_2 et N_3.

- N_2 partage les attributs workplace et status avec le contexte du message, donc $L_{M_2,2} = \{2,3\}$ et $p_2(M_2) = 2/3$;
- N_3 partage uniquement l'attribut status avec le contexte du message, donc $L_{M_2,3} = \{3\}$ et $p_3(M_2) = 1/3$.

Ni N_2 ni N_3 ne partagent l'intégralité du contexte du message donc aucun des deux n'est la destination de M_2. Cependant N_2 a plus de chance de rencontrer la destination de M_2 que N_3, c'est pourquoi N_4 transmet le message M_2 à N_2.

N_2 réitère le processus et envoie M_2 à N_1 qui présente un ratio de correspondance $p_1(M_2) = 1$ et qui est donc une destination de M_2.

0.5.2 Problématiques de sécurité

Le scenario précédent présente le déroulement du protocole dans le cas où tous les noeuds sont honnêtes et se font entièrement confiance, cependant dans une approche plus réaliste il faut considérer le cas de noeuds curieux voire de noeuds malveillants. Par exemple, dans ce scenario, le noeud 4 est un professeur déclarant sa flamme à une étudiante et il est donc clair que ce genre de message doit être chiffré et accessible à son destinataire uniquement. Dans la suite nous présentons donc les problèmes de sécurité et la façon de sécuriser la communication tout en permettant le bon déroulement du protocole. Pour cela nous commençons par décrire le problème de protection de la confidentialité des données, mais aussi la protection de la vie privée des utilisateurs et enfin la garantie de l'honnêteté du calcul du ratio de correspondance.

0.5.2.1 Confidentialité du contenu

Le premier besoin de sécurité est, comme dans toute communication classique, la protection de la confidentialité des données : il faut chiffrer le contenu de bout en bout pour que seule la destination puisse accéder au contenu du message.

La difficulté vient toutefois du fait que, contrairement aux solutions classiques de type SSL/TLS [Res00] ou IPSec [KS05], la communication basée sur le contexte ne peut s'appuyer sur une clef partagée de bout-en-bout du fait de la nature tolérante au délai de la communication.

Il faut donc envisager des solutions asymétriques et ne requérant pas de certificats comme le chiffrement basé sur l'identité. Cependant dans le cas de la communication basée sur le contexte, la difficulté supplémentaire vient du fait que la destination n'est pas connue explicitement mais elle est définie implicitement (la source n'a pas d'identifiant a priori de la destination). Ainsi donc il faut définir un nouveau mécanisme de chiffrement et de déchiffrement de telle sorte que la clef de déchiffrement ne puisse être dérivée que par la destination du message.

Ainsi, la protection de la confidentialité du contenu passe par la définition des deux primitives suivantes :

- ENCRYPT_PAYLOAD : utilisée par la source pour chiffrer le contenu du message. Cette fonction doit être publique et la clef de chiffrement doit dépendre des attributs

de la destination précisés dans l'en-tête du message.
- DECRYPT_PAYLOAD : utilisée par la destination pour déchiffrer le contenu du message. Cette fonction doit être privée et seuls les noeuds ayant l'intégralité des attributs requis dans l'en-tête du message doivent être en mesure de calculer la clef de déchiffrement requise.

0.5.2.2 Vie privée des utilisateurs

Le chiffrement du contenu ne suffit pas toutefois à assurer le secret de la communication. En effet, le contexte d'un message renseigne implicitement sur la destination du message. Ainsi dans le scenario proposé, une simple lecture de l'en-tête du message permet de voir que N_4 communique avec une étudiante, et si le nombre de message échangés entre eux est important, cela suffira à éveiller des soupçons.

Le contexte est donc une information privée qu'il faut éviter de divulguer à tout un chacun et qu'il faut donc chiffrer. Dans le même temps il est important de laisser la possibilité aux noeuds intermédiaires de calculer le ratio de correspondance et donc de déterminer les attributs partagés. Ainsi les noeuds intermédiaires doivent être à même de découvrir les attributs partagés sans rien apprendre sur les attributs non partagés. La capacité de correspondance doit donc dépendre des attributs des noeuds. Le respect de la vie privée peut donc être obtenu jusqu'au modèle 3 au plus, car les noeuds intermédiaires doivent nécessairement découvrir les attributs partagés, et c'est donc à ce modèle que nous nous intéresserons. Enfin, notons que la fonction de chiffrement est en fait une fonction d'encodage plutôt, vu qu'il n'y a pas de déchiffrement requis. Enfin, la fonction doit contenir une part d'aléa pour éviter que les messages successifs a une même destination soient relies par de simples observateurs.

Au final, la préservation de la vie privée des utilisateurs dans le modèle 3 requiert la définition des deux primitives suivantes :
- ENCRYPT_HEADER : utilisé par la source pour chiffrer les informations de contexte du message. Cette fonction doit être publique et permettre aux noeuds intermédiaires de comparer leur profil avec le contexte chiffré afin de transmettre le message de façon correcte.
- MATCH_HEADER : utilisée par un noeud intermédiaire pour déterminer si un attribut chiffré est partagé par le noeud ou non. Cette fonction ne doit toutefois pas révéler d'information sur les attributs non partagés.

0.5.2.3 Assurance de calcul

Les problèmes de protection de la confidentialité et de la vie privée traitent principalement le cas de noeuds honnêtes mais curieux. Un tout autre type d'attaque est possible si l'on considère des noeuds carrément malveillants et qui ne respectent donc pas le processus du protocole.

En effet, un noeud malveillant pourrait très bien falsifier son ratio de correspondance et annoncer une valeur plus élevée dans le but de récupérer un message. Ce faisant il détourne le trafic et peut mener une attaque de type trou noir pour dégrader les performances d'ensembles du réseau.

Il est donc nécessaire, pour faire face à ce type d'attaques, de disposer d'un mécanisme qui prouve la valeur du ratio de correspondance. Ce problème est donc très différent de celui de la vie privée, mais le fait d'imposer un respect de la vie privée rend ce problème encore plus difficile. En effet une preuve toute simple serait de demander aux voisins de divulguer leurs profils, mais cela expose leur vie privée de façon criante. Le défi est donc de proposer une solution qui garantisse l'exactitude de la valeur du ratio de correspondance tout en préservant la vie privée des noeuds. Pour ce faire nous définissons une primitive de sécurité supplémentaire :

- VERIFY_RATIO : utilisée par un noeud intermédiaire pour déterminer si le ratio de correspondance annoncé par ses voisins est correct, dans le sens où il correspond à l'en-tête chiffrée qu'ils ont reçu. Cette fonction ne doit révéler aucune information quant aux attributs non partagés par les noeuds.

0.5.3 Bilan et vue d'ensemble

Nous avons identifié trois besoins principaux de sécurité et avons introduits des primitives pour répondre à ces besoins. Ainsi la source N_S utilise ENCRYPT_PAYLOAD et ENCRYPT_HEADER pour chiffrer respectivement le contenu et l'en-tête du message. Lorsqu'un noeud intermédiaire N_i reçoit un message chiffré il transmet l'en-tête à ses voisins. Les voisins utilisent MATCH_HEADER pour calculer le ratio de correspondance tout en préservant la vie privée de la destination et renvoient le résultat à N_i. N_i utilise VERIFY_RATIO pour s'assurer de l'exactitude du ratio reçu et prend ensuite une décision de transmission. Lorsque le message atteint un noeud qui partage tous les attributs de l'en-tête, ce dernier est une destination et il exécute DECRYPT_PAYLOAD pour accéder au contenu du message.

Toutes ces primitives requièrent certaines informations secrètes liées aux profils des noeuds et nous proposons donc de diviser nos protocoles en deux phases pour respecter les contraintes de la communication opportuniste :

- une phase d'organisation au cours de laquelle les noeuds contactent une entité tierce de confiance TTP pour obtenir les informations secrètes liées à leur profil ainsi que les paramètres globaux du système,
- une phase d'exécution au cours de laquelle la communication opportuniste à proprement parler a lieu sans accès au TTP (qui est donc considéré hors-ligne).

Dans les grandes lignes, les caractéristiques principales de nos solutions sont :

1. Tous les noeuds ont un profil composé de m attributs. Le nom des attributs est le même pour tous les noeuds mais la valeur de ces attribut change d'un noeud à l'autre. Les noeuds obtiennent des informations secrètes correspondant à leur profil de la part d'un TTP hors-ligne.

2. Confidentialité du contenu :

   (a) Le contenu de chaque message est chiffré à l'aide d'une méthode de chiffrement basée sur l'identité.
   (b) Un noeud peut déchiffrer le contenu d'un message via la fonction de déchiffrement basée sur l'identité associée seulement s'il partage tous les attributs du contexte du message.
   (c) La principale idée dans l'implémentation de ces fonctions de confidentialité bout-en-bout est d'utiliser une somme de $|L_M|$ arguments au lieu d'un seul argument dans les chiffrements basés sur l'identité classiques.

3. Vie privée des utilisateurs :
   (a) Chaque en-tête de message contient $|L_M| \leq m$ attributs, dont les valeurs sont chiffrées avec une fonction appartenant à un système de chiffrement permettant la recherche de mots clefs (Public key Encryption with Keyword Search PEKS).
   (b) Un noeud peut déterminer les attributs partagés en utilisant une autre fonction du système PEKS ainsi que ses clefs privées.
   (c) L'idée principale permettant d'offrir la possibilité de vérifier les attributs partagés réside dans la modification des rôles joués par les différentes entités du système PEKS et par l'introduction d'un TTP hors-ligne.

4. Assurance de calcul :
   (a) La vérification des attributs partagés requiert le calcul d'une valeur pseudo-aléatoire.
   (b) Cette valeur pseudo-aléatoire est insérée dans une fonction de hachage et n'est connue que des noeuds partageant l'attribut en question. Cette valeur est donc une preuve de la possession de l'attribut.
   (c) Les preuves de possessions sont insérées dans un filtre de Bloom pour protéger la vie privée du noeud.
   (d) Le filtre de Bloom construit par le noeud est comparé à un filtre de Bloom construit par la source pour calculer le ratio de correspondance correct.

Nous abordons ces trois problèmes plus en détail dans les trois prochaines sections et consacrons la section suivante à une évaluation globale de la solution.

0.6 Confidentialité du contenu dans les Protocoles Basés sur le Contexte

Comme nous l'avons mentionné en section 0.5 la protection de la confidentialité du contenu requiert une solution non interactive de chiffrement bout-en-bout, qui rappelle le chiffrement basé sur l'identité [BF01]. La différence est toutefois que la source du message ne connaît pas d'identifiant unique de la destination mais un ensemble d'attributs de cette dernière. Nous proposons donc un nouveau schéma de chiffrement où l'identité est remplacée par de multiples attributs.

0.6.1 Chiffrement basé sur des identités multiples

Nous définissons un schéma basé sur de multiples identités à partir des quatre algorithmes probabilistes suivants :

1. `MIB-Setup :` prend en entrée un paramètre de sécurité sp et retourne $params$ (les paramètres du système) et $master - key$ (clef de l'autorité).

   De façon plus précise, étant donné $sp \in \mathbb{Z}^+$ l'algorithme opère comme suit :

   (a) générer un nombre premier q, deux groupes $\mathbb{G}_1$ et $\mathbb{G}_2$ d'ordre q, et un couplage bilinéaire $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Choisir un générateur $P \in \mathbb{G}_1$ au hasard.

   (b) Choisir au hasard $s \in \mathbb{Z}_q^*$, et fixer $P_{pub} = sP \in \mathbb{G}_1$.

   (c) Choisir deux fonctions de hachage cryptographiques :
   - $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$,
   - $H_2 : \mathbb{G}_2 \to \{0,1\}^\nu$ pour un certain $\nu \in \mathbb{Z}^+$.

   (d) Les paramètres du système sont alors définis comme étant :

   $$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, H_1, H_2 \rangle ,$$

   Et la $master - key$ est $s \in \mathbb{Z}_q^*$.

2. `MIB-Extract :` prend en entrée $params$, $master - key$, et une identité arbitraire $ID \in \{0,1\}^*$, et retourne la clef privée d_{ID}. Cette dernière est construite de la façon suivante :

   (a) Calculer $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$,

   (b) Fixer $d_{ID} = sQ_{ID}$.

3. `MIB-Encrypt :` prend en entrée $params$, *un ensemble d'identités* $\{ID_i\}_{1 \le i \le \mu}$ (avec $\mu \in \mathbb{Z}^+$) et un message M, et retourne un texte chiffré M' construit de la façon suivante :

   (a) Pour chaque $1 \le i \le \mu$, calculer $Q_{ID_i} = H_1(ID_i) \in \mathbb{G}_1^*$,

   (b) Choisir $r \in \mathbb{Z}_q^*$ de façon aléatoire,

   (c) Fixer $g_{\{ID_i\}_{1 \le i \le \mu}} = \hat{e}(\sum_{i=1}^{\mu} Q_{ID_i}, P_{pub}) \in \mathbb{G}_2$

   (d) M' est alors le couple :

   $$M' = \left\langle rP, M \oplus H_2(g^r_{\{ID_i\}_{1 \le i \le \mu}}) \right\rangle .$$

4. `MIB-Decrypt :` prend en entrée $params$, M' (texte chiffré) et *un ensemble de clefs privées* $\{d_{ID_i}\}_{1 \le i \le \mu}$. Retourne le texte clair M. Pour cela, notons $M' = \langle U, V \rangle$. Si $U \notin \mathbb{G}_1^*$ rejeter le texte chiffré. Sinon, le texte clair s'obtient en calculant :

   $$V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} d_{ID_i}, U)) = M$$

0.6.2 Application au problème de confidentialité du contenu

Pour garantir la confidentialité du contenu nous proposons de mettre en oeuvre le système de chiffrement basé sur des identités multiples en deux phases.

0.6.2.1 Phase d'organisation

Durant cette phase une entité tierce de confiance PKG exécute le protocole `MIB-Setup` et en retire $params$ et une clef maitresse $master - key$.

Ensuite les noeuds contactent individuellement le PKG qui leur fournit les paramètres globaux $params$ ainsi que des clefs privées correspondant à leurs profils. Ainsi, le noeud N_i dont le profil est $Prof(i) = ||_{1 \leq j \leq m} A_{i,j}$ recevra les clefs $\{A_{priv_{i,j}}\}_{1 \leq j \leq m}$, avec

$$A_{priv_{i,j}} = \texttt{MIB-Extract}(params, master - key, A_{i,j}).$$

A l'issue de cette phase, chaque noeud a donc m secrets, et le PKG n'est plus requis (il est hors-ligne).

0.6.2.2 Phase d'exécution

Durant la phase de communication opportuniste, la source d'un message peut chiffrer son contenu en fonction de l'en-tête du message, en utilisant les attributs de l'en-tête comme identités multiples :

$$\begin{aligned} \mathcal{PLD}(M') &= ENCRYPT\_PAYLOAD(M) \\ &= \texttt{MIB-Encrypt}(params, \{A_{M,j}\}_{j \in L_M}, \mathcal{PLD}(M)). \end{aligned}$$

La destination quant à elle dispose par définition de tous les attributs de l'en-tête du message et donc des clefs privées associées. Elle peut donc déchiffrer le contenu de la façon suivante :

$$\begin{aligned} DECRYPT\_PAYLOAD(M') &= \texttt{MIB-Decrypt}(params, \mathcal{PLD}(M'), \{A_{priv_{M,j}}\}_{j \in L_M}) \\ &= \mathcal{PLD}(M). \end{aligned}$$

Une caractéristique importante de cette solution est que la source peut chiffrer le contenu du message avec un ensemble d'attributs quelconques, elle n'a pas besoin de partager ces attributs pour pouvoir chiffrer. Nous étudions maintenant la sécurité de cette solution.

0.6.3 Evaluation de Sécurité

Le schéma MIBE est dérivé du schéma IBE propose par Boneh [BF01], qui a prouvé que IBE était sémantiquement sûr face aux attaques à texte clair (IND-ID-CPA) par un raisonnement réductionniste. Nous nous inspirons donc de leur preuve pour prouver également que notre schéma est sémantiquement sûr.

Nous introduisons tout d'abord un nouveau modèle de sécurité IND-MID-CPA qui étend le modèle IND-ID-CPA en prenant en compte les possibilités accrues des attaquants du schéma MIBE. En effet dans ce dernier il est possible de demander le chiffrement d'un message sous plusieurs identités à la fois et il faut donc prendre en compte l'ensemble de ces identités dans la preuve.

La différence est donc que l'adversaire peut exiger de la part du challenger d'être testé sur deux messages chiffrés sous un ensemble d'identités de son choix, à condition que pour au moins une des identités il ne connaisse pas la clef privée associée.

Nous prouvons dans ces conditions que la probabilité de réussite de l'adversaire est négligeable et donc que notre schéma est sémantiquement sûr dans le modèle IND-MID-CPA sous l'hypothèse largement acceptée que le problème de Diffie-Hellman bilinéaire est difficile dans les groupes considérés.

La conséquence pratique de cette preuve, est que notre solution protège bien le contenu de bout-en-bout : un noeud intermédiaire qui possèderait une partie des clefs privées mais pas toute n'obtiendrais pas plus d'information sur le contenu qu'un noeud qui n'en possède aucune.

Ceci conclue la présentation de notre solution pour la protection de la confidentialité du contenu et nous présentons dans la section suivante la solution visant à protéger la vie privée des utilisateurs.

0.7 Respect de la Vie Privée des Utilisateurs dans les Protocoles Basés sur le Contexte

Comme évoqué en section 0.5, le respect de la vie privée des utilisateurs requiert un mécanisme qui permette d'encoder les attributs de l'en-tête d'un message de façon à autoriser les noeuds intermédiaires à détecter les attributs correspondants à leur profil et seulement ceux-là.

Une première idée proposée dans [NGP07] est d'utiliser des fonctions de hachage cryptographiques comme fonction d'encodage. Et les noeuds intermédiaires peuvent hacher les attributs de leurs profils et détecter les correspondances par simple identification entre les attributs encodés de l'en-tête et leur profil haché. L'idée est, qu'étant donné que les fonctions de hachage cryptographiques sont difficiles à inverser, un noeud ne pourrait retrouver un attribut encodé à partir de son empreinte. La faille dans ce raisonnement vient du fait que les valeurs des attributs appartiennent à un espace restreint (un dictionnaire) et ne sont pas des chaines pseudo-aléatoire. Il est donc aisé d'exécuter une attaque dictionnaire en hachant toutes les valeurs possibles et en effectuant la correspondance par la suite. Cette première approche montre bien que le problème de respect de la vie privée est loin d'être simple. Il requiert des méthodes de calcul (correspondance en l'occurrence) sur des données chiffrées et pour cela nous proposons une solution basée sur des chiffrements qui permettent la recherche de mots-clefs. Nous présentons tout d'abord cette technique puis nous montrons comme elle est appliquée à notre problème de façon spécifique.

0.7.1 Chiffrement à clef publique avec recherche de mots-clefs (PEKS)

Cette méthode de chiffrement a pour objet au départ de permettre la recherche d'un mot clef dans une liste de mots-clefs chiffrés et semble donc adaptée à notre problème. Nous présentons ici une méthode de construction due à Boneh et al. [BCOP04] qui consiste en les cinq algorithmes probabilistes suivants :

1. `SE-Setup`(sp) : Prends en entrée un paramètre de sécurité sp, et retourne les paramètres du système $params$, de la façon suivante :
   (a) Générer un nombre premier prime q, deux groupes $\mathbb{G}_1$ and $\mathbb{G}_2$ d'ordre q, et un couplage bilinéaire cryptographique $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
   (b) Choisir un générateur P de $\mathbb{G}_1$.
   (c) Choisir deux fonctions de hachage cryptographiques :
   – $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$,
   – $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^{\log q}$.
   Les paramètres du système sont alors :
   $$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, H_1, H_3 \rangle .$$
2. `SE-KeyGen`$(params)$: prends en entrée $params$ et génère une paire de clefs publique/privée pk_A, sk_A. Pour cela il suffit de choisir $s \in \mathbb{Z}_q^*$ de façon aléatoire et de poser $pk_A = sP$ et $sk_A = s$
3. `SE-PEKS`$(params, pk_A, W)$: étant donnés une clef publique pk_A et un mot-clef W, l'algorithme retourne un mot-clef chiffré S_W en choisissant aléatoirement $r \in \mathbb{Z}_q^*$ puis en calculant :
   $$S_W = \langle rP, H_3(\hat{e}(H_1(W), rpk_A)) \rangle .$$
4. `SE-Trapdoor`$(params, sk_A, W)$: étant donnés une clef privée sk_A et un mot-clef W l'algorithme produit la trappe suivante :
   $$T_W = sk_A H_1(W).$$
5. `SE-Test`$(params, S_{W_1}, T_{W_2})$: Cet algorithme prend en entrée :
   – un mot-clef chiffré $S_{W_1} =$ `SE-PEKS`$(params, pk_A, W_1) = \langle U, V \rangle$,
   – et une trappe $T_{W_2} =$ `SE-Trapdoor`$(params, sk_A, W_2)$.
   Il retourne
   – 1 si $H_3(\hat{e}(T_{W_2}, U)) = V$,
   – 0 si $H_3(\hat{e}(T_{W_2}, U)) \neq V$.
   Soit 1 si $W_1 = W_2$ et 0 sinon.

Ce schéma permet donc à une source de créer des mots-clefs chiffrés pour une destination en utilisant la clef publique de la destination et la fonction `SE-PEKS`. La destination peut créer des trappes pour certains mots-clefs à l'aide de sa clef privée et `SE-Trapdoor`. Enfin un noeud intermédiaire peut verifier si une trappe correspond à un mot-clef chiffre en utilisant `SE-Test`. Ce scenario est présenté en Figure 7.1 page 161.

0.7.2 Nouvelle instanciation du protocole PEKS pour garantir le respect de la vie privée des utilisateurs

Le mécanisme PEKS est intéressant pour le problème de protection de la vie privée des utilisateurs car la fonction d'encodage `SE-PEKS` est publique, et la capacité à verifier une correspondance avec `SE-Test` est soumise à la connaissance de la trappe associée qui est une information secrète qui peut être considérée comme une sorte de clef privée. Il y a toutefois deux problèmes majeurs dans l'application de ce mécanisme dans un cadre de réseaux opportunistes :

- Le premier vient du fait qu'un noeud qui souhaite chiffrer un mot-clef avec `SE-PEKS` doit connaître la clef publique de la destination. PEKS a donc certaines propriétés similaires à celles des chiffrements à clef publiques traditionnels comme RSA, et nous avons déjà évoqué le fait que ces propriétés étaient inadaptées aux réseaux opportunistes.
- Le second est que les noeuds intermédiaires qui voudraient effectuer une correspondance avec `SE-Test` ont besoin de la trappe associée. Cette trappe ne peut être générée que par la destination puisqu'elle requiert la clef privée de la destination. La destination devrait donc diffuser des trappes dans tout le réseau ou au moins sur toute la route entre source et destination, ce qui, encore une fois, est inadapté aux réseaux opportunistes.

Nous proposons donc une instanciation alternative de PEKS en introduisant une entité tierce de confiance TTP hors-ligne qui remplacerait la destination. L'idée est similaire à celle déployée dans le chiffrement basé sur l'identité par rapport au chiffrement traditionnel, où l'identité et la clef publique d'une entité tierce de confiance remplacent la clef publique de la destination. Notre protocole s'articule donc encore une fois en deux phases.

0.7.2.1 Phase d'organisation

Au cours de cette phase, le TTP exécute `SE-Setup` et `SE-KeyGen` obtenant de la sorte $params$ et une paire de clefs pk_{TTP}/sk_{TTP}. $params$ et pk_{TTP} sont publiques et donc connus de tous les noeuds.

Ensuite chaque noeud contacte le TTP qui lui fournit les trappes correspondants à son profil calculées avec la clef privée du TTP. Ainsi un noeud N_i reçoit
$T_{i,j} = \texttt{SE-Trapdoor}(params, sk_{TTP}, A_{i,j})$ pour $1 \leq j \leq m$.

Ces trappes sont des informations secrètes dans cette instanciation et ne sont révélées à aucun autre noeud.

Le TTP est ensuite considéré hors-ligne.

0.7.2.2 Phase d'exécution

Durant cette phase la source d'un message encode l'en-tête en utilisant la fonction `SE-PEKS` appliquée à chaque attribut et la clef publique de TTP au lieu de la clef publique de la destination. L'encodage de l'en-tete $\mathcal{H}(M) = ||_{j \in L_M} A_{M,j}$ est donc :

$$\mathcal{H}(M') = ENCRYPT\_HEADER(M) = ||_{j \in L_M}(E_j, S_{M',j}),$$

avec, pour chaque $j \in L_M$,

$$S_{M',j} = \texttt{SE-PEKS}(params, pk_{TTP}, A_{M,j}).$$

Un noeud intermédiaire recevant un en-tête chiffré peut alors utiliser la fonction `SE-Test` avec les trappes qu'il a reçues pendant la première phase. En effet pour chaque $j \in L_M$:

$$\texttt{SE-Test}(S_{M',j}, T_{i,j}),$$

retourne :

- 1, si $A_{M,j} = A_{i,j}$,
- 0, si $A_{M,j} \neq A_{i,j}$.

N_i est donc en mesure de reconstituer l'ensemble $L_{M,i}$ et de calculer le ratio de correspondance $p_i(M) = \frac{|L_{M,i}|}{|L_M|}$.

0.7.3 Evaluation

Le schéma original de PEKS a été prouvé sémantiquement sûr contre une attaque à mot-clef choisi par Boneh et al. La nouvelle instanciation de PEKS hérite donc de cette propriété qui signifie en particulier qu'un noeud intermédiaire ne découvre aucune information quant aux attributs pour lesquels il ne possède pas de trappe. Hors les trappes ne sont distribuées par le TTP qu'aux noeuds qui possèdent les attributs correspondants. De plus nous émettons l'hypothèse raisonnable dite de "communauté de confiance" : cette hypothèse stipule que les noeuds ne s'attaquent pas à leur propre communauté. Ainsi les noeuds ne distribuent pas leurs trappes à d'autres noeuds illégitimes.

Par ailleurs la fonction `SE-Test` est publique et permet donc à la source d'encoder les attributs de son choix, même si elle ne les partage pas. La fonction est également probabiliste puisqu'elle fait intervenir une part d'aléatoire. De la sorte, la sortie de cette fonction pour une même entrée change à chaque exécution et rend difficile les tentatives d'analyses de trafic via l'en-tête afin de déterminer la destination d'un message.

Enfin notons que le TTP qui joue un rôle central dans cette instanciation n'est pas requis pendant la communication et est donc adapté au paradigme de tolérance au délai.

Ceci conclue la présentation de la solution destinée à protéger la vie privée de la destination et nous abordons maintenant le problème d'assurance de l'exactitude du ratio de correspondance.

0.8 Exactitude des Calculs Chiffrés dans les Protocoles Basés sur le Contexte

Comme mentionné en section 0.5, un noeud malveillant peut très bien mentir sur son ratio de correspondance afin de détourner le trafic ou réaliser un déni de service. L'objet de cette section est donc de proposer une solution à ce problème en concevant un mécanisme de verification du ratio de correspondance.

0.8.1 Première approche

Le problème d'exactitude des données chiffrées est indépendant de celui de la vie privée mais nous allons exploiter la solution présentée dans la section précédente de façon à lui faire remplir également le rôle de preuve de ratio de correspondance.

L'idée est d'utiliser les fonctions de hachage et la difficulté de trouver la préimage d'une empreinte comme preuve de possession d'un secret.

Nous avons donc un noeud N_i qui envoit a son voisin N_k un en-tête chiffre comme suit :

$$\mathcal{H}(M') = ||_{j \in L_M}(E_j, S_{M',j}) = ||_{j \in L_M}(E_j, \texttt{SE-PEKS}(params, pk_{TTP}, A_{M,j})).$$

Et N_k doit calculer $p_k(M')$ à l'aide de la fonction `SE-Test`.

Observons tout d'abord que les attributs chiffrés sont en fait un couple :

$$S_{M',j} = \left\langle r_{M',j}P, H_3(\hat{e}(H_1(A_{M,j}), r_{M',j}pk_{TTP}))\right\rangle = \left\langle r_{M',j}P, H_3(x_{M',j})\right\rangle,$$

où $x_{M',j} = \hat{e}(H_1(A_{M,j}), r_{M',j}pk_{TTP}) \in \mathbb{G}_2^*$.

Observons ensuite que, dans le cadre de l'exécution `SE-Test`, N_k verifie si

$$H_3(\hat{e}(T_{k,j}, r_{M',j}P)){=}H_3(x_{M',j})$$

ou non. Cela demande de calculer $\hat{e}(T_{k,j}, r_{M',j}P)$ et revient donc à verifier si

$$\hat{e}(T_{k,j}, r_{M',j}P) = x_{M',j}$$

ou non. Cela signifie que si N_k connaît la trappe correspondant à un mot-clef, il peut retrouver la préimage correspondante mais dans le cas contraire cela n'est pas possible à cause des propriétés des fonctions de hachage cryptographiques. En exhibant $x_{M',j}$, N_k prouve donc qu'il possède la trappe $T_{k,j}$ correspondante, sans révéler cette dernière.

Un protocole simple de vérification de ratio est donc le suivant :

1. N_i envoie $\mathcal{H}(M')$ à N_k.
2. Pour chaque $j \in L_{M,k}$, l'ensemble de correspondance entre N_k et M', N_k peut calculer $x_{M',j}$. N_k envoie l'ensemble $\{x_{M',j}\}_{j \in L_{M,k}}$ à N_i.
3. Pour chaque $j \in L_{M,k}$, N_i calcule $H_3(x_{M',j})$ et vérifie qu'il est bien égal au second élément du couple $S_{M',j}$. N_i calcule alors le ratio de correspondance garanti $p_k(M) = \frac{|L_{M,k}|}{|L_M|}$.

Le calcul du ratio s'effectue désormais au noeud N_i et donc N_k ne peut plus tricher sur son ratio. De plus N_k ne révèle pas ses attributs à N_i et conserve donc le contrôle de ses informations privées.

Cette première approche offre donc déjà une solution intéressante au problème de garantie de l'exactitude du ratio de correspondance mais elle a deux limites :

– Du point de vue de la vie privée, s'il est vrai que N_i n'obtient pas les attributs de N_k, il découvre quels sont les attributs partagés entre N_k et la destination (le nom des attributs et non leurs valeurs), ce qui représente une certaine fuite d'information.

- Du point de vue de la performance N_k doit maintenant envoyer une préimage, c'est-à-dire un élément d'un groupe d'ordre q par attribut partagé alors qu'il ne renvoyait que le ratio dans la version originale du protocole.

Afin de palier à ces deux limites, il serait intéressant d'améliorer ce schéma de sorte à ce que N_i puisse verifier le ratio de façon globale et non pas attribut pas attribut. C'est l'objet de la solution épurée suivante.

0.8.2 Solution complète avec les filtres de Bloom

Afin d'améliorer notre solution, nous proposons d'utiliser les filtres de Bloom [Blo70] en tant que représentation compacte d'un ensemble d'éléments, en l'occurrence les $x_{M',j}$. Les filtres de Bloom sont en effet une structure de données probabilistes qui permet d'insérer des éléments grâce à des fonctions de hachage (pas nécessairement cryptographiques) puis de tester si un élément appartient au filtre ou non avec une faible probabilité d'erreur. Leur principal avantage réside dans l'encombrement bien plus réduit d'un filtre de Bloom comparé à l'ensemble des éléments qu'il contient.

Un filtre de Bloom est caractérisé par trois paramètres principaux :

- sa taille (nombre de case du tableau), ϕ,
- le nombre de fonction de hachage utilisé par élément, t,
- le nombre d'éléments qu'il contient, n.

Nous nous intéressons plus particulièrement à une extension des filtres de Bloom que nous appellerons les filtres de Bloom numériques (counting Bloom filter) [FCAB00]. Ces filtres ont été introduits pour supporter le retrait dynamique d'un élément (alors que les filtres de Bloom classiques ne supportent que l'ajout d'éléments). Nous les utilisons pour une tout autre raison qui est que le poids w_{CBF} d'un filtre de Bloom numérique CBF (la somme de ses éléments) est proportionnel à son nombre d'éléments. Un filtre de Bloom dynamique révèle donc la cardinalité de l'ensemble d'éléments ayant servi à le construire. La construction d'un filtre de Bloom dynamique est présentée en Figure 8.1 page 175.

Nous utilisons les filtres de Bloom dans notre protocole de la façon suivante.

La source chiffre l'en-tête du message comme présenté auparavant et construit en plus un filtre de Bloom numérique $CBF_S(M')$ contenant l'ensemble des préimages de l'en-tête chiffré $\{x_{M',j}\}_{j \in L_M}$. Ce filtre sera utilisé comme référence de correspondance.

Supposons maintenant que le message soit arrivé au noeud N_i. Le protocole de communication entre N_i et N_k est alors le suivant :

1. N_i envoie $\mathcal{H}(M')$ à N_k. De plus, N_i informe N_k des paramètres publics (ϕ et t) de $CBF_S(M')$ mais il n'envoie pas $CBF_S(M')$.
2. N_k construit un nouveau filtre de Bloom numérique $CBF_k(M')$ dans lequel il insère l'ensemble $\{x_{M',j}\}_{j \in L_{M,k}}$ des préimages qu'il est en mesure de calculer, en utilisant les même paramètres publics pour le filtre. N_j envoie $CBF_k(M')$ à N_i.
3. N_i verifie la consistence de $CBF_k(M')$ par rapport à la référence de correspondance $CBF_S(M')$ en effectuant les tests suivants :
   - $CBF_k(M') \prec CBF_S(M')$, c'est-à-dire que pour chaque case, la valeur de $CBF_k(M')$ est inferieur à celle de $CBF_S(M')$,

- le poids $w_{CBF_k(M')}$ de $CBF_k(M')$ est un multiple de t.

Si les deux vérifications réussissent, alors N_i valide la réponse de N_k et calcule le ratio de correspondance comme étant $\frac{w_{CBF_k(M')}}{w_{CBF_S(M')}}$.

Etant donné que le poids d'un filtre de Bloom numérique est proportionnel au nombre d'éléments qu'il contient, la dernière étape du protocole donne bien le ratio de correspondance car :

$$\frac{w_{CBF_k(M')}}{w_{CBF_S(M')}} = \frac{t|L_{M,k}|}{t|L_M|} = p_k(M').$$

Cette solution permet donc de calculer le ratio de correspondance de façon globale sans découvrir le nom des attributs partagés entre le voisin et la destination, et est moins encombrante que la solution simple n'utilisant pas de filtres de Bloom. Nous évaluons maintenant la sécurité de cette solution.

0.8.3 Evaluation de sécurité

Il y a deux aspects de sécurité à considérer dans cette solution :

- Le vie privée des utilisateurs : contrairement a la solution proposée en section 0.7, il s'agit ici de protéger la vie privée des voisins (la vie privée de N_k vis-à-vis de N_i) et non de la destination.
- L'assurance de calcul : N_i doit être en mesure de verifier que le ratio de correspondance annoncé par N_k est consistent avec l'en-tête du message et qu'elle correspond au ratio effectif.

Nous analysons ces deux aspects successivement.

0.8.3.1 Respect de la vie privée

La vie privee de N_k est préservée dans notre protocole car il est difficile d'inverser un filtre de Bloom $CBF_k(M')$ et d'en extraire les informations $x_{M',k}$ qu'il contient. Cela provient de deux remarques :

- Les fonctions de hachage $h_1, ..., h_t$ utilisées par le filtre de Bloom ne sont pas bijectives et il y a une multitude d'antécédents possible à chaque empreinte.
- L'ordre dans lequel les fonctions de hachage ont été calculées est perdu une fois le résultat injecté dans le filtre, résultant dans une augmentation de l'entropie.

Au final il est possible en moyenne de trouver $\frac{q}{\phi^t}$ éléments qui produiront le même filtre de Bloom et il n'est pas possible de les distinguer. Ce résultat est une borne inferieure en prenant en compte uniquement les filtres de Bloom contenant un seul élément mais le résultat est encore plus important si on augmente le nombre d'éléments insérés.

Du point de vue d'un attaquant cela signifie qu'il est capable de réduire l'espace des possibilités en entrée de q à $\frac{q}{\phi^t}$ en supposant qu'il dispose d'une méthode efficace pour calculer ces ensembles. En choisissant les paramètres de façon correcte on s'aperçoit que cet ensemble $\frac{q}{\phi^t}$ reste bien trop important pour une attaque systématique (brute force).

0.8.3.2 Assurance de calcul

Le but ici est de verifier qu'un voisin N_k ne puisse produire un filtre faux mais qui serait validé par N_i et amènerait à un ratio de correspondance plus grand que le ratio légitime. Dans la pratique, N_k doit produire ce filtre sans la connaissance du filtre de référence et donc il ne peut faire que des suppositions sur le contenu de ce dernier. Nous avons donc procédé à une analyse probabiliste des caractéristiques d'un filtre de Bloom numérique contenant des éléments inconnus. Au final nous aboutissons au résultat suivant :

Theorem 0.8.1 *Soit M' un message avec un en-tête contenant $|L_M|$ attributs.*

Supposons que le filtre de Bloom numérique a une taille ϕ et utilise t fonctions de hachage.

La probabilité $\mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')]$ de succès d'un adversaire N_k de produire un tableau $CBF_{k_{mal}}(M')$ qui sera validé par N_i et mènera à une augmentation du ratio de correspondance de $p_{k_{mal}}(M')$ est majorée par :

$$\begin{aligned}
\mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')] &\leq \left(1 - e^{-\frac{(1-p_{k_{legit}}(M')|L_M|t}{\phi}}\right)^{w_{CBF_{k_{mal}}(M')}} \\
&\leq \left(1 - e^{-\frac{(1-p_{k_{legit}}(M')|L_M|t}{\phi}}\right)^{t} \\
&\leq \left(1 - e^{-\frac{|L_M|t}{\phi}}\right)^{t}
\end{aligned}$$

Dans le théorème précédent, l'indice *legit* fait référence aux valeurs légitimes des données concernées, et *mal* aux ajouts malveillants. La preuve de ce théorème est relativement longue et nous nous attachons simplement à son explication dans cette partie.

Ce théorème signifie en effet que la probabilité de réussite d'un noeud malveillant décroit exponentiellement avec l'augmentation de ratio que l'adversaire essaie d'introduire. Elle décroit également en fonction du ratio légitime. Cela signifie que la probabilité maximale de réussite s'obtient pour un adversaire ayant un ratio de correspondance de 0 et qui essaie de prétendre à un ratio de $\frac{1}{|L_M|}$, et dans ce cas la probabilité de succès est la dernière présentée dans le théorème.

Par ailleurs, si l'on suppose connu par avance le nombre maximum n_{max} d'attributs dans un en-tête, on peut alors obtenir un compromis entre niveau de sécurité et performance intéressant : le degré de sécurité augmente exponentiellement lorsque les performances diminuent linéairement. Cela nous permet d'établir une stratégie pour fixer les paramètres du filtre :

1. Choisir le nombre maximum d'éléments qui vont être insérés n_{max},
2. Choisir un paramètre de sécurité t de telle sorte que la probabilité $\mathcal{P}_{adv}$ de succès d'un adversaire soit majorée par 2^{-t},
3. Fixer la taille du filtre à $\phi = \left\lceil \frac{n_{max}t}{\ln(2)} \right\rceil$.

Cette stratégie garantie que la probabilité de succès d'un adversaire sera bornée par 2^{-t} dans le cas de l'attaque la plus bénigne. Cette dernière n'aura sans doute aucun impact sur le protocole, et si l'on considère les attaques plus dévastatrices la probabilité de succès chute jusqu'à $2^{-tn_{max}}$.

Cela signifie qu'il n'est pas nécessaire de considérer des valeurs de t trop importante et nous détaillons ce point dans l'évaluation globale des solutions de sécurité dans la section suivante.

0.9 Evaluation Globale des Mécanismes de Sécurité Proposés pour les Protocoles Basés sur le Contexte

Dans les précédentes sections nous avons présenté des mécanismes qui répondent à trois besoins dans les communications basées sur le contexte :

- La confidentialité bout-en-bout du contenu,
- Le respect de la vie privé de la destination,
- L'assurance de l'exactitude du calcul du ratio de correspondance (avec des données chiffrées) qui respecte la vie privée des voisins.

Dans cette section nous présentons une étude de la combinaison de ces mécanismes et des synergies qu'ils présentent, puis nous décrivons quelques extensions possibles.

0.9.1 Etude de performance

0.9.1.1 Stockage

Chaque noeud doit stocker les paramètres globaux du système ainsi que les secrets qui lui sont propres : les clefs privées et les trappes. En mutualisant les entités tierces de confiance requises pour les solutions de confidentialité et respect de la vie privée, on s'aperçoit que les clefs privées et les trappes correspondent aux mêmes quantités et qu'il n'est donc pas nécessaire de stocker deux séries de données secrètes mais une seule, qui correspondent au total à qm bits.

Quant au TTP il ne doit stocker que sa paire de clef privée (pk_{TTP}/sk_{TTP}).

0.9.1.2 Impact sur la communication

Nous considérons ici l'impact de nos solutions sur la taille des messages durant la phase d'exécution.

La taille de l'en-tête des messages est proportionnelle au nombre d'attributs qu'elle contient avec ou sans solution de sécurité. Notre solution d'encodage modifie simplement la taille de chaque attribut qui devient $2q$ bits et donc fait croitre la taille d'un facteur constant. Quant à la solution de confidentialité son impact est négligeable.

Concernant la solution visant à garantir l'exactitude des ratios elle fait appel à des filtres de Bloom qui sont des tableaux de taille ϕ. Pour estimer l'encombrement induit par ces tableaux il faut évaluer la taille de chaque case du tableau que l'on note β. Cette taille doit être aussi petite que possible pour réduire l'encombrement, mais elle doit être suffisamment grande pour éviter que la valeur dans la case ne dépasse la valeur maximale ($2^\beta - 1$) ce qui conduirait à une perte des propriétés du filtre de Bloom numérique. Nous montrons qu'en prenant $\beta = 4$, la probabilité d'un dépassement est négligeable, et la taille des filtres de Bloom est donc 4ϕ bits.

0.9.1.3 Charge de calcul

Du point de vue du calcul les opérations les plus couteuses sont celles demandant des couplages bilinéaires sur des courbes elliptiques : il en faut une par chiffrement ce contenu (peu importe le nombre d'attributs) et une par attribut pour l'encodage de l'en-tête. Il en faut également une par évaluation de la fonction `SE-Test`. Le cout de chaque évaluation est comparable à celui d'un déchiffrement RSA. Pour les textes de petites tailles (comme les attributs) ce cout n'est pas rédhibitoire, en revanche il est problématique pour le chiffrement du contenu. Dans ce cas, il est intéressant de chiffrer le contenu avec une clef secrète et un algorithme de chiffrement symétrique tel AES, et de ne chiffrer que la clef avec la méthode que nous proposons.

Enfin concernant le coût de la solution de garantie de ratio, elle requiert simplement $|L_M|t \leq n_{max}t$ hachage ce qui représente un cout negligeable.

0.9.1.4 Exemple numérique

Nous considérons un scenario ou chaque noeud a $m = 100$ attributs et où le nombre maximum d'attribut par en-tête est $n_{max} = 20$. En considérant une courbe elliptique de degré MOV 2, et avec $q = 512$ bits on obtient une sécurité équivalente à RSA 1024 bits. Dans ce cas le stockage des données requiert environ 50 Kbits ce qui est raisonnable au vu de la capacité de stockage des téléphones actuels.

En ce qui concerne la taille de l'en-tête elle est de 128 bits sans la solution d'encodage et environ 1 Kbit avec, soit 8 fois plus. Cependant avec un choix approprié de paramètres (dans le cas ou l'encombrement en communication est primordial) il est possible de descendre à un facteur de 2.5 seulement (au détriment de la vitesse de calcul).

Pour ce qui est de l'encombrement de la solution d'assurance de calcul, nous avons déjà défini n_{max} et devons encore définir ϕ et t. t est un paramètre de sécurité mais qui peut être choisi relativement petit. En effet si l'on choisi $t = 10$ alors la probabilité de succès de la plupart des attaques est négligeable (voir tableau 9.1 page 206). La taille ϕ du filtre est alors 289 selon la formule que nous avons etabli. Au total la taille des filtres de Bloom excède légèrement 1 Kbit et est donc réellement négligeable en comparaison des autres éléments du protocole.

0.9.2 Extensions

Nous abordons maintenant quelques extensions possibles des mécanismes introduits.

0.9.2.1 Révocation

Le problème de la révocation est difficile de façon générale dans les schémas non interactifs, et encore plus dans le cas des réseaux opportunistes.

Pour palier à ce problème nous proposons d'introduire des fenêtres de temps glissantes appelées époques. Etant donné que le réseau doit être tolérant au délai, chaque noeud doit garder les informations relatives à trois époques consécutives afin d'être en mesure de communiquer. Chaque noeud doit aussi contacter le TTP au moins une fois par époque.

0.9.2.2 Protection envers un TTP malveillant

Le TTP joue un rôle central dans nos protocoles et il est supposé être de confiance. Il est cependant légitime d'envisager le cas d'un TTP malveillant et son impact sur la confidentialité et le respect de la vie privée.

Afin d'éviter les abus d'un tel TTP qui concentre trop de pouvoir, nous proposons d'introduire de multiples TTP, chacun donnant une parties des données secrètes requises par les noeuds pour communiquer.

Nous montrons que cette nouvelle approche est compatible avec nos solutions et qu'elle répond bien au problème de TTP malicieux car la confidentialité et la vie privée des noeuds est préservée à partir du moment où au moins un TTP est honnête.

0.9.2.3 Hiérarchiser les attributs

Une dernière extension est la possibilité d'accorder des poids différents aux attributs selon leur importance.

Notre solution de sécurité s'adapte sans problème à cette extension grâce a l'utilisation de filtres de Bloom numériques. Il suffit alors d'incrémenter les positions du filtre de Bloom du poids de l'attribut au lieu de 1 à chaque évaluation de fonction de hachage.

0.9.3 Conclusion

Dans cette seconde partie, nous nous sommes concentrés sur l'analyse des questions de sécurité dans les mécanismes de transmission basés sur le contexte. Nous avons étudié les problèmes de confidentialité du contenu, de la vie privée des utilisateurs et les exigences d'assurance de calcul dans ce type de protocoles et défini les primitives de sécurité requises pour effectuer la transmission sécurisée basée sur le contexte au sein des communautés de confiance. Ces primitives nécessitent l'utilisation de fonctions publiques soigneusement choisies pour assurer à la fois la vie privée et les opérations de transmission.

Nous avons ensuite présenté une solution originale pour résoudre les problèmes de confidentialité et de vie privée qui est dérivée du chiffrement basé sur l'identité et du chiffrement avec recherche de mot-clef. L'utilisation du chiffrement basé sur l'identité avec des attributs multiples assure la confidentialité du contenu de bout-en-bout sans gestion de clef de bout-en-bout, tandis que l'utilisation spécifique de PEKS permet aux noeuds intermédiaires de découvrir les correspondances entre leur profil et le contexte du message, tout en préservant la vie privée des utilisateurs sous l'hypothèse de communautés de confiance. Les fonctions de chiffrement sont protégées contre les attaques dictionnaire et protège de l'analyse du trafic grâce à l'utilisation d'un nombre aléatoire interne. La solution s'appuie sur un TTP hors-ligne.

Préserver la vie privée par le biais de calcul sur les données chiffrées ne garantit cependant pas l'exactitude des données calculées. Nous avons donc défini un mécanisme supplémentaire qui garanti l'exactitude du ratio de correspondance revendiqué. La conception de ce schéma prend également en compte les exigences de confidentialité. Ce mécanisme d'assurance de calcul est basé sur les préimages de fonctions à sens unique pour la partie assurance, et sur des filtres de Bloom numériques pour la protection des données privées et les aspects liés à la performance.

Nos solutions sont adaptées aux réseaux opportunistes parce qu'elles induisent un coût de stockage et de calcul relativement bas et qu'elles s'appuient sur un TTP hors-ligne qui n'est pas requis pour l'exécution correcte du protocole lors de la communication.

0.10 Confidentialité et Respect de la vie Privée dans les Protocoles Basés sur le Contenu

Nous nous intéressons maintenant à une autre grande catégorie de protocoles, celles des protocoles basés sur le contenu. Dans ce type de protocole il n'y a plus de destination définie a priori, ni explicitement, ni implicitement. Les messages sont simplement publiés et acheminés vers les noeuds qui y sont intéressés.

Du point de vue des solutions de sécurité, la grande différence entre protocoles basés sur le contexte et protocoles basés sur le contenu réside dans le fait que le contexte est intrinsèquement lié à un utilisateur via son profil (et on peut alors définir des clefs privées liées au profil) tandis que les intérêts en matière de contenu n'ont pas de liens directs avec les utilisateurs et peuvent être modifiés complètement. Il faut donc des solutions de sécurité plus dynamiques dans le second cas.

Dans cette section nous présentons les problèmes de sécurité dans les communications basées sur le contenu et apportons une solution au problème de routage (et pas seulement transmission) sécurisé dans le modèle 4, en prenant en compte les contraintes liées au caractère opportuniste.

0.10.1 Modèle de communication

Classiquement, les protocoles bases sur le contenu considèrent trois types d'utilisateurs :

- les utilisateurs finaux :
  - Les éditeurs, qui publient des informations sous forme de notifications d'évènements. Ces évènements sont composés de deux parties : un attribut de routage et des données.
  - Les abonnés, qui expriment leurs intérêts pour un certain contenu via des filtres de souscriptions.

- Les noeuds intermédiaires, qui disséminent le contenu des éditeurs vers les abonnés et construisent des tables de routage basées sur les intérêts des abonnés.

Dans les réseaux opportunistes tous les noeuds sont égaux, et tous doivent donc être capables d'assumer les trois rôles à la fois.

Pour les réseaux opportunistes, l'intérêt de la communication basée sur le contenu provient des propriétés suivantes :

- le découplage entre éditeurs et abonnés,
- l'asynchronicité du processus de publication et diffusion,
- l'efficacité de la diffusion qui permet de toucher un grand nombre d'utilisateurs rapidement.

Le rôle de l'éditeur est spécifique et même si nous considérons que tous les noeuds doivent pouvoir assumer le rôle d'éditeur, nous considérons également que le réseau peut être vu comme un arbre enraciné au niveau de l'éditeur (du point de vu de chaque noeud).

Nous pouvons donc naturellement définir pour chaque noeud son parent et son l-ième parent ainsi que l'ensemble de ses fils et l-ieme petit-fils vis-à-vis d'un éditeur, que nous notons respectivement pour le noeud i : $Par(i)$, $Par^l(i)$, $Chd(i)$ et $Chd^l(i)$. Enfin le voisinage $\mathcal{N}^l(i)$ du noeud N_i à l'horizon l est constitué de l'ensemble des enfants et parents de degré au plus l.

Ces notations sont illustrées avec l'exemple de la Figure 10.1 page 219.

0.10.2 Problématique de sécurité

Nous nous intéressons principalement aux problèmes de confidentialité et de respect de la vie privée. En la matière, les abonnés souhaitent accéder au contenu de leur choix sans révéler leurs intérêts à quiconque. Il est donc important pour eux de chiffrer leurs filtres de souscription. De façon symétrique, les notifications d'événement doivent elles aussi être chiffrées. Ce chiffrement ne doit toutefois pas entraver le routage des données. Les noeuds intermédiaires (donc potentiellement tous les noeuds) doivent pouvoir router les données chiffrées et pour ce faire il y a donc deux aspects importants :

- La construction de tables de routages sécurisées basées sur des filtres de souscription chiffrés. Ces tables doivent être optimisées : si deux filtres chiffrés se réfèrent au même intérêt ils doivent être fusionnés.
- La capacité à faire correspondre une notification d'événement chiffrée avec les informations contenues dans les tables de routages chiffrées afin de prendre la bonne décision de routage.

A cet effet nous définissons donc quatre primitives de sécurité :

- **ENCRYPT_FILTER :** utilisé par un noeud pour chiffrer ses filtres de souscription. Prends en entrée un filtre de souscription, des clefs de chiffrement et renvoi le filtre de souscription chiffré.
- **ENCRYPT_NOTIFICATION :** utilisé par un éditeur pour chiffrer les notifications d'évènements. Prend en entrée une notification et des clefs de chiffrement et renvoi la notification chiffrée.

– **SECURE_LOOK_UP :** permet à un noeud de décider si une notification chiffrée correspond à un filtre de souscription chiffré dans sa table de routage. Cette primitive ne retourne que le résultat booléen de l'opération de correspondance.
– **SECURE_TABLE_BUILDING :** permet à un noeud de construire une table de routage et de comparer deux filtres de souscription chiffrés. Comme la précédente primitive, cette dernière ne doit retourner que le résultat booléen de l'opération de correspondance et ne doit pas révéler d'autres informations à propos des filtres de souscription.

0.10.3 Protocole de routage basé sur le contenu sécurisé

Dans cette section nous présentons notre solution de routage basé sur le contenu qui préserve la vie privée des utilisateurs, et en particulier la façon dont nous implémentons les quatre primitives présentées auparavant.

0.10.3.1 Chiffrement commutatif multiple

L'idée fondamentale de notre solution est d'utiliser un système de chiffrement commutatif à couches multiples. En effet les noeuds n'ont pas de connectivité bout-en-bout et ne peuvent donc pas établir de clef de bout-en-bout (entre éditeur et abonné), car cela entrerait en conflit non seulement avec la nature tolérante aux délais des réseaux opportunistes mais aussi avec le découplage entre les utilisateurs finaux inhérent au routage basé sur le contenu.

En revanche il est tout à fait envisageable pour un noeud de connaître son voisinage proche et d'établir des clefs avec ses voisins, puis de chiffrer ses filtres de souscriptions avec ces clefs. Ceci étant, si les filtres ne sont chiffrés qu'avec une seule clef, le voisin qui connaît cette clef sera à même de déchiffrer le filtre et de menacer sa confidentialité. Il faut donc chiffrer le message avec plusieurs clefs de sorte à ce qu'aucun noeud ne puisse déchiffrer le message seul.

Par ailleurs les messages doivent être routé et donc transmis à des noeuds hors du voisinage de l'abonné. Ces nouveaux noeuds ne partagent aucune clef avec l'abonné et se retrouveraient donc avec du contenu inutilisable. Il faut donc pouvoir retirer une couche de chiffrement avant d'envoyer le message hors du voisinage, en l'occurrence la première couche. Cela suppose donc de pouvoir retirer une couche située sous de multiples autres couches sans détruire ces dernières. Cela n'est réalisable que si le chiffrement est commutatif dans le sens où, pour une donnée quelconque d et deux clefs k_1, k_2 :

$$\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(d)) = \mathcal{E}_{k_1}(\mathcal{E}_{k_2}(d)),$$

où $\mathcal{E}$ représente la fonction de chiffrement.

Nous nous baserons donc sur ce principe de chiffrement commutatif à couches multiples pour définir notre solution, et nous abordons dans un premier temps la question du choix du système de chiffrement.

0.10.3.2 Choix du système cryptographique commutatif

Afin de proposer une solution crédible nous nous sommes intéressés à l'existence pratique de systèmes de chiffrement commutatifs. Les systèmes de chiffrement symétriques sont en général non-commutatif (c'est une propriété négative par défaut) à l'exception du chiffrement avec masque jetable (one-time pad). Cette méthode de chiffrement est très efficace d'un point de vue performance mais elle requiert de changer les clefs à chaque chiffrement et n'est donc pas adaptée à notre problème.

Dès lors nous nous tournons vers les systèmes de chiffrements asymétriques. Ces derniers sont principalement basés sur des exponentiations qui sont donc commutatives a priori. Ceci étant, les systèmes de chiffrement sont probabilistes ce qui change la donne et enlève la commutativité de ces systèmes. Les systèmes de chiffrement commutatifs sont donc plutôt rare en pratique.

Nous avons cependant réussi à trouver un système qui correspond à tous nos besoins à savoir le système de chiffrement de Pohlig-Hellman [PH78]. Ce systeme est bien particulier puisqu'il s'agit d'un système de chiffrement asymétrique à clefs privées (il requiert une paire de clefs mais les deux doivent rester secrètes). Il est défini par quatre éléments :

- q, un nombre premier de grande taille connu par tous les noeuds (il s'agit d'un paramètre du système).
- $\mathcal{K}$, un algorithme de génération de clefs, il retourne une paire de clefs (k_i, d_i) de telle sorte que $k_i d_i \equiv 1 \mod (q-1)$;
- $\mathcal{E}(q, k_i, x)$, la fonction de chiffrement qui retourne $x^{k_i} \mod q$;
- $\mathcal{D}(q, d_i, y)$, la fonction de déchiffrement qui retourne $y^{d_i} \mod q$.

Il est alors aisé de constater que ce système de chiffrement est bien commutatif.

Dans la suite de cette section nous désignerons la clef partagée entre les noeuds N_i et N_j indifféremment par $(k_{i,j}, d_{i,j})$ ou $(k_{j,i}, d_{j,i})$.

0.10.3.3 Description du protocole

Nous pouvons maintenant décrire le fonctionnement du protocole s'appuyant sur plusieurs couches de chiffrements. Le nombre de couches lr est un paramètre de sécurité que nous discuterons dans la section suivante.

0.10.3.4 Traitement des filtres de souscription

Les filtres de souscription émis par les abonnés sont d'abord chiffrés en utilisant lr couches de chiffrement correspondant aux lr noeuds suivants c'est-à-dire les lr parents. Ainsi un noeud N_i chiffre un filtre de souscription f de la manière suivante :

$$\begin{aligned} ENCRYPT\_FILTER(f, k_{i,Par(i)}, ..., k_{i,Par^{lr}(i)}) &= \underbrace{\mathcal{E}(q, k_{i,Par^{lr}(i)}, \mathcal{E}(...\mathcal{E}(q, k_{i,Par(i)}, f)))}_{lr \texttt{ couches}} \\ &= f^{k_{i,Par^{lr}(i)} \dots k_{i,Par(i)}} \mod q \end{aligned}$$

Les noeuds intermédiaires qui reçoivent un filtre chiffré peuvent enlever une couche de chiffrement (la première). Ainsi un noeud N_j qui reçoit un message en provenance de N_i avec $j = Par^{lr}(i)$, possède la clef $k_{i,j}$ et peut donc retirer une couche de la façon suivante :

$$\mathcal{D}(q, k_{i,j}, \underbrace{f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)}\dots k_{i,j}}}_{lr \texttt{ couches}}) = \underbrace{f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)}\dots k_{Par(i),Par(j)}}}_{lr-1 \texttt{ couches}} \mod q.$$

Il construit ensuite sa table de routage avec ce filtre chiffré avec $lr-1$ couches : soit ce filtre existe déjà soit il l'ajoute dans une nouvelle ligne.

Ensuite N_j chiffre le filtre avec une nouvelle couche avec la clef correspondant à son lr-ième parent $N_{Par^{lr}(j)}$ à savoir $k_{j,Par^{lr}(j)}$ et envoie le résultat à son parent immédiat $N_{Par(j)}$.

0.10.3.5 Diffusion des notifications d'évènements

Lorsqu'un éditeur N_p veut notifier un évènement il doit également le chiffrer avec lr couches correspondant au lr prochains noeuds.

Ainsi, si N_i vérifie $i \in Chd^{lr}(p)$ alors N_p chiffre l'évènement (ra, pld) de la façon suivante :

$$ENCRYPT\_NOTIFICATION(ra, pld, k_{p,i}, \dots k_{p,Par^{lr-1}(i)}) = [en_1, en_2, en_3],$$

où :

$$en_1 = ra^{k_{p,i}\dots k_{p,Par^{lr-1}(i)}} \mod q,\ en_2 = pld^{k_{p,i}\dots k_{p,Par^{lr-1}(i)}} \mod q,\ en_3 = N_p.$$

Le message est ensuite envoyé au fils $N_{Par^{lr-1}(i)}$ de N_p.

Lorsque le message arrive au niveau d'un noeud intermédiaire N_j, ce dernier retire une couche de chiffrement et identifie l'évènement chiffré avec $lr-1$ couches avec les filtres de souscriptions également chiffrés avec $lr-1$ couches dans sa table de routage, pour prendre une décision de routage. N_j chiffre alors l'evennement avec une nouvelle couche correspondant au lr-ième enfant déterminé par la table de routage et transmet le message.

Ce protocole est difficile à expliquer succinctement de façon formelle, et la meilleure façon de visualiser de façon pratique le déroulement du protocole est de suivre l'exemple donné en section 10.6 qui se base sur la Figure 10.3 page 242.

0.10.4 Evaluation

Le protocole proposé utilise de multiples couches de chiffrements et il est au moins aussi sur qu'un protocole n'utilisant qu'une seule couche de chiffrement comme prouvé par Bellare et al. dans [BBM00].

L'utilisation de multiples couches avec des clefs appartenant à différents voisins permet de plus de garantir qu'aucun noeud n'ait accès en clair à un message, l'opération de correspondance entre notification d'évènement et table de routage se fait avec des données chiffrées. De plus, contrairement aux solutions existantes comme [RR06, SL07], étant donné que le protocole ne requiert pas de clefs de groupes pour les différentes catégories d'utilisateurs, tous les noeuds peuvent assumer tous les rôles à la fois et ce protocole est donc bien adapté pour le routage basé sur le contenu dans les réseaux pair-à-pair en général et les réseaux opportunistes en particulier. Pour la même raison, cette solution est la première à maintenir un découplage complet entre les utilisateurs finaux.

Au niveau de la performance on remarque que les calculs sont essentiellement concentrés au niveau des utilisateurs finaux qui doivent ajouter lr couches de chiffrements et donc réaliser lr exponentiations. Les noeuds intermédiaires eux n'ont qu'un déchiffrement et un chiffrement à effectuer soit deux exponentiations. Le coup d'une exponentiation est similaire au coup d'une opération de type déchiffrement dans RSA.

En fait le paramètre lr joue un rôle central dans ce protocole. Du point de vue de la sécurité, le protocole est sûr, au sens où il protège la vie privée des utilisateurs, tant que $lr-1$ noeuds consécutifs au plus s'entraident de façon malveillante. Ainsi plus lr est élevé, plus la sécurité apportée par ce protocole est grande. Mais lr a aussi un impact sur les performances et sur les clefs à établir dans le voisinage : le voisinage en question correspond à tous les noeuds distants de moins de lr sauts, et donc plus lr est grand plus le voisinage et le nombre de clefs à établir est grand lui aussi. A ce sujet nous avons jusqu'à présent supposé que les clefs étaient simplement disponibles pour les utilisateurs et nous décrivons dans la section suivante un protocole d'établissement de ces clefs.

0.11 Associations de Sécurité dans les Protocoles Basés sur le Contenu

Le protocole présenté dans la section précédente s'appuie fortement sur des clefs partagées avec le voisinage à lr sauts. Ce voisinage doit donc être découvert de façon exacte pour ensuite permettre un établissement de clef basé sur la topologie locale. Nous commençons donc par analyser les besoins de sécurité d'un tel protocole de gestion de clef.

0.11.1 Analyse des besoins de sécurité

La gestion de clef est un défi majeur des réseaux opportunistes. En effet l'absence de connectivité bout-en-bout implique qu'il est impossible d'établir des associations de sécurité de bout-en-bout. Nous avons déjà vu également que les clefs publiques classiques étaient inadaptées dans ce contexte, et que les systèmes de chiffrement basés sur l'identité pouvaient être une bonne alternative lorsque l'identité de la destination est connue. Or ce n'est pas le cas dans les communications basées sur le contenu qui garantissent le découplage entre utilisateurs finaux. La seule solution viable reste donc d'envisager une gestion de clef locale (à une distance telle que la connectivite puisse être assurée) et auto-organisante.

Par ailleurs, la sécurité du chiffrement à couches multiples se base fortement sur une vision correcte de la topologie locale et requiert donc un mécanisme de découverte du voisinage qui soit sécurisé. Toutefois pour découvrir le voisinage de façon sécurisée il est nécessaire d'avoir des clefs préétablies et on se retrouve avec un cycle de dépendance entre découverte de voisnage et établissement de clefs. Pour briser ce cycle de dépendance nous proposons d'effectuer ces deux opérations simultanément.

Une attaque particulièrement efficace dans ce contexte est l'attaque Sybil, dans laquelle un noeud prend plusieurs identités et prétend être à différentes positions. Un tel noeud peut alors établir des clefs à différents niveaux avec sa victime et peut alors déchiffrer plusieurs couches à lui seul. Il est donc important de se protéger contre ce type de menace, et nous proposons donc une solution en deux étapes, la première étape visant simplement à éradiquer la menace Sybil.

0.11.2 Protocole d'établissement de clef et de découverte sécurisée du voisinage

0.11.2.1 Phase d'organisation

Au cours de cette phase, les noeuds contactent une entité tierce de confiance appelée gestionnaire d'identité (Identity Manager IM). Cette dernière fourni à chaque noeud un pseudonyme unique et certifié dont le but est double :

- Protéger la vie privée des utilisateurs en évitant de révéler leur identité,
- Empêcher les noeuds de lancer des attaques Sybil.

Pour cela chaque noeud N_i génère une paire de clefs pk_i/sk_i. Il contacte ensuite le gestionnaire d'identité qui requiert des informations I_i à propos de N_i. Suite à cela, l'IM utilise une clef secrète K connue de lui seul pour générer le pseudonyme $\mathcal{P}_i$ via une fonction de hachage à clef de la façon suivante :

$$\mathcal{P}_i = MAC(K, I_i).$$

Ce pseudonyme est unique car les informations requises par l'IM sont toujours les même et identifient le noeud. L'IM génère alors un certificat de pseudonyme $\mathcal{C}_i$ en signant la clef public de N_i :

$$\mathcal{C}_i = \{\mathcal{P}_i, pk_i, signature_{sk_{IM}}(\mathcal{P}_i, pk_i)\}.$$

Cette étape est illustrée plus en détail sur la Figure 11.1 page 247. On note qu'un noeud peut obtenir plusieurs certificats en présentant à chaque fois des clefs publiques différentes, mais tous ses certificats contiendront le même pseudonyme.

L'IM n'est ensuite plus requis pour la phase suivante de découverte de la topologie locale et d'établissement d'associations de sécurité a proprement dit.

0.11.2.2 Phase d'exécution

Pendant la phase d'exécution, un noeud N_S souhaite découvrir son voisinage à lr sauts et établir des clefs avec ses voisins. Il s'agit donc d'un protocole de communication particulier au cours duquel il faut garantir la distance parcourue par le message. Nous proposons une approche inspirée de certains protocoles de routage sécurisé dans les réseaux ad-hoc MANET, qui consiste à signer le message à chaque étape : les signatures encapsulées interdisent ainsi la modification d'une étape du message sans affecter la vérification de l'intégralité de celui-ci.

Dans le même temps, le protocole d'établissement de clefs est une version légèrement modifiée du protocole de Diffie-Hellman sécurisé appelé STS [DVOW92].

Plus precisemment le noeud N_S initie un message de requête d'association qui a la forme générique suivante :

$$< SARq, remaining\_hop\_count, Certificate\_list,$$
$$DH\_share\_list, signature\_list > .$$

où :

- $SARq$ est un identifiant du message et signifie qu'il s'agit d'une requête,
- $remaining\_hop\_count$ est un compteur qui mesure la distance restant à parcourir,
- $Certificate\_list$ est une liste comprenant les certificats de tous les noeuds que le message a traversés,
- DH_share_list est une liste de portions du protocole Diffie Hellman,
- $signature\_list$ est la liste des signatures du message à chaque étape.

Lorsqu'un noeud N_i reçoit ce genre de message, il vérifie la validité de la première signature, décrémente le compteur $remaining\_hop\_count$ et complète les listes $Certificate\_list$, DH_share_list et $signature\_list$. La signature s'appuie sur un nombre aléatoire qui n'est pas révélé à cette étape.

Le message est transmis tant que $remaining\_hop\_count$ est supérieur à 0. A ce moment une réponse est générée et renvoyée vers N_S via le même chemin qu'à l'aller. La réponse est de la forme

$$< SARp, Certificate\_list, DH\_share\_list, signature\_list,$$
$$random\_number\_list > .$$

où $random\_number\_list$ révèle les nombres aléatoires qui ont été utilisés dans la requête.

Lorsque la réponse atteint la source N_S, cette dernière vérifie que toutes les étapes du protocole ont été respectées en vérifiant toutes les signatures de la liste $signature\_list$ et, si tout est correct, elle calcule les clefs partagées. Une description plus formelle du protocole est présentée en section 11.3.2 page 248 avec notamment un exemple présenté dans le tableau 11.1 page 249. Nous évaluons la sécurité et les performances du protocole dans la section suivante.

0.11.3 Evaluation

Du point de vue de la sécurité, il faut d'abord remarquer que le gestionnaire d'identités n'a pas pour rôle de certifier l'identité d'un noeud en liant cette identité à une clef publique unique, mais simplement de donner à chaque noeud un pseudonyme unique. Cela signifie que l'IM n'a pas besoin de garder en mémoire les certificats qu'il a délivré et est donc une entité légère et hors-ligne. Les certificats de pseudonyme sont simplement utilisés comme ancres de départs pour le déroulement du protocole et garantissent simplement qu'un noeud ne peut prétendre être à deux positions différentes dans une même requête disqualifiant de la sorte les attaques Sybil.

Par ailleurs, les mécanismes de signatures encapsulées permet de garantir la distance des noeuds à la destination (distance logique) et garantit que la requête et la réponse suivent le même chemin en sens inverse grâce à l'introduction des nombres aléatoires révélés uniquement au retour.

De plus le protocole est également protégé contre les attaquants passifs, ainsi que les attaquants au milieu (Man-In-the-Middle). Le mécanisme d'établissement de clefs se base sur le protocole STS, et l'authentification qui y est associée n'a pas pour but de réellement lier une clef à une identité mais simplement de garantir que le même noeud n'apparaît pas deux fois. Le but du protocole est, pour la source, de déterminer qu'elle partage une clef à distance tel nombre de sauts avec un noeud unique, peu importe le noeud. Les critiques du protocole STS qui ont amené au déploiement du protocole SIGMA pour contrer les attaques de mauvaise-liaison (misbinding) ne sont donc pas pertinentes pour notre protocole.

Enfin du point de vue de la performance, le protocole s'appuie sur des signatures et présente donc un coût non négligeable, mais il faut garder a l'esprit que ce protocole n'est requis que pour établir les premières associations de sécurité, et que les clefs peuvent être ensuite mises à jour de façon efficace. De plus nous avons pris en compte les paramètres de performance dans la conception du protocole, et c'est la raison pour laquelle nous avons introduit les nombres aléatoires : ils permettent en effet de diviser le nombre de signatures requis pour maintenir la sécurité du protocole par deux.

0.11.4 Bilan

L'analyse des caractéristiques des réseaux opportunistes et des communications basées sur le contenu, nous ont amène à la conclusion que la gestion des clés dans ces réseaux devrait être auto-organisée et locale. Cette localité suppose une vue correcte de la topologie du voisinage. Nous avons donc conçu une solution complète qui permet à la fois d'établir des associations de sécurité et de découvrir le voisinage de façon sécurisée.

Cette solution basée sur les certificats de pseudonyme et des signatures encapsulées permet d'établir des clefs entre un noeud et tous ses voisins qui sont à une distance de moins de lr sauts, sans relation de confiance préétablie ou d'infrastructure. La solution permet également la découverte de la topologie du voisinage et résiste à toute manipulation par des noeuds malveillants. Nous avons également proposé l'utilisation d'un gestionnaire d'identité, qui fournit à chaque noeud un pseudonyme unique certifié. Ce gestionnaire léger empêche ainsi efficacement les attaques Sybil. En outre, le gestionnaire est hors-ligne et n'est pas requis pendant la phase d'exécution, donc le schéma de gestion des clés est auto-organisée.

Le schéma proposé peut donc être utilisé comme une ancre au protocole de chiffrement à couches multiples commutatives pour la communication basée sur le contenu dans les réseaux opportunistes, réalisant ainsi la confidentialité et le respect de la vie privée de bout-en-bout sur la seule base d'une gestion au niveau local et auto-organisée des clefs.

Conclusion

Les réseaux opportunistes reposent sur un paradigme de communication très prometteur et plein de défis. Jusqu'à présent l'immense majorité des efforts de recherche dans ce domaine s'est focalisée sur les aspects purement réseaux du problème. Nous sommes les premiers à avoir analysé dans le détail les problématiques de sécurité liées aux communications opportunistes et à avoir proposé un ensemble de solutions couvrant une large gamme de besoins de sécurité, et à les avoir appliqués à une plateforme expérimentale, le projet Haggle.

Nous avons en particulier présenté une solution au problème de confidentialité du contenu et de respect de la vie privée dans les communications basées sur le contexte. Les solutions que nous proposons sont inspirées de mécanismes basés sur les couplages bilinéaires que nous avons adaptés à nos besoins spécifiques. Nous avons de plus identifié le problème nouveau de besoin d'assurance de calcul et avons proposé une solution originale qui exploite en particulier les filtres de Bloom sous un jour nouveau.

Nous nous sommes également intéressés au problème de la confidentialité et du respect de la vie privée dans les communications opportunistes. Nous avons proposé une solution basée sur de multiples couches de chiffrement commutatives qui permettent la construction de tables de routage avec des données chiffrées puis de prendre des décisions de routage basée sur des évènements chiffrés également. Cette solution est la première à permettre le routage basé sur le contenu et respectueux de la vie privée qui maintient le découplage entre les utilisateurs finaux. Enfin nous avons présenté une solution qui permet de découvrir la topologie environnante de façon sécurisée et d'amorcer des associations de sécurité.

De façon générale, le coeur de tous les problèmes que nous avons étudiés est le calcul sur données chiffrées dans des scenarios divers. En la matière, Gentry a annoncé une percée importante : une méthode de chiffrement entièrement homomorphe [Gen09]. Un tel système permet en effet l'évaluation d'un polynôme quelconque sur des données chiffrées, et pourrait être utilisé pour résoudre tous les problèmes liés au calcul sur des données chiffrées. Cependant, la recherche dans ce domaine est loin d'être terminée. Tout d'abord, le schéma proposé par Gentry est un bon résultat théorique, mais est très coûteux. Selon le scénario qui est considéré, il est donc intéressant de concevoir des solutions efficaces qui répondent aux exigences exactes du scénario, et c'était notre but dans cette thèse dans la conception de solution préservant la vie privée. Cette même recherche d'efficacité nous a amené à concevoir un schéma de chiffrement basé sur des identités multiples : les méthodes de chiffrement à base de politiques [BMC06] ou d'attributs [BSW07] auraient pu résoudre le problème de la confidentialité du contenu, mais à un coût plus élevé.

Chapter 1

Introduction

The subject of this dissertation is security in opportunistic communications. Both security and the opportunistic nature of communications raise new and compelling research problems. We thus introduce the concept of opportunistic networks, then stress on security issues raised by opportunistic communications.

1.1 Opportunistic Networks

Communication in its simplest and oldest form is a simple face to face conversation between two persons. People then tried to communicate at distance first by using simple signals (fires, beacons, smoke signals, horns, drums), and, with the invention of writing, by sending written messages carried by couriers and then by pigeons (used by the Romans to aid their military over 2000 years ago, and by Persians even before): these methods are the forerunners of postal systems, which are still in use nowadays.

The invention of the electrical telegraph (1838 AD) increased the speed of communication, and was closely followed by the invention of the telephone (the speaking telegraph, patented in 1876) which enabled transmission of voice over long distances (originally few miles). Concerning images, soon after the invention of television (1927), the world's first public video telephone service was developed by Dr. Georg Schubert in Germany in 1936 and it allowed people to talk and see their correspondent using square displays of 8 inches. The service was interrupted by World War II, and afterwards, despite many other deployments in different countries, the service generally lacked public acceptance.

In the 1980s, after the invention of computers and computer networks, two technologies that affect our daily life and our approach towards communication made their first appearance: the Internet and mobile phones. These technologies have had a lot of developments since, and they have been deployed massively so that they are almost ubiquitous nowadays.

The evolution of communication methods thus seems to follow the Olympic motto, the famous hendiatris *Citius, Altius, Fortius,* which is Latin for "Swifter, Higher, Stronger". Swifter first, because from mails delivered by postmen on vans, bicycles or even on foot, communication evolved to electronic mails or voice conversations delivered through optical fiber networks at (almost) the speed of light. Higher then, which can be taken in the broader sense of farther away, because satellite communications allows to reach any point on the surface of the terrestrial globe, or to communicate with robotic spacecrafts on space exploration missions to the boundaries of the sun system and beyond. Stronger finally, as the rate of data transmission is ever increasing, from dial-up to DSL and optical fibers, or from SMS (Sort Message Service) and WAP (Wireless Application Protocol), to MMS (Multimedia Message Service) and 3G internet access.

This evolution brought the democratization of long-distance communication means, which were historically the privilege of elites and military. Ideally, if pushed to the extreme, this evolution of communication possibilities should allow anybody, anywhere and at any time to access any information, any data, any resource available on any network quickly and at a reasonable cost (for free at best). Unfortunately this dream of uniting humanity through universal communication is deemed to remain utopia for political and economical reasons. Political reasons include the fact that governments want to maintain some control or to monitor communication networks. Economical reasons are even more significant: even though scientific innovation always brings forth better communication means, the associated infrastructures have a high cost, and it is for example completely unthinkable to deploy optical fibers to any house in the world. The cost of deployment is in fact one of the discriminating factors that explains the difference of development between rich and poor countries.

Even though this utopia is out of reach in the near future, the scientific community came up with an exciting new communication paradigm: opportunistic networking. In opportunistic networking, the idea is to overcome the limitations of existing networks by exploiting them all as the opportunity comes. The goal is for a message to reach its destination eventually, in spite of all possible obstacles, by forwarding it hop-by-hop, each hop using the best available option according to its local knowledge of the network to forward the message. The service offered is therefore truly best effort: when no better option is available, messages can still get closer to the destination thanks to mobility in a physical sense, e.g. people carrying their devices and walking or driving from one place to another.

Opportunistic communication thus follows the *store, carry, and forward* principle: when a communicating device receives a message, it stores it, carries it by using physical mobility, and forwards it when a communication opportunity arises. This principle is close to the situation in postal systems: the user carries its letter and drops it in a pillar box, where it is stored. A postman takes the letter out of the pillar box, stores it in a postal truck, carries it to a post office where it is forwarded. The process goes on until the letter is forwarded to the letter box of the destination. Opportunistic networks use the same principle with the notable difference that there is no organization in the forwarding of messages, messages are rather forwarded whenever a communication opportunity arises.

At first glance, this might look as regressing to couriers or traditional postal systems. We argue that evolution does not always follow an ever increasing performance graph, as humorously expressed by the cartoon in Figure 1.1. Furthermore, taking advantage of physical mobility is beneficial in many situations, e.g.:

Mike Keefe The Denver Post 03.27.09
MASS PUBLICATION
MOVABLE TYPE
FIRST WRITTEN WORD
EMAIL
TWITTER
140 CHARACTERS. WHAT MORE IS THERE TO SAY?
TWEET TWEET
THE EVOLUTION OF COMMUNICATION
www.caglecartoons.com

Figure 1.1: A cartoon representing the evolution of communication.

- In many developed cities, users enjoy access to broadband infrastructure in "islands of connectivity" (e.g. home or office), but they are also likely to sporadically be in range of many other users while in between. Opportunistic communication can thus be used to provide network access for users on the move from one "island of connectivity" to another.

- In many regions of the world, remote places have very limited access to broadband communication networks (if any). In that case, opportunistic communication is useful to provide a minimum service and enable communication despite the lack of infrastructure. The same idea can be used in disaster recovery scenarios where the communication infrastructure goes down: opportunistic networking may be the only feasible way to carry important data.

Opportunistic networking is thus a useful novelty in the landscape of communication paradigms, although atypical: the philosophy of opportunistic communication is indeed closer to the second Olympic motto, "The most important thing is not to win but to take part!", in the sense that the cooperation between humans can overcome the limitations of the communication infrastructures. Technically, the ever increasing number of portable devices with wireless technologies that do not require any infrastructure to transmit data hop-by-hop (like bluetooth, or WiFi in ad-hoc mode), shows that this vision of opportunistic networking is realistic and that it can be made available in a not so distant future.

1.2 Security in Opportunistic Networks

Security is an essential part of all communication systems. It encompasses a broad range of areas, from information security to communication and user privacy through reliability and trust establishment.

Information security is a traditional requirement for communications, and its core objectives are stated as confidentiality, integrity (authenticity) and availability. For example, strategic messages were historically carried by couriers. To prevent the courier from tampering with the message, the message was sealed: the seal provided integrity and authenticity. The message was also encrypted (originally through transposition or substitution ciphers) to enforce confidentiality of the message in case the courier was caught by an enemy. These methods have been improved by modern cryptography to provide the same type of services for digital information, through modern encryption systems and digital signatures.

Information security is an even more essential requirement in opportunistic communication, as messages are transmitted through devices that are unknown and hence untrusted by the source and the destination of the message: it is thus important to protect the confidentiality and integrity of messages. However, classical solutions deployed in legacy networks need to be revisited to take into account the specific constraints of opportunistic communication. Indeed, opportunistic communications follow the store, carry, and forward principle which implies further constraints as follows:

- There is no end-to-end connectivity between the source and the destination of the message. This implies in particular that source and destination cannot run interactive security protocols.
- There is no infrastructure, and in particular no security infrastructure (e.g. Public Key Infrastructure) during opportunistic communication. Security solutions should therefore be self-organizing.
- Messages are forwarded through different routes depending on communication opportunities. Nodes are mobile and route disruptions are frequent, hence security solutions should be flexible and dynamic.

The specific constraints of opportunistic networks thus require to revisit the challenges related to information security.

Furthermore, the issues of availability, reliability and trust establishment, also need a thorough rethinking in the light of opportunistic communication. In legacy networks, such issues are typically delegated to the infrastructure, which is trusted to perform networking operations. Rather than data confidentiality or integrity, which are addressed through classical security mechanisms, trust in the context of opportunistic communication pertains to the ability of correctly routing data from source to destination. Opportunistic communication does not assume a routing infrastructure, messages are forwarded by all communicating devices, hence cooperation between devices is crucial. These devices have limited resources though and have a priori no reason to help forwarding data of other users. Incentives are therefore required to enforce cooperation among nodes to enable opportunistic communication.

Privacy is another critical issue since communicating devices are becoming more and more pervasive, and are managing an ever increasing amount of information about users. To solve privacy issues, cryptographic primitives are required but are insufficient: privacy protection also requires specific security architectures and policies. All companies and governmental institutions commit to some privacy policies pertaining to the data they manage, and they are audited with this respect: users only have to trust these policies. A more complex requirement pertaining to communication privacy is to protect users, such that an attacker performing traffic analysis cannot discover the source and destination of a message. This requirement can be achieved through specific architectures or protocols (e.g. onion routing [GRS99]).

The concept of opportunistic communication makes analysis of communication flows unpractical as messages from source to destination take different paths each time due to the mobility of users and frequent topology changes. It is thus sufficient to hide the addresses of the source and destination of messages to ensure communication privacy. However hiding the destination's address of a message hinders communication itself, as nodes take forwarding decisions based on the destination's address. The security mechanism should therefore hide the destination while revealing enough information to take forwarding decisions. One approach to solve this issue is to use dynamical addresses that cannot be linked to a destination, but still provide some information on the location of the destination.

The privacy issue is even more challenging in rich forwarding mechanisms which were specifically proposed to enable communication in opportunistic networks. Routing in opportunistic networks is indeed a compelling issue, and using the address of the destination is not a good strategy for at least two reasons:

- classical forwarding protocols are based on addresses, and applications on top of them require some means of deriving addresses from a names. This is not an easy task in opportunistic networks as access to a naming service (e.g. DNS) cannot be assumed.

- even if the source knows one address of the destination related to a given infrastructure, this address does not have a topological meaning from the perspective of forwarding devices in the opportunistic communication.

To overcome the issues pertaining to the use of addresses in opportunistic forwarding, radically new approaches have been proposed: conversational communication between a source and a destination is replaced by content dissemination where destinations are implicitly defined by their interests or their context rather than an explicit address.

In context-based forwarding, the destination is implicitly defined by its context, such as the destination's working address and institution, the probability of meeting with other users or visiting particular places, or the social community users belong to. The main idea of context-based forwarding is to look for devices that show increasing match with known context attributes of the destination. High degree of match means high similarity between the device's and destination's contexts and, thus, high probability for the device to bring the message in the destination's community (possibly, to the destination). However the destination's context is private information that is related to the user of the device and is therefore more sensitive from a privacy perspective than the network addresses. The same remark applies to the context of devices that forward the message, as the context links the device to the user owning it. It is therefore important to assure the secrecy of the contexts that are disseminated through the network. On the other hand, the encryption mechanism requires devices to be able to compute the match between their context and the destination's context in order to take forwarding decisions. Context-based forwarding and user privacy thus present conflicting requirements.

In content-based routing, there is a complete decoupling between sender and receiver: messages are forwarded based on their content, and the destination is not identified at the time of message transmission. In a content-based communication service, users declare their interests while senders simply publish messages to all interested users; intermediate nodes build their routing tables based on the interests of users and look-up published content in their routing table to forward the content to interested users. The interests of users are private information and it is therefore important to guarantee the confidentiality of interests while still enabling intermediate nodes to build their routing tables. Encrypting the interests is a solution to protect confidentiality but it brings along two difficulties:

- how can intermediate nodes build their routing tables based on encrypted data?
- how can intermediate nodes look up content in encrypted routing tables ?

This issue is very different from the previous one because contrary to context, interests are not intrinsically linked to a node and they change frequently, thus calling for more dynamic solutions.

Opportunistic forwarding, both in its context-based and content-based versions, thus raises an interesting and challenging issue, which is the main focus of our work: how to meet the conflicting requirements between routing and user privacy? Is it possible to encrypt a message and then take forwarding decisions based on the encrypted content?

1.3 Contributions

In this thesis, we first present an overview of opportunistic networks and opportunistic forwarding strategies and classify the latter in three main categories: oblivious forwarding, context-based forwarding, and content-based forwarding (Chapter 2). We then analyze the security issues raised by opportunistic networks (Chapter 3), and we define in particular several privacy models and a trust assumption, which are at the core of the solutions presented in the manuscript. We then present a practical implementation of security primitives in an opportunistic communication architecture based on off-the-shelf cryptographic functions (Chapter 4). This implementation consists of preliminary solutions to ensure secure opportunistic communication with simple hypothesis, and paves the way for more advanced and more innovative solutions taking into account more complex requirements in the next parts.

The second part of the thesis focuses on security in context based-forwarding. We first propose a context-based forwarding model (Chapter 5) and analyze the security issues pertaining to context-based forwarding. We identify three main security requirements which are data confidentiality, user privacy, and Computation assurance. We then propose solutions that meet each of these requirements. In Chapter 6, we propose a solution to payload confidentiality. The solution is based on an extension of identity-based cryptography in a multiple identity setting. We present a dedicated solution and a specific security model for multiple identity setting. We then formally analyze the security of the solution with a reductionist proof of security in the defined model. Concerning user privacy (Chapter 7), we propose a solution to enable intermediate nodes to discover matching attributes between the context of a message and their own context, while preserving the secrecy of non-matching attributes of the message. The solution is based on a searchable encryption scheme applied in an original instantiation making use of an offline Trusted Third Party, and we analyze the security of the scheme in this particular instantiation. We then propose a mechanism to guarantee computation assurance in Chapter 8. The scheme provides proof of correctness of a computation on encrypted data without exposing privacy of the node performing the computation. The proof of correctness is based on the intractability of finding preimages of cryptographically secure hash functions. The scheme is enhanced for privacy and performance reasons with the use of counting Bloom filters that efficiently prevent a malicious node from tampering with the result of the computation on encrypted data. We then evaluate the security of the proposed mechanism through a probabilistic approach. We conclude this part by shedding light on advantages that result from the combination of the three previous solutions and by proposing extensions to meet additional security requirements (Chapter 9).

The third part of the thesis presents a privacy-preserving content-based protocol. We first present our content-based routing model for opportunistic networks and then analyze the security requirements in this model (Chapter 10). We then propose an original solution to preserve privacy of user, and more specifically their interests, which is based on Multiple Layer Commutative Encryption (MLCE). In this solution, interests are encrypted lr times using lr different keys under a commutative cryptosystem (Pohlig-Hellman). The security of the scheme relies on the parameter lr, on the security associations between a node and its lr-hops neighbors, and on the neighborhood topology. The MLCE scheme requires a topology-dependent key management solution. We complete our scheme in Chapter 11, by analyzing the requirements of key management and the threats on such a protocol. We then present a solution to bootstrap security associations along with a secure neighborhood discovery. This key management solution is based on a modified version of the STS protocol combined with encapsulated signatures to prevent tampering with the neighborhood topology. We analyze the performance of this solution and its resilience to classical network attacks.

Part I

Opportunistic Communications: Forwarding Strategies and Security Challenges

Chapter 2

Forwarding in Opportunistic Networks

Opportunistic Networking is an exciting new communication paradigm, that aims at enabling communication in challenged environments. In opportunistic networking, the idea is to overcome the limitations of existing networks by exploiting all available communication opportunities. The goal is for a message to reach its destination eventually in spite of all possible obstacles by forwarding it hop-by-hop, and by exploiting physical mobility as an additional opportunity of bringing messages closer to the destination.

Since opportunistic communication is a rather novel concept, the literature on this topic encompasses several works which use very different hypothesis and definitions of opportunistic networking. In the following, we present the settings that we adopt for opportunistic communication. We then present an overview of existing forwarding approaches that fit in our settings, and present an incremental classification.

2.1 Opportunistic Communication: A New Communication Paradigm

The vision of Opportunistic Networks aims at exploring radically new ways for data communication in a mobile environment. The range of possible applications is very wide (see section 2.1.2) and these applications have quite different characteristics under the common denominator of Opportunistic Networks.

2.1.1 Characteristics

In this section we first identify the main properties of opportunistic networks in order to characterize them and highlight the differences with respect to existing communication paradigms.

2.1.1.1 Mobility

In opportunistic networks mobility is assumed to be the rule rather than the exception. This is justified by the development of handheld devices with various communication capabilities (for example smartphones feature bluetooth, infrared and WiFi interfaces in addition to 3G telephony and Internet access). Mobility gave rise to Mobile Ad-Hoc NETworks (MANETs) which aim at establishing wireless networks between mobile devices in ad-hoc mode (without the aid of any infrastructure). MANET therefore enable communication between an ad-hoc and ephemeral community of users (e.g meeting participants). In MANETs, there are pro-active (e.g. OLSR [CJ03], DSDV [PB94], SEAD [HJP02a]) and reactive (e.g. AODV [PBRD03], ARIADNE [HPJ05a]) routing strategies to enable communication between parties. Yet, MANET protocols are not very flexible with respect to mobility or node failures: if a node moves or fails then all the tables of its neighboring nodes need to be updated in pro-active approaches, while all routes passing through the node under failure need to be recomputed in reactive approaches. All MANET routing approaches work only under the assumption that there exists a stable route between the sender and the receiver of a message to establish a communication and therefore the support for mobility is restricted.

Moreover, wireless transmissions are less reliable than wired ones (error and packet loss rates are significantly higher) and an idea to overcome this drawback is to take advantage of the broadcast nature of the wireless medium. As stated in [LZM+09] opportunistic forwarding improves performance over that of traditional best path routing. For example in the case of a wireless network as presented in figure 2.1, links have a delivery probability indicated on each edge. Traditional routing selecting only one path in unicast mode achieves 20% delivery between source and destination. However by taking advantage of broadcast communication which is inherent to wireless networks it is possible to consider all five neighbors of the source as relays for the message therefore achieving a delivery probability of $1-(1-.2)^5 = 67\%$. Thus, by considering that any relay might have an opportunity to receive and forward a packet, it is possible to drastically increase the end-to-end delivery ratio with an opportunistic forwarding strategy.

Furthermore, the capacity of wireless ad-hoc networks is strongly affected by phenomena like multi-path fading, path loss (distance attenuation), shadowing by obstacles, and interference from other users. Therefore, the capacity tightly depends on the total number of nodes of the network. In a study on the theoretical limit for this capacity [GK00], Gupta and Kumar have found that the throughput achievable per single source-destination pair decays approximately like $1/\sqrt{n}$ in the best case, and like $1/\sqrt{n\log(n)}$ in the worst case, n being the total number of nodes in the network. These results hold strictly true for fixed ad-hoc networks with nodes located in random positions and supposed to be immobile, but these results can also be extended to MANETs where the support for mobility is limited: MANETs can be viewed as momentaneous fixed networks for a period of time, and when nodes move, incuring a modification of the network topology, routes need to be recomputed in the new fixed network in order to resume the communication. According to Gupta and Kumar, the throughput decreases because, increasing the number of nodes

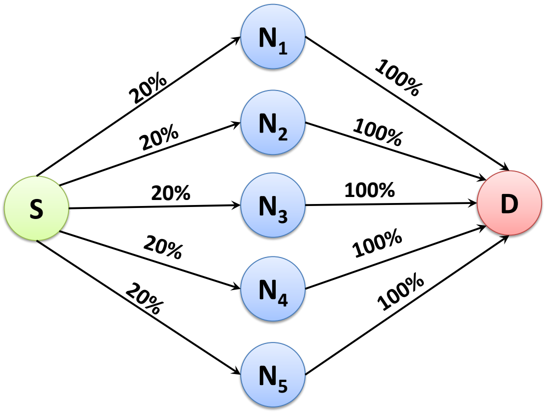


Figure 2.1: An illustration of opportunistic forwarding taking advantage of wireless communication.

involves an increase of concurrent transmissions between source and recipient over long distances, which results in many interferences: the throughput is said to be interference limited. This is a negative result as it implies that MANETs are not scalable.

Opportunistic networking goes beyond the limits of MANETs because the former does not require the existence of an end-to-end path between source and destination nor does it rely on end-to-end routing. Therefore opportunistic networking offers full support for mobility by forwarding data on a hop-by-hop basis and by being tolerant to delay and disruptions.

2.1.1.2 Delay Tolerance

MANETs' limitations come from the lack of flexibility of the proposed proactive and reactive routing algorithms, which aim at establishing a complete path between source and destination to allow communication. When establishing such a path is impossible because source and destination nodes are disconnected due to mobility or because some intermediate nodes entered sleep mode to save energy, MANETs' routing protocols fail and cause the rejection of the transmission: MANETs cannot cope with frequent disruptions due to high mobility or challenged environments.

In order to cope with such disruptions, alternative routing strategies making use of delay as a resource parameter have recently been defined, for example in the Delay-Tolerant Networks (DTN) architecture whose specifications have been proposed by the Internet Research Task Force (IRTF) Delay-Tolerant Networks Research Group(DTNRG) [DTN]. A DTN, following the philosophy of a former project called InterPlanetary Network (IPN), consists of a network of independent Internets (each characterized by Internet-like connectivity), but having only occasional communication opportunities among them. Independent Internets located apart from each other form the so-called DTN regions and a system of DTN gateways is in charge of providing interconnection among them. Hence, in DTNs points of possible disconnections are known in advance and isolated at gateways, which store the data coming from a region and forward it to the other region when a communication opportunity arises, occurring a communication delay, hence the denomination delay-tolerant. The delay can be either controlled in the case where communication opportunities are scheduled or completely random if communication opportunities are random as well. In the latter case, DTNs are considered as best effort networks because there is no guarantee on the waiting time contrary to the Internet where it is possible to estimate a maximum expected delay.

In opportunistic networks, the concept of delay-tolerant communication is taken even further. While DTNs assume the knowledge of DTN regions, which are Internet-like topologies, with interconnections that are only intermittently available, opportunistic networks do not make any a priori assumption on network topology. While routes in DTNs are typically computed via legacy Internet techniques by taking into consideration the link unavailability just as another component of the link cost, such an approach is unpractical in opportunistic networks: in opportunistic networks routes are computed at each hop while the data is forwarded. Thus, each node receiving a message for an eventual destination exploits local knowledge to decide which is the best next hop, among its current neighbors, to reach the eventual packet destination. When no forwarding opportunity exists (e.g., no other nodes are in the transmission range, or the neighbors are deemed unsuitable for the particular communication), the node stores the message and waits for future communication opportunities. As opposed to DTNs, in opportunistic networks each single node acts both as a router and a gateway. Opportunistic networks are therefore more flexible than MANETs and even than DTNs, because they allow each node to trade delay for capacity in terms of getting closer to the destination. In [GT02], Grossglauer and Tse showed that the throughput achievable per single source-destination pair can be kept constant, in case nodes are mobile and communication is delay-tolerant: nodes should only forward packets when a close range communication opportunity appears, and they should store it and wait in the absence of such an opportunity. An extreme strategy to cope with such requirement would be that source and destination wait until they are in direct contact to communicate without the involvement of any intermediate node: such a routing policy is called wait-for-destination. Such a strategy obviously increases the transfer delay a lot because it relies on mobility alone. Opportunistic networking makes use of this policy in a relaxed way, by considering that a communication opportunity arises when it is possible to transmit at close range to an intermediate node which is closer to the destination node with respect to the source node. Therefore, by considering mobility as an opportunity rather than a hindrance, delay tolerance increases the capacity of opportunistic networks far beyond MANETs.

2.1.1.3 Disseminational Communication

Opportunistic networks are very flexible environments that aim at taking advantage of all communication opportunities to reach a destination. In order to benefit from various communication architectures, all the networking information and in particular addressing information of packets created to take advantage of opportunistic networking should be available at a high level of abstraction, which can be assimilated to application layer in the Internet, and should not be dependent on lower layers that are network and protocol specific. Indeed, the naming space (including addresses) differs from one network to another, hence addresses have no meaning outside of a unified infrastructure. Therefore packets in opportunistic networks have a collapsed architecture where all information whether concerning the application or networking operations is at the same level. With such a cross-layer design, packets can be slightly modified to fit any network they are forwarded through.

A concept that nicely fits with the underlying opportunistic networking model is offered by content-based communication ([CW03, CRW04]) whereby messages are forwarded from source to destinations based on their content rather than explicit addresses. In a content-based communication service, receivers declare their interests through receiver advertisements while senders simply publish messages without specifying a destination, and nodes in-between forward published content to interested receivers by matching the said content to the advertisements. Hence, content-based forwarding is a considerable enrichment over classical (address based) routing and is independent from the network and the associated low-level infrastructure, therefore content-based communication suits the opportunistic networking vision particularly well. Depending on the forwarding protocol, only part of the packet's content is used, and depending on which part is used, the forwarding protocol is sometimes referred to as context-based forwarding. In fact, the difference between content and context based forwarding is rather thin and there is no clear separation or commonly agreed-upon terminology in the literature. We further present our definition of context and content based communication in the section 2.3, and do not develop further at this point.

The implication of the intrinsic possibilities of such rich forwarding strategies, is that the targeted communication model in opportunistic networks is not conversational but rather disseminational by essence. This comes from the fact that most communication in opportunistic networks occurs over wireless links, and therefore packets are broadcasted by nature. Furthermore, the information used by rich forwarding protocols supports multiple destinations by default. Content-based communication, for instance, aims at disseminating a given content to all interested users without the need to define by advance each one of them. This does not mean that conversational communication is excluded of course, since a conversation is actually a dissemination involving only two parties: disseminational communication is in fact yet another enrichment offered by opportunistic communication over traditional communication paradigms.

To summarize, opportunistic communication is an exciting communication paradigm that can be seen as a generalization of MANETS and DTNs: in opportunistic networking, high mobility and delay-tolerance are essential characteristics that are envisioned as advantages from a capacity perspective rather than constraints. Furthermore, the collapsed architecture feature highlights another important and intrinsic characteristic of opportunistic networking, which is that communication is disseminational rather than conversational. An illustration of a typical opportunistic network is presented in figure 2.2: node mobil-

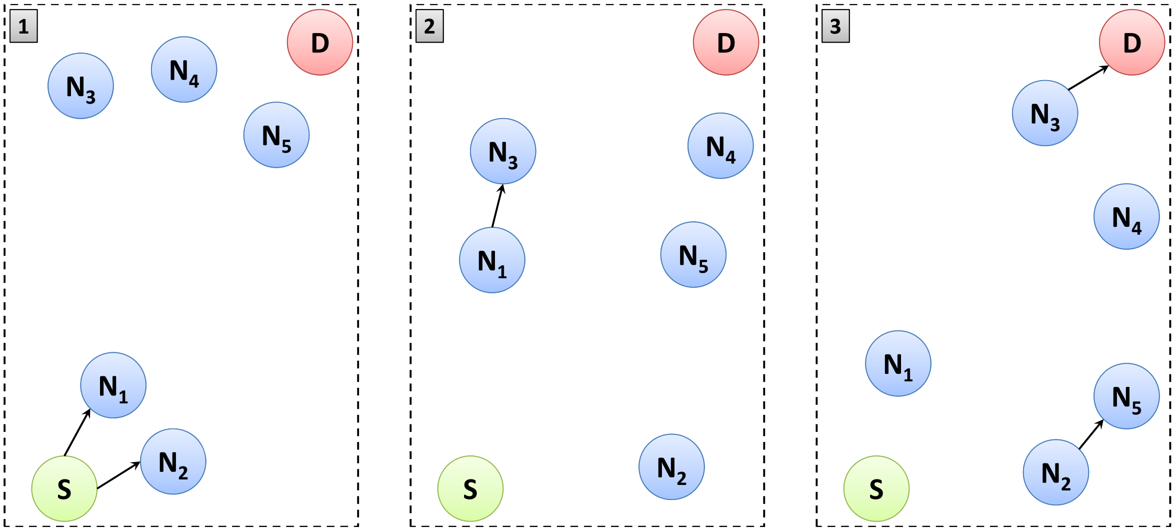


Figure 2.2: An illustration of time evolving networks and opportunistic forwarding. Time is running from left to right.

ity changes the topology of the network over time and allows communication between disconnected nodes. In the next section we present several application scenarios.

2.1.2 Application Scenarios

Many research projects tackle the delay-tolerant and opportunistic networking paradigm, and they differ by the application scenario and the underlying mobility model. From InterPlaNetary networks to experiences on colleges campuses, through emergency recovery from natural disasters or wars, vehicular networks or Ad-Hoc cities, there is a broad spectrum of realistic case studies. Following the classification of Pelusi et al. in [PPC06a] these applications mainly differ in the following aspects:

- user mobility pattern, as pedestrians, cars or satellites each have different speed and different constraints in the path they follow,

- communicating devices, which have different characteristics in terms of communication range, storage capacity, computation capability, and more generally resource constraints,

- connectivity pattern, which depends on the network density, transmission range of devices and the interference present in the environment.

We present three well-known examples amongst the most popular opportunistic applications.

2.1.2.1 Wildlife Monitoring

The first interesting application field for opportunistic networks is wildlife monitoring. Wildlife monitoring consists in attaching communicating devices on wild species in order to track them. Such tracking is indeed useful to study the behavior of these species and understand how they interact with each other or what is the influence of a disruptive phenomenon (like human activity or the introduction of non-native species) on the ecosystem. The information collected is also valuable to understand the mobility pattern of the tracked species and to monitor their reaction to weather changes for example.

The main challenge in wildlife monitoring is that the tracked species cover vast areas and it is therefore impractical to deploy a communication infrastructure that covers such a vast field of investigation. Opportunistic communication stands thus as a very attractive communication paradigm in this case.

There are two main projects aiming at wildlife species monitoring which use opportunistic communication. The first one called ZebraNet ([Zeb02]) is an inter-disciplinary effort with thrusts in both Biology and Computer System lead by university of Princeton. From a biological perspective the goal is to study the long-range migration, inter-species interactions, and nocturnal behavior of wildlife species. To this extent, researchers have deployed sensors (including a GPS) embedded in collars attached to zebras at the Sweetwaters Reserve managed by the Mpala Research Centre ([Mpa]) in the vast savanna area of central Kenya. From a communication system perspective the challenge consists in collecting positions and reporting interactions between zebras with extremely power-constrained devices and intermittent connectivity. The communication range of the collars are indeed relatively small (when compared with the covered area) and the base-station is attached to a mobile vehicle which cannot span the whole savanna in reasonable time. Therefore, collecting data with only collar-to-base station communication leads to poor results.

Juang et al. decided to overcome these limitations by allowing collar-to-collar communication in addition to collar-to-base station communication ([JOW$^+$02]), and they experimented two protocols: a flooding based protocol where collars exchange all their stored data upon meeting each other and a history based protocol where collars choose to transmit their data only to collars with the highest probability of encountering the base station. They showed through simulations that both approaches lead to significant improvement over the basic collar-to-base station communication. Furthermore, the history based approach performs better than the flooding-based approach when considering bandwidth, storage and energy constraints.

The second research project developed at Cornell University aims at monitoring whales in the ocean to investigate the constant decrease of the species. Similarly to ZebraNet, researchers equipped whales with specific tags to track them. The constraints of the project conducted by Cornell University are slightly different than in ZebraNet because the water medium further decreases the communication speed, but other than that problems are common: the ocean is too vast to have direct communication with whales at any time. Small and Haas therefore developed the Shared Wireless Infostation Model [SH03] that supports communication from whale-to-whale and from whale-to-base station to collect data as quickly as possible. We explain the principles of SWIM in section 2.2.1 along with other opportunistic forwarding approaches.

2.1.2.2 Opportunistic Networks for Developing Areas

Opportunistic networks are also an appealing paradigm to provide connectivity to rural and developing areas. Indeed some of those areas are considered not lucrative enough for business oriented deployment of a fixed telephony network, let alone high-speed Internet.

An example of opportunistic network for developing areas is the DakNet project [PFH04]. DakNet is an ad hoc network that uses wireless technology to provide asynchronous digital connectivity. The vision behind DakNet is that combining wireless and asynchronous services may be the start of a road to universal broadband connectivity. DakNet has been deployed in remote parts of both India and Cambodia at a cost two orders of magnitude less than that of traditional landline solutions. The DakNet architecture consists of kiosks deployed in villages and equipped with reasonable storage capacity and short range communication capability. Furthermore, mobile access points are deployed on vehicles (e.g. buses, motorcycles). These mobile access points carry data from rural kiosks to fixed access points in nearby towns, where broadband connectivity is available. DakNet deployment helps the development of local economy as it supports digital messaging (e.g. e-mail and e-commerce) but it also helps alleviating administrative burden as it also supports distribution of information (e.g. for e-governance, public health announcements, community bulletin boards but also news) and collection of information (e.g. voting, health records but also environmental sensor information).

The KioskNet project [Kio06] developed at university of Waterloo builds on the Daknet experience, with a focus on security and dependability. The basic architecture is therefore close to the DakNet deployment (with rural kiosks based on recycled computers, mobile access points and fixed access points) but both hardware and software architectures are improved. The hardware architecture makes the physical kiosks reasonably tamper-proof (to provide virus-free boot images and binaries) and resistant to pervasive dust and mechanical wear-and-tear. Furthermore, the local hard disk is not used to avoid hard disk failures and disk-resident viruses. Finally, Ur Rahman et al. investigate practical security solutions to provide dependability but also security (e.g. authentication) in [URHIK08].

Finally we mention the Saami Network Connectivity(SNC) project [DUP02] which aims at providing email, file transfer and cached web services to the Saami people. The Saami people are indeed nomadic reindeer herders living in lapland and following the migration of the reindeers.

At this point it is worth mentioning that all presented applications so far are only marginally opportunistic: opportunistic communication is used only as a way to reach an infrastructure (which can be considered a sink), but none of these applications supports opportunistic communication between nodes without relaying through the infrastructure.

2.1.2.3 Pocket-Switched Networks

The Haggle Project [Hag06] is a four-year project, started in January 2006 and funded by the European Commission in the sixth framework programme of the FET-SAC initiative. Haggle is a new autonomic networking architecture designed to enable communication in the presence of intermittent network connectivity, which exploits autonomic opportunistic communications (i.e. in the absence of end-to-end communication infrastructures).

In this framework, researchers are studying the properties of Pocket Switched Networks (PSNs), which are opportunistic networks composed of portable devices that users can carry in their pockets. Building on iMote experiments [HCS+05], the goal is to further characterize pair-wise contacts between devices based on contact durations and inter-contact times. The contact duration is the total time that a couple of mobile nodes are within communication range of each other and can therefore exchange data, while the inter-contact time is the time in between two contact opportunities between the same couple of devices. Both these parameters are important in opportunistic networks; contact duration affects the maximum amount of data that can be transfered during contact, while inter-contact time are useful to characterize the frequency of arising opportunities.

The analysis of these characteristics are based on real-life experiments on campuses or at conferences and it shows that both inter-contact times and contact durations are characterized by heavy-tailed distribution functions approximately following power laws. This has interesting implications on the delay that each packet is expected to experience throughout the network. In particular, Chaintreau et al. proved in [CHC+06] that "naive" forwarding protocols which follow statically computed rules that limit the number of replicas of each message and do not use any enriched information (e.g. previous contacts or the context) have an expected delay which is infinite under the heavy-tailed inter-contact times distribution found in the traces. This is a very important result, as it calls for more evolved forwarding paradigms exploiting knowledge about the users' behavior.

The Haggle project is not limited to characterizing opportunistic networks though, it also proposes a complete architecture for opportunistic communication that we present in Chapter 4, and it focuses on all aspects of opportunistic communication such as forwarding protocols, interaction with legacy network, human factors, resource management, and security, which is the focus of this thesis. Establishing trust and fairness among peer nodes, preserving user's privacy, and protecting against attacks aiming at tampering with data are indeed key factors that need to be addressed in order to make the opportunistic networking paradigm realistic. The required security mechanisms tightly depend on the forwarding strategy used during the communication. We therefore describe and classify the state-of-the-art opportunistic forwarding protocols in the next section and provide a detailed security analysis in chapter 3.

2.2 Opportunistic Forwarding Protocols

Routing is a compelling issue in opportunistic networks and it was the focus of many research efforts recently. Opportunistic forwarding is indeed radically different from traditional routing. For example, routing on the Internet implicitly assumes the following:

- Continuous and end-to-end connectivity,
- Reasonably low propagation delay,
- Very low packet loss rate.

MANETs also follow the same assumptions except the last one, but efficient opportunistic forwarding cannot rely on any of these assumptions. In the Internet, routing and forwarding refer to different operations: the routing operation consists in determining the best end-to-end path and implies methods for building routing tables at routers, while forwarding consists simply in transmitting a packet to a given next hop and can be considered as the last step of the routing process. In opportunistic networking there is not any more such a clear split between routing and forwarding since both routing and forwarding take place on a hop-by-hop basis. In the opportunistic networking literature, the terms routing and forwarding are thus used interchangeably. In this thesis, we choose to use the term forwarding for hop-by-hop approaches and we keep the term routing only to protocols taking decisions over several hops.

As shown in the previous section there are many application scenarios hence many different possible approaches for opportunistic forwarding. There are also many different ways of classifying these opportunistic protocols: in [PPC06b], Pelusi et al. classify them according to the existence or not of an infrastructure first with further branches in each case, while Zhang categorize them in deterministic or stochastic approaches in [Zha06]. We do not present deterministic approaches in this thesis because they are not suitable to opportunistic communication in the way we defined it. In [LZM$^+$09] Liu et al. sort opportunistic forwarding protocols out based mainly on the metric for prioritization used to select the relays, but their vision of opportunistic forwarding is focused on MAC layer protocols. As mentioned in section 2.1.1.3, packets in opportunistic networks are handeld as part of a collapsed architecture and therefore we do not consider MAC layer protocols independantly, we rather adopt a holistic approach toward opportunistic forwarding. In the following section we present some of the most significant examples of opportunistic or delay-tolerant forwarding protocols and then classify them according to our own criteria.

2.2.1 Epidemic Forwarding

The basic idea behind **epidemic forwarding (EF)** is inspired by epidemiology and consists in simulating the propagation of a disease: similarly to viruses or bacteria which are transmitted whenever two entities enter in contact, epidemic forwarding consists basically in forwarding a packet to all encountered nodes. This approach is therefore based on flooding in essence.

The scheme presented by Vahdat and Becker in [VB00] includes an additional mechanism to prevent transmitting messages to nodes that already received them: when two nodes are within communication range they exchange with one another message lists and then they exchange only messages that are not shared. Authors show that with the assumption of sufficiently large (or infinite) buffer this protocol manages to deliver nearly all messages in extremely mobile environment where ad hoc routing performs poorly.

The problem of the previous approach is obviously the overhead in terms of useless message exchanges, hence the assumption of infinite buffer size. At the other extreme, Grossglauser and Tse theoretically explored in [GT02] the following strategy: the source delivers the message to a random node which then stores and carries the message until it enters in direct contact with the destination at which point the message is delivered. This two hops approach has minimal overhead. This approach is efficient in very dynamic networks with the assumption that nodes follow random and independent paths, such that eventually a node meets all other nodes (including the destination). In such conditions the delivery rate is 100% but there is no guarantee (not even an upper bound) on the delivery time.

These two extreme approaches show the advantages and drawbacks of epidemic forwarding: the main advantage is that epidemic forwarding achieves almost always 100% delivery ratio with very few information required, while the disadvantage is either a huge bandwith waste or a potentially high delivery time. The main focus of research on epidemic forwarding therefore has been oriented towards methods to limit the flooding in epidemic forwarding. A notable proposal in this area is **Self Limiting Epidemic Forwarding (SLEF)** [EFLBS06a, EFLBS06b] by El Fawal et al. which consists in limiting the spread factor based on the Time To Live (TTL) feature. Authors of this proposal explore several variations on how to decrement the TTL: either at each hop, or at each transmission or with an aging mechanism (meaning that the TTL is decremented even if the message is not forwarded). El Fawal et al. show that with the appropriate factors it is possible to keep a relatively low delivery time with limited overhead in terms of bandwith waste.

Spyropoulos et al. presented another version of epidemic forwarding called **Spray and Wait** [SPR05]. Spray and Wait focuses on limiting network congestion by keeping control on the number L of copies of messages in the network. There are two versions of Spray and Wait: in the Vanilla version the source starts with a Spray phase in which it distributes copies of a message to the first L nodes that it encounters. Then the source enters the Wait phase and intermediate nodes do not spread the message again unless they encounter the destination. This approach is therefore very close to [GT02] with the exception that L relays are chosen instead of one. In the Binary version of Spray and Wait the source starts, as before, with L copies. The source then forwards $\lfloor \frac{L}{2} \rfloor$ of its copies to the first node it encounters and keeps the rest. Each of the nodes then transfers half of the total number of copies they have to other nodes that they meet and that do not already have copies of the message. When a node eventually gives away all of its copies, except for one, it switches into the wait phase where it waits for a direct transmission opportunity with the destination. This second version is more efficient because it disseminates messages faster than the basic version.

An even more sophisticated approach was proposed by Balasubramanian et al. in [BLV07], where they consider DTN forwarding mainly as a resource allocation problem. Therefore they propose a protocol called **Resource Allocation Protocol for Inten-**

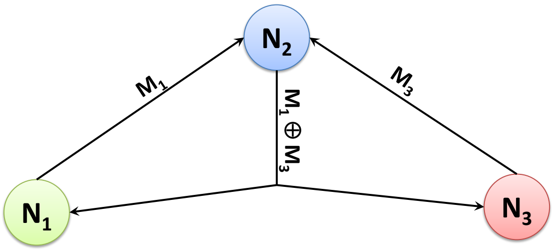


Figure 2.3: Network Coding scenario: N_1 and N_3 send M_1 and M_3 respectively to N_2, then N_2 broadcasts $M_1 \oplus M_3$ to N_1 and N_3.

tional DTN routing (RAPID) that intentionally minimizes one of three metrics: average delay, missed deadlines, and maximum delay. To this extent they define a utility function that assigns a utility value U_i to every packet i, which is based on the metric being optimized. RAPID then replicates packets that locally result in the highest increase in utility first. The overall protocol first exchanges metadata to gather information and estimate packet utilities and then replicates packets based on marginal utility. In essence this is an epidemic forwarding protocol with a utility function that offers a general framework to limit the flooding.

An orthogonal approach proposed by Widmer and Le Boudec in [WLB05] consists in applying network coding principles to increase the efficiency of epidemic forwarding and to take advantage of the broadcast nature of the wireless medium (we denote it by **NCOR**). The idea is to send linear combinations of several packets instead of sending each packet individually. For example if we consider a simple scenario with three nodes N_1,N_2, N_3 where N_1 and N_3 want to exchange information M_1 and M_3 respectively through the middle node N_2. Using classical epidemic forwarding this scenario requires four message exchanges while it is possible to achieve the same result in three message exchanges by using network coding as illustrated in Figure 2.3: N_1 sends M_1 to N_2 and N_3 transmits M_3 to N_2, then N_2 broadcasts $M_1 \oplus M_3$ to N_1 and N_3. Since N_1 has packet M_1 and N_3 has packet M_3 they can use the message broadcasted by N_2 to retrieve the missing packet through decoding (M_3 for N_1 and M_1 for N_3).

Finally epidemic forwarding was also proposed in conjunction with the existence of an infrastructure. In these approaches infrastructure is composed of base stations that represent gateways toward traditional networks, e.g. Internet or Local Area Networks (LANs). In the **Infostation Model (IM)** proposal [IR02], mobile nodes forward messages only to the base stations and the base station then delivers the message to the destinations. This approach is very similar to [GT02], with the difference that the relay nodes are part of the infrastructure instead of being randomly selected terminals. This approach therefore has the same drawback as [GT02], namely the fairly high delivery time. An enhanced version of this protocol called **Shared Wireless Infostation model (SWIM)** [SH03] allows node-to-node communication in addition to node-to-base station communication. In this approach nodes forward the messages to the base station if they are in communication range with a base station, otherwise they opportunistically forward the message through any other node that will eventually forward it to the base station. SWIM adds an opportunistic forwarding mechanism in the node-to-base station communication by enabling node-to-node forwarding, but base station-to-node communication remains limited to one hop (direct forwarding). The infrastructure is not necessarily a fixed one, and several protocols have been proposed with mobile access points which are referred to as mules, ferries or carriers. In the **data-MULE** system [JSB^{+}06], MULEs are mobile agents that move in an area covered by sensors (which are either fixed or with limited mobility) and gather the sensed data to transfer it to the sinks through wired access points. Another proposal called **Message Ferrying (MF)** [ZAZ04] features mobile ferries that offer a message relaying service. The originality of this solution lies in the fact that two different modes are proposed. In the first one, called Node-Initiated Message Ferrying the ferries move around a predefined and known path and nodes need to move to meet the ferries and transmit the message, while in the Ferry-Initiated Message Ferrying source nodes are fixed and send a ServiceRequest to ferries (like calling a taxi) and the ferry changes its trajectory to meet up with the source node.

2.2.2 Protocols Based on a Simple Estimation of the Forwarding Likelihood

The previously presented protocols aim at replicating packets such that they eventually reach the destination with a focus on limiting the flooding. However, these protocols do not feature a mechanism to actually route the packet in the sense of bringing it closer to the destination. We now present protocols that estimate likelihood of delivery and then forward the packet following increasing likelihood of delivery.

A first approach was presented by Davids et al. in [DFL01]. In this approach each packet is associated with a likelihood of delivery. When two nodes meet, they exchange packets with the highest likelihood of delivery and discard packets to maintain a constant buffer size. Packets are discarded according to four possible strategies (Drop-Random, Drop-Least-Recently-Received, Drop-Oldest and Drop-Least-Encountered). According to simulation results, Drop-Oldest and Drop-Least-Encountered yield the best performance. The likelihood of delivery denotes the probability for a node N_i of encountering a node N_j. In the beginning the likelihood is set to 0 for all nodes, then when node N_i meets node N_j the likelihood is set to 1. The likelihood then automatically decreases with time until N_i meets N_j again (at which point it is set to 1) or until it eventually reaches 0. This basic approach was then extended by Burns et al. in [BBL05] yielding the **Meetings and Visits (MV)** protocol, where the only difference is the likelihood estimation method. The method used in MV is more complex and fine-grained since it determines the probability that a node N_i can successfully transfer a packet to a region r (instead of a precise node) in n transfers (n hops).

One of the most popular approaches based on estimation of delivery probability is the **Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET)** developed by Lindgren et al. and which was first described in [LDS03]. In

PROPHET every node N_i estimates a delivery predictability $P(N_i, N_j)$ to node N_j: this probability represents how likely it is that N_i will be able to deliver the message to N_j. N_i does not need to meet N_j directly: N_i could be a good forwarder to N_j by transitivity (if N_i meets often N_k which in turn often meets N_j). $P(N_i, N_j)$ is initialized at P_{init} which is a constant in $[0, 1]$. Then when nodes N_i and N_j meet (we describe what happens from N_i's side only but the same happens at N_j):

- The predictability for N_j is increased:

$$P(N_i, N_j)_{new} = P(N_i, N_j)_{old} + (1 - P(N_i, N_j)_{old})P_{init},$$

- The predictability for all other nodes N_k is decreased :

$$P(N_i, N_k)_{new} = \gamma^u P(N_i, N_k)_{old},$$

  where γ is an aging constant in $[0, 1]$ and u is the number of time units that has elapsed since the last aging,

- Predictabilities are exchanged between N_i and N_j and the predictability of destinations N_k for which N_j has a probability $P(N_j, N_k)$ is updated to reflect the transitivity of predictability:

$$P(N_i, N_k)_{new} = P(N_i, N_k)_{old} + (1 - P(N_i, N_k)_{old})P(N_i, N_j)P(N_j, N_k)\beta,$$

  where β is a scaling constant.

PROPHET was successful in the research community because it is easy to implement and simulations showed that in certain conditions, PROPHET would improve the delivery ratio by up to 40% over epidemic forwarding.

A somehow similar protocol called **MaxProp** was presented by Burgess et al. in [BGJL06]. MaxProp prioritizes messages that should be transmitted or dropped based on an estimated likelihood of a future transitive path to the destination. It is assumed that the total number n of nodes in the network is fixed and known in advance. The likelihood for N_i to meet any node N_j at the beginning is set to $P(N_i, N_j) = \frac{1}{n-1}$. These predictability values are stored in a node-meeting likelihood vector of size $n - 1$. When N_i meets a node N_j it increments $P(N_i, N_j)$ by 1 and then it normalizes the entire vector such that the sum of all probabilities remains equal to 1, which means that each updated entry in the vector is aged by a factor of $\frac{1}{2}$. N_i and N_j also exchange their estimated node-meeting likelihood vectors. Eventually, every node will have a vector from every other node. With these n vectors at hand, N_i can compute the weight of a path to the destination as the sum of the weights of its one hop components. The weight of a direct transmission between N_i and N_j is the complementary of the probability of N_i encountering N_j $(1 - P(N_i, N_j))$. The path with the least total weight is chosen as the cost for that particular destination. The messages are then ordered by destination costs, and transmitted and dropped in that order.

Another protocol which uses history of encounters is **FRESH** proposed by Dubois-Ferriere et al. in [DFGV03]. In FRESH, each node stores the time of last encounter of other nodes. When node N_1 meets node N_2, they exchange their history of meetings, and they respectively forward to each other the messages destined to nodes which have been seen more recently. FRESH does not use transitivity as opposed to PROPHET. Generalizing FRESH, Erramilli et al. propose in [ECCD08] a more efficient protocol called **Delegation Forwarding (DF)** that supports any quality metric (the metric of FRESH being one particular example used for performance evaluation). The idea is to forward messages only to nodes who have encountered the destination the most recently based on previous observations. More generally, messages are transmitted to the node with the maximum quality observed so far, but no a priori global knowledge is required. DF therefore decreases the number of replicas of each message in the network by more than half compared with classical strategies like FRESH, while delivery time are very close.

Finally, the **Bubble Rap** protocol presented by Hui et al. in [HCY08] is a social based forwarding protocol and therefore differs from the previous protocols by the metric which is used. In Bubble Rap it is assumed that nodes are clustered in cliques or social groups and are highly non-homogeneous: the number of social links (also called social connectivity degree) each node has towards others is highly variable and is distributed according to power laws. The social structure of a clique is known by members of this clique and therefore forwarding a message inside a clique is straightforward. According to its social connectivity degree a node has a global ranking (among all other nodes) and a local ranking within its clique. For messages where the source and the destination do not belong to the same clique, the idea is that messages are pushed up (bubbled) from the source towards more sociable users using the global ranking until a contact with the destination's clique is found. Then inside the clique, the local ranking is used to bubble the message again towards locally more sociable users until it eventually reaches the destination. Simulations based on the Reality dataset [ESP06] show that most of the time Bubble Rap achieves a similar delivery ratio to Prophet while creating in average half of the copies of each message.

2.2.3 Enriched Forwarding Proposals

All the previously described protocols exploit a particular piece of information to optimize the forwarding task: PROPHET, MaxProp and MV use the frequency of meetings between nodes and optionally the frequency of visits to specific regions, while Bubble Rap uses the social cliques and social connectivity degree. Yet, other approaches considering a collapsed architecture offer enriched forwarding strategies which are more suitable for opportunistic networking.

The **Context Aware Routing (CAR)** protocol proposed by Musolesi et al. in [MHM05] assumes a typical delay-tolerant network with several MANET regions which are intermittently connected. Inside each MANET region a proactive routing protocol is used (e.g. DSDV [PB94]). To reach nodes outside the cloud, the source looks for the node in its region with the highest probability of delivering the message successfully to the destination: this node stores the message and waits for an opportunity to reach the destination's region. The evaluation of delivery probability relies on a multi-attribute utility-based framework that uses a time series analysis technique called Kalman Filter. This framework accommodates different types of context information, e.g. the residual battery life, the rate of connectivity change and the probability of meeting between nodes are used in [MHM05]. Yet this context information is only available inside a given region, therefore nodes which have never been in the same region cannot exploit each other's context information. The simulations performed by Musolesi et al. show that the delivery ratio of CAR is better than EF when the buffer size is small, but the opposite holds when the buffer capacity is large.

Next, there are two other context-based forwarding approaches that are more general in the sense that they do not assume the existence of MANET regions, are able to exploit context information for all nodes (even those that have never been within the same region or in contact), and take into account some social information in the context.

The first approach called **Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (Propicman)** was presented by Nguyen et al. in [NGP07]. In Propicman, every node has a node profile which describes the context of the node, in the form of couples evidences/values. The set of evidences is fixed and is the same for all nodes but it can be adjusted to encompass any context information. The set of values describes the node context that is different for each node. When a source node N_S wants to send a message M to a destination N_D, N_S needs to know at least some part of the profile of N_D: this information is appended as header of the message $H(M)$ and the routing is based on $H(M)$ only. To select the best forwarder, N_S sends the header $H(M)$ to its neighbors that then can compute the matching ratio between the destination profile and their own profile. The idea is to forward the message to nodes with increasing matching ratio until the message reaches the destination eventually. As a refinement, Propicman does not base its forwarding decision on the next hop only but on the next two hops as well: each nodes needs to collect the matching ratio between the header of the message and its neighbors and the neighbors of the neighbors in order to increase the performance of the protocol. Propicman is therefore a hop-by-hop routing protocol that uses two hop context information. Simulation results show that in a sufficiently dense network, Propicman achieves faster delivery than PROPHET, but EF is the fastest. Nevertheless EF performs poorly from a bandwidth consumption perspective, therefore if buffer size was taken into account EF would perform worse.

History Based Routing Protocol for Opportunistic Networks (HiBOp) proposed by Boldrini et al. in [BCJP07] is the second fully context aware forwarding protocol in opportunistic networks. The HiBOp approach is partly similar to Propicman and the main difference is the way the context information is managed. HiBOp distinguishes between three different types of contexts:

- context of the node: each node locally stores an Identity Table (IT) which contains personal information on the user structured as a set of attributes,

- context of the neighbors: nodes exchange their ITs when getting in touch. The set of current neighbors' ITs and the node's own IT represent the current context.

- historical context: each attribute seen in the current context is recorded in a History Table (HT) together with a Continuity Probability index that represents the probability of encountering that attribute in the future.

As in Propicman, messages are forwarded to nodes with increasing match with the context attributes of the destination. However, as opposed to Propicman, in HiBOp the matches are evaluated as delivery probabilities, and distinct probabilities are evaluated based on the Current Context P_{CC} only and on the History P_H only. The final probability is computed as a weighted average: $P = \alpha P_H + (1 - \alpha) P_{CC}$, where $0 \leq \alpha \leq 1$ is a parameter that sets the relative importance of the Current Context and the History. Finally it is worth noting that in HiBOp only the source node is allowed to replicate the message, in order to control the trade-off between reliability and message spread. Boldrini et al. study the impact of many parameters on the efficiency of their scheme and, in [BCP08], they show in particular that information on the users' social behavior can be exploited to highly increase the efficiency of the forwarding scheme.

Another sophisticated routing concept is content based routing where messages are routed depending on their content and the interests of users instead of specifying a destination. Content based routing schemes were first proposed for fixed networks and then MANETs but few adaptations to opportunistic networks have recently been proposed. The first one, that we denote **CBRHDM** (for Content-Based Routing in Highly Dynamic MANETs), was proposed by Baldoni et al. in [BBQ$^+$05]. CBRHDM broadcasts messages to efficiently disseminate them in the network from neighbor to neighbor. Neighbors then decide whether to forward the message or not based on an estimation of their distance from a potential subscriber of the message. Therefore, CBRHDM does not require any network-wide structure to support forwarding decisions and is very resilient to topological changes that occur in highly mobile networks. The protocol does not support asynchronous communication though and therefore communication between disconnected regions is not possible. A similar idea, that we call **CBCDM** (for Content-Based Communication in Disconnected MANETs), was proposed by Haillot and Guidec in [HG08] but their solution features two layers: one abstraction layer that supports opportunistic content-based communication at one hop distance, and an underlying layer that supports multi-hop communication in connected regions. The first layer enables nodes to advertise in broadcast mode their interests and a directory of the content that they offer such that interested nodes can request in unicast specific documents. The requested documents are then broadcasted and nodes receiving a document opportunistically (through the broadcast communication without having requested it) keep the document and add it to their document base only if they are interested in it.

The two previously described protocols achieve content-based forwarding on a hop-by-hop basis but they do not allow the construction of a content-based routing structure. Costa et al. propose in [CMMP06] an interesting hybrid approach that we call **ACBRDTM** (for Adaptive Content-Based Routing for Delay-Tolerant MANETS): they propose to locally build a content-based routing structure in regions with continuous connectivity. They also complement the subscription information necessary to content-based routing with information about the changes in the context observed by nearby nodes. This information is used to estimate which nodes are potentially good message carriers towards disconnected regions. The evolution of context information is predicted based on previous observations using Kalman filter forecasting techniques as in CAR.

2.3 Classification of Opportunistic Forwarding Strategies

The protocols that we presented in the previous section generally take forwarding decisions on a hop-by-hop basis (with small variants like in the case of Propicman). This hop-by-hop communication is combined with a *"store, carry and forward"* paradigm. This paradigm encompasses the forwarding philosophy best suited to opportunistic networks and the operations that are required:

- Store: nodes need to store received packet in their cache, even if they are not the final destination of the packet: collaboration among nodes is mandatory and nodes should not act selfishly by keeping only packets destined to them. Since storage capacity is a limited resource, a cache management mechanism is required, with a prioritization metric to decide which packets should be kept and which ones can be dropped.
- Carry: this is the core of opportunistic networking. In order to overcome the limitations of the infrastructure or the restricted communication capabilities of devices, opportunistic networks rely on node mobility to carry messages and enable communication between entities that are disconnected otherwise.
- Forward: a node N_1 has to decide which packet to forward to a newly encountered node N_2. While it is obvious that N_1 should forward all packets destined to N_2 first, it is more difficult to decide which other packets should be forwarded to N_2 because N_2 would be a better carrier of the message than N_1. This implies a metric to measure the quality of a node with respect to a given message, as performed by a number of strategies suggested in the literature.

Opportunistic forwarding mechanisms are therefore characterized first by the prioritization metric used, whether at the cache management level or at the forwarding step. There are numerous possible metrics and we presented many examples in the previous section. It is yet hard to compare them in terms of efficiency because they are based on very different measurements ranging from time to live (TTL) to content through contact history, social rank or location, and each metric usually is designed for a specific scenario where it works bests. We propose therefore a first classification based not on the efficiency of the metric but on an evaluation of its **complexity**: complexity to gather and manage the information and then complexity to process it. For example TTL as used in SLEF is straightforward to gather and it is easy to process while history of encounters and transitivity used by PROPHET requires more efforts to manage, and fully context based approaches require even more information and processing resources.

A second aspect for the classification is whether or not the protocol creates replicas of messages. In [BLV07], Balasubramanian et al. classify many forwarding protocols in this way and they call protocols that never replicate a message "forwarding-based", while protocols that do replicate messages are considered "replication-based". This taxonomy is interesting but it is too manichean: many "forwarding-based" (or single copy) protocols make few copies of messages in practice to improve reliability, and this classification does not differentiate "replication-based" (or multiple copies) protocols that tightly control the number of replicas (e.g. Spray and wait) from protocols that do not (e.g. EF). We therefore propose a more flexible scale of the protocol **cost** in terms of bandwith requirement, which is based on a rough estimate of the number of message replicas.

Based on these two characteristics we classify the protocols presented in the previous section in Figure 2.4. The colors correspond to the section in which the protocols were described: red and bold font for epidemic forwarding approaches (section 2.2.1), blue and regular font for protocols with simple estimation of the forwarding likelihood (section 2.2.2), and green and italic font for enriched forwarding proposals (section 2.2.3). The scale on the axis is intentionally omitted since we only focus on the relative positions of protocols. This graph shows for example that EF is clearly the most costly protocol but also the simplest. On the contrary, PROPICMAN and ACBRDTM are the most efficient ones because they really attempt to forward a single copy of each message but they are amongst the most complex solutions. In general, the less costly and the less complex the protocol, the better, but we need to keep in mind that there are many other parameters that are not taken into consideration here. For example SWIM and MV are much more bandwidth efficient than EF while having more or less similar complexity, but EF offers much better delivery delay. Also EF works in any network setting, while MV and SWIM require a network infrastructure. Therefore the point is that this graph is not sufficient to compare protocols, a good comparison would be protocol to protocol in a precise scenario, but the graph provides useful information on the relative cost and complexity of the various protocols presented.

The figure also shows an interesting link between protocol complexity and mode of communication. We distinguish between two main communication modes as follows:

- **Conversational communication**, which implies a communication between two well defined nodes. This is the classical communication paradigm that is used in telephony and legacy internet routing.

- **Disseminational communication** is a multi-party communication mode which aims at routing a message from a source to a set of interested destinations. In the classical layered architecture, disseminational communication is represented by multicast at the network layer and by publish-subscribe or other messaging applications at the application layer.

In wired networks, conversational communication is the natural communication paradigm, but disseminational communication should be the default mode in wireless networks because of the broadcast nature of the medium. It is interesting to note that, although

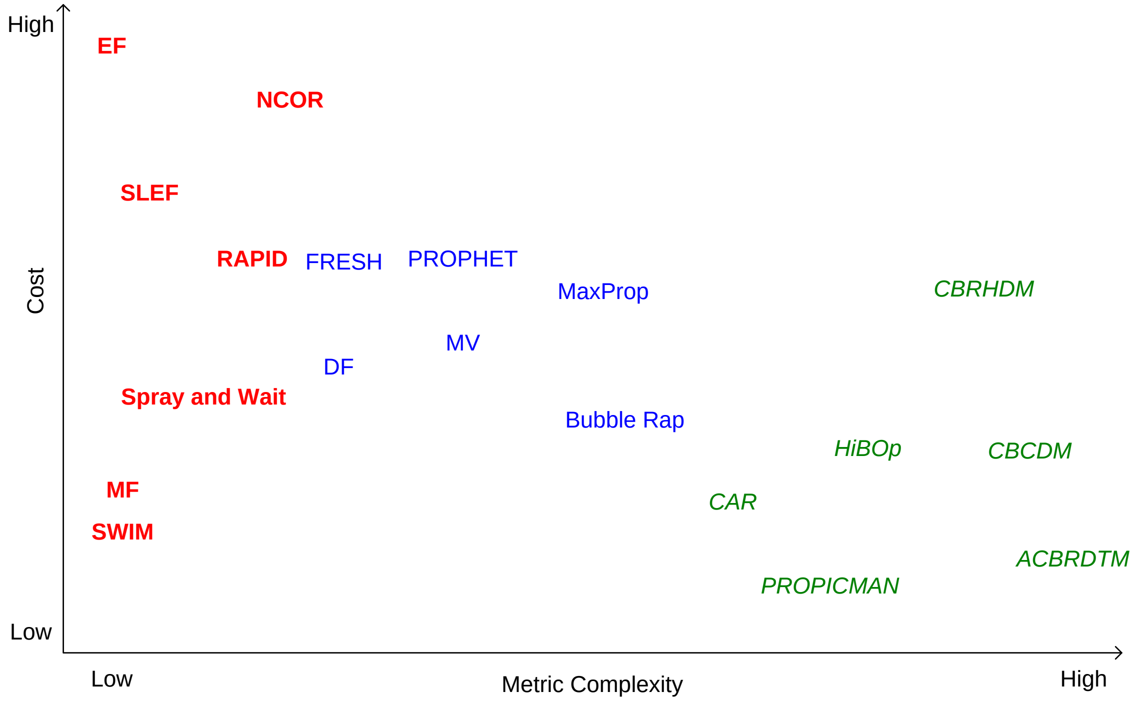


Figure 2.4: Cost/Metric Complexity comparison of opportunistic networks.

opportunistic networks are mainly wireless networks and most proposed protocols rely solely on broadcast, support for disseminational communication is the exception rather than the rule. The reason is that disseminational communication requires radically different architecture from the classical source-destination model, but such a structure is inherently offered by collapsed architectures. The collapsed architecture allows indeed to define destinations not only explicitly by a precise identifier (like an IP address) but also implicitly as being any node satisfying a set of conditions. At this step, we define four categories of protocols depending on the way the destination is defined.

- **Content-based Communication** (sometimes referred to as content-centric communication) is a paradigm where messages do not include a destination and rather only include descriptors that describe the content of the message. Nodes receive messages based on their interests: these interests can be advertised in a content-based routing structure or simply be used on the fly for nodes to decide whether to keep a message or not. Content-based communication is disseminational in essence.

- **Context-based Communication** (sometimes referred to as fully context-based or context aware communication) is a paradigm where the destination is implicitly defined through its context. The context of a message consists mainly of the profile of the destination and is independent of the content of the message. The profile of a node consists of information related to the physical node and includes personal information about the user (e.g. name, email address), its location (e.g. addresses), its activities (e.g. work, hobbies), social relations (e.g. friends) and so on. In context-based communication, messages are forwarded to nodes with increasing matching context until they eventually reach the implicit destination which matches all the context. Depending on the number of information included in the context of the message and its precision, context-based communication can accommodate both disseminational and conversational communication.

- **Partially Context-aware Communication** follows the same general principles except that context is reduced to few attributes (e.g history of encounters) that are used solely for forwarding purposes, while the destination is explicitly defined by the source. These protocols thus fall under the conversational model.
- **Oblivious communication** or to be more precise content and context oblivious communication is the classical source-destination paradigm and where routing is performed independently of context or content but only based on the destination of the message and possibly on other control information of the packet (e.g. time to live).

It is interesting to observe that the complexity is tightly linked with the category of communication: content-based communication (ACBRDTM, CBCDM and CBRHDM) is the most complex mechanism followed by context-based communication (PROPICMAN, HiBOp, CAR), partially context-aware communication (Bubble Rap, MaxProp, MV, PROPHET, FRESH, DF) and finally the most simple metric corresponds to oblivious protocols (NCOR, RAPID, Spray and Wait, SLEF, MF, SWIM, EF). This result is not really surprising but it confirms that the increment in complexity is justified as it comes with conceptually different communication paradigms.

Another way to present this evolution in concepts is illustrated in Figure 2.5. This graph represents the amount of information in the packet that is used for forwarding as a function of the degree of inaccuracy of knowledge about the destination.

Oblivious protocols lie in the lower left corner, because they offer the greatest accuracy on the destination of the message (the destination is defined with a unique node identifier or with a network address) while they scarcely rely on the information included in the packet (mainly the destination) to perform the forwarding operation. Above them lie partially context-aware forwarding protocols that feature the same destination accuracy but rely more on the information included in the packet (few information context in addition to the destination). Then, there are fully context-based protocols which use even more information than partially context-aware protocols, and where the destination is not precisely defined by an identifier but is implicitly defined as being a node that verifies a set of context information. Finally content-based forwarding protocols are located in the upper right corner as they do not specify any information about the destination and make use of the content of the packet itself to perform forwarding operations.

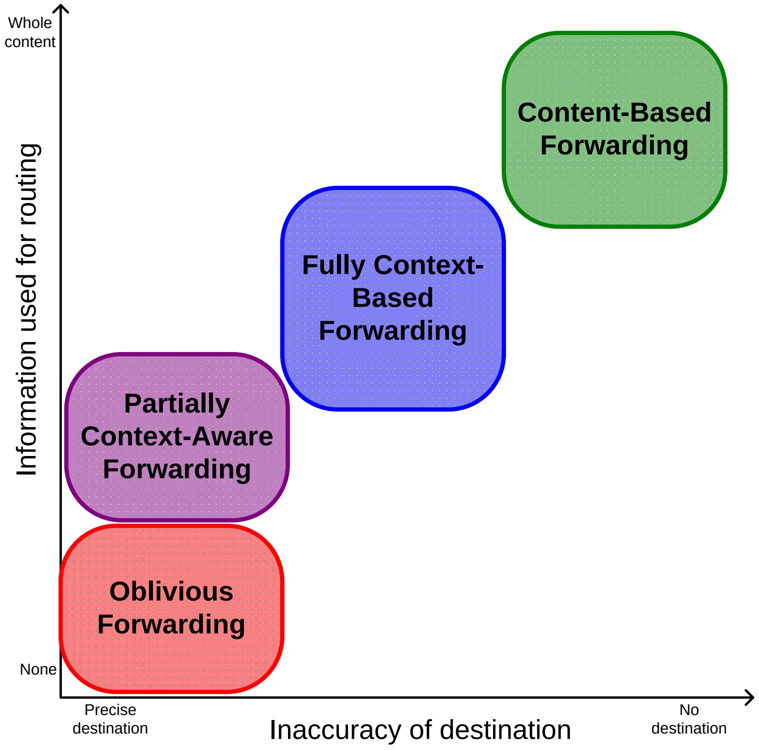

Whole content
Information used for routing
None
Precise destination
Inaccuracy of destination
No destination
Content-Based Forwarding
Fully Context-Based Forwarding
Partially Context-Aware Forwarding
Oblivious Forwarding

Figure 2.5: Communication categories in a message forwarding information/destination's inaccuracy graph.

Again one should be careful about how to interpret this classification. At a first glance, one would think that the more accurate the information on the destination is, the more efficient the forwarding protocol would be. This intuition is true indeed in the Internet or any network with a well-organized infrastructure but it does not hold when it comes to opportunistic networks. In opportunistic networks, forwarding is performed by peer intermediate nodes (as opposed to routers), and for these intermediate nodes the unique identifier of the destination has no meaning (unless they already encountered it). On the contrary context or content information can be evaluated locally as these are more expressive and refer to data concerning the intermediate node itself or its neighbors. Thus, context or content information are more meaningful and thus more useful than identifiers like addresses to take relevant opportunistic forwarding decisions.

Finally, it is interesting to think about the meaning of the remaining blank regions in Figure 2.5. The lower right corner would represent protocols that do not specify the destination or specify it indirectly and with no information from the message used for forwarding. This means that forwarding is performed blindly and that only the destination knows that it is actually the destination. Hence, this situation is even worse than epidemic forwarding. The upper left corner would correspond to protocols where the destination is precisely defined and where the content of the packet itself is used to perform forwarding. Protocols in this region would therefore benefit from two different ways or performing forwarding and so could be used as generic forwarding algorithms: the algorithm would use oblivious forwarding in the presence of an infrastructure and content-based or context-based forwarding in opportunistic networks. The main advantage in this scenario would be adaptability rather than efficiency. Another interesting scenario would be to complement both information for infrastructured networks: use the oblivious information to perform forwarding, and use the content or context to enhance Quality of Service, load balancing or maybe even to filter unwanted traffic. We believe that this idea can be pursued further as an interesting research direction for the next generation Internet.

2.4 Summary

In summary, opportunistic networking is a new communication paradigm that enables data delivery in challenged conditions. These conditions imply strong assumptions such as high mobility, lack of end-to-end connectivity, and limited infrastructure and resources. Consequently, new forwarding mechanisms need to be devised to cope with these assumptions: routing in MANETs already innovated with respect to traditional routing on the Internet, and opportunistic forwarding needs to go even beyond, with flexible, highly dynamic, and local forwarding strategies. In the past few years, researchers proposed many different solutions to forwarding in opportunistic networks, and we classified these solutions in four conceptual categories:

- **Oblivious forwarding protocols**, which precisely define the destination of messages and mainly adopt an epidemic forwarding strategy to reach the destination with a focus on heuristics to limit the flooding.

- **Partially context-based forwarding protocols**, which precisely define the destination of messages but take forwarding decisions based on one piece of context information (such as history of encounters or social relationships).

- **Fully context-based forwarding protocols**, where the destination is implicitly defined through its profile. Forwarding decisions take into account the whole context information and messages are forwarded to nodes with increasing matching context until they reach the destination.

- **Content-based forwarding protocols**, which do not define a destination at all: content is forwarded from publishers to receivers based on nodes' interests.

The main interest of this classification is to stress on the conceptual evolution of forwarding approaches: from traditional forwarding based on addresses that we refer to as oblivious forwarding, to context-based forwarding that offers a flexible forwarding paradigm through implicit definition of the destination and finally to content-based forwarding where no destination is defined and which is particularly suited for data dissemination. These radically different forwarding paradigms raise distinct concerns and have specific requirements, not only for forwarding aspects, but also from data management point of view and from a security perspective. Security is indeed a primary concern in networking and opportunistic networks require to revisit security traditional solutions to adapt them to the specific characteristics of opportunistic communication (e.g. delay tolerance, lack of infrastructure). On top of that the aforementioned new forwarding paradigms raise challenging and original security concerns especially from a privacy perspective. These considerations are the focus of this thesis and we therefore detail in the next chapter the security issues pertaining to opportunistic networking in general, and to context and content based forwarding principles in particular.

Chapter 3

Security Issues in Opportunistic Networks

Enforcing security services is important both for the dependability and the user's acceptability of opportunistic networks. As part of the broad field of pervasive computing, opportunistic networks raise important problems as stated by Thibodeau in [Thi02]:

"Pervasive computing has pervasive problems, not the least of which are interoperability, security and privacy."

Many security aspects have been studied in traditional communication infrastructures and numerous solutions have been proposed, but, as presented in [Shi10], radically new solutions are required to fit the specific needs and constraints of opportunistic networks. Indeed, nodes' high mobility implies that security solutions should be dynamic and local. The ad-hoc nature of opportunistic networks also calls for self-organized security solutions. Furthermore, delay tolerance, which is one of the main characteristics of opportunistic networks, has a strong impact from a security perspective as it amounts to the infeasibility for a node to contact at any time a centralized distant security server or the destination, hence interactive protocols are infeasible in opportunistic networks. Moreover, disseminational communication requires enriched forwarding strategies that use information provided by the application to take forwarding decisions: the collapsed architecture raises completely new security issues, as it implies that security solutions should take into consideration the requirements of application and networking at the same time, and cannot secure information at each layer separately.

Ideally, security solutions should perform as autonomous and self-organizing services with no security infrastructure at all. However, even though online security infrastructures are not feasible in opportunistic networks, the following intermediate paradigms can offer a good compromise:

- **Offline security infrastructure** refers to the existence of a security infrastructure that can be contacted prior to communication establishment or possibly to provide delay-tolerant security services, but that cannot be contacted during a communication.

- **Optimistic security service** is a service that does not require a security infrastructure during regular network operation with honest users ; the security infrastructure is contacted only in case of node misbehavior to resolve conflicts between honest and malicious nodes.

We now present the security challenges in opportunistic networks and describe requirements and directions for adapted solutions. We start by the general issues of cooperation enforcement and trust establishment. Then, we focus on requirements of secure communications, such as authenticity, integrity, confidentiality, and privacy, and finally the underlying component of all cryptographic applications, namely, key management.

3.1 Cooperation Enforcement

Collaboration amongst nodes is essential in all Peer-to-Peer (P2P) networks for the benefits of all users. Saroiu et al. showed thus in [SGG03] that a quarter of all Gnutella users do not share any file and only take advantage of the service without returning the favor. These selfish nodes, sometimes called as free-riders [AH00], decrease the performance of the network and it is therefore important to enforce cooperation amongst nodes. Mannak et al. for example showed in their study [MdRK04] that half of the users would share more if they had a materialistic incentive to do so, and several works proposed such incentives (e.g. Golle et al. [GLBML01] proposed several micro-payment mechanisms to foster file sharing in centralized P2P networks).

MANETS and opportunistic networks are specific cases of P2P networks and therefore collaboration is an important issue, even more vital than in classical P2P systems. The lack of infrastructure in these networks and in particular the lack of designated routers means that all nodes are expected to take part in the forwarding process in order to increase the communication opportunities and the throughput along. The cooperation between nodes is therefore vital for communication independantly of which forwarding strategy is adopted, and nodes' selfishness, which is increased for small devices because of the scarcity of resources, should be restrained. The solutions proposed in classical P2P networks are however not adapted because they rely on an online centralized infrastructure.

There are two approaches towards enforcing cooperation: either providing incentives for cooperation or punishing selfish nodes. The latter is not really practical in opportunistic networks because some nodes might not want to interact with other nodes at certain times for a valid reason (e.g. low remaining resources). Incentives are not necessarily materialistic though, and we distinguish between two classes:

- **reputation mechanisms** whereby nodes agree to cooperate with each other based on their past behavior in the network, which is expressed by their reputation.

- **rewarding mechanisms** whereby in return for each contribution, collaborating nodes receive a certain amount of reward that they further can use for their own benefit.

Examples of reputation mechanism include CONFIDANT [BLB02a, BLB02b] and CORE [MM02]. In CONFIDANT, reputation is used to evaluate routing and forwarding behaviour according to the network protocol. Nodes monitor their neighbours and change the reputation accordingly. If they have a reason to believe that a node misbehaves, they can take action in terms of their own routing and forwarding and they can decide to inform other nodes by sending an ALARM message. When a node receives such an ALARM either directly or by promiscuously listening to the network, it evaluates how trustworthy the ALARM is based on the source of the ALARM and the accumulated ALARM messages about the node in question. It can then decide whether to take action against the misbehaved node in the form of excluding routes containing the misbehaved node. In CORE, the reputation is calculated based on various types of information on each node's rate of collaboration (for subjective, indirect and functional reputation). This reputation is then used as a measurement of the trustworthiness level of a node from a routing perspective. Michiardi and Molva then used game theory in [MM03] to evaluate CORE and in particular to define a lower bound on the number of legitimate nodes in an ad hoc network when the CORE mechanism is adopted and to describe the asymptotical behavior of a selfish node that is controlled by CORE. Reputation mechanisms are quite interesting in MANETs but they are not really adapted to opportunistic networks: they are efficient only over time, for example when nodes frequently meet each other, but they are inefficient in case of high mobility networks where the frequent topology changes prevent reputation establishment.

In the category of rewarding mechanisms, there are first protocols with source-centric management of rewards such as [RFJY03, HKLM03, GA04]. In these protocols the source of messages advertises content and then rewards only intermediate nodes who spread the message towards clients interested in the advertisement. The rewards, also called Coupons, are managed by the source of the advertisement only and are useful only with respect to this source, e.g. to get a discount at the source's shop. The differences between these protocols lie in the rewarding models used or in the flexibility with respect to the number of coupons that can be used ([HKLM03] is the only solution that allows nodes to increase their chance of being rewarded if they are highly motivated and forward many messages). These mechanisms thus work similarly to loyalty programs and really target commercial application but they are not practical for generic routing mechanisms.

Rewarding mechanisms also include credit-based solutions such as Nuglets ([BH03]) and Sprite ([ZCY03]). In [BH03], Buttyán and Hubaux define two different payment models based on a new virtual currency named "nuglets". In the first model the source pays for sending a packet by loading nuglets within the packet. Intermediate nodes acquire some nuglets from the packet when they forward it and if the packet runs out of nuglets then it is dropped. In the second model, the destination pays for the packet: intermediate nodes buy packets from previous intermediate nodes and the total cost of forwarding the packet is covered by the destination. Both models require tamper-proof hardware at each node to circumvent the need for a "nuglets" management authority. In [ZCY03], Zhong et al. propose a credit-based system denoted by Sprite that relies on the existence of a third party named Credit Clearance Service (CCS). In this solution, the source pays all intermediate nodes which must then contact the CCS whenever they forward the message in order to receive their rewards from the source. Sprite requires an immediate reachability of the TTP and is therefore not suited to opportunistic networks.

The aforementioned rewarding mechanisms foster nodes to cooperate by rewarding them for providing a forwarding service: cooperation is encouraged. In [ÖSM07b] we proposed an original approach where cooperation is mandatory for active nodes. This solution is based on a hot potato approach where nodes have to take a decision of accepting a packet or not blindly, as depicted in Figure 3.1. When a node N_1 has a message M to forward, it notifies its neighbors without giving any detail about the destination of M. Neighbors should therefore answer blindly if they are interested in M or not. Suppose node N_2 is interested, N_2 sends a payment to N_1 and then N_1 sends M to N_2. If M is of interest to N_2, then N_2 paid for an interesting packet which is fair, but if N_2 is not interested in M, N_2 is incited to forward the packet to other nodes in order to get its payment back, hence cooperation is enforced. This approach is optimistic in that the authority is required only to:

- give nodes rewards for the packets they forwarded and sustain the system,
- solve conflicts among nodes which occur if node N_1 does not send M to N_2 after receiving the payment from N_2.

This protocol therefore enforces optimistic fair exchange: if a conflict occurs the authority guarantees fairness by giving each party its rightful part, but the involvement of the authority is not required in case of correct execution of the protocol. This protocol is also adapted for opportunistic networks because it is flexible: for example, nodes with low resources can decide not to receive packets at the cost of missing packets destined to themselves, or alternatively to collaborate with others to receive messages destined to themselves and forward messages for other nodes.

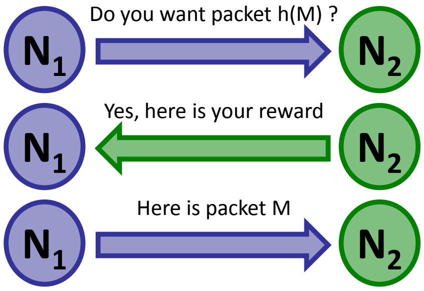


Figure 3.1: Cooperation enforcement based on the Hot-Potato principle. During the first step a hash of the message is sent only to allow N_2 to verify that it did not already receive message M, but N_2 does not learn the content of the destination of M: N_2 takes a decision blindly.

Ultimately, cooperation enforcement brings nodes confidence that they can rely on one another to perform networking operations fairly, i.e. forwarding the packet. If it is possible to establish a long term relationship, then reputation works as an indirect measurement of trust, while in case of ephemeral encounters trust is deferred on rewards and on the authority managing them. Cooperation enforcement is therefore related to trust: cooperation enforcement can be seen as a substitute of trust while cooperation incentives are a mean of providing elementary trust. Trust among peers is actually a more general concept that has implications beyond cooperation, and establishing trust is a challenging issue that we tackle in the next section.

3.2 Trust Establishment

Trust is an elusive concept in that there is no precise definition of trust. The reason is that trust is not a simple manichean concept but rather a graduated one with subjective levels. Gambetta defined trust in [Gam88] as *"a particular level of the subjective probability with which an agent will perform a particular action."* The particular action in opportunistic networks corresponds not only to security operations but also to networking ones (forwarding data, sharing data or resources). It is therefore important to define for each action and each scenario the corresponding trust assumptions: for example, in context-based communication, can a node trust its neighbors to correctly forward a message based on the context? Also, can a node trust its neighbor not to use the context of the destination for malicious purposes (such as profiling the destination)?

In traditional networks, trust relies mainly on the infrastructure: there is a dedicated infrastructure for routing (with dedicated routers) and this infrastructure is trusted by end-users to fulfill the routing task. The infrastructure is of course prone to malicious activities and attacks, but in practice this is of no concern to end users: the infrastructure should include mechanisms to protect against malicious activities or at least to recover from attacks, but users simply benefit from the network services while being oblivious to the networking aspects managed by the infrastructure. In that sense, users place their trust in the underlying infrastructure. The layered networking architecture is actually instrumental from this perspective, because it ensures that the security of the application layer does not depend on the network layer's or lower layer's security. Moreover, the infrastructure does not only perform routing operations, it also provides naming service which also simplifies the establishment of trust between users: from a human perspective, people tend to trust an easy to remember name of a famous website but it is unnatural to trust a raw IP-address. Furthermore, when higher level of trust is required, for example to access sensitive sites (e.g. banking or online monetary transactions in general), the network infrastructure can be complemented by a security infrastructure (e.g. a Public Key Infrastructure (PKI) that provides identity certificates).

This directly leads to the question of establishing trust in opportunistic networks. In opportunistic networks, there is no routing infrastructure with dedicated routers, peer nodes act as message carriers and forwarders instead. Furthermore, not only is there no naming service accessible but also identifiers are meaningless from a trust perspective because names do not necessarily mean trust if not part of a shared a priori trust relationship (which is traditionally provided by an infrastructure). Even if we assume that each node has an identity certificate to prove its identity, this still has no implication on the trust relationship: if Alice meets Bob for the first time and Bob certifies that he is Bob with a certificate, Alice has still no reason to trust Bob on anything except that he is indeed Bob. It is therefore important in opportunistic networks to first build trust among parties so that they can rely on one another.

As stated by Abdul-Rahman and Hailes in [ARH97] it is important to consider a decentralized approach of trust in distributed systems. Even though there might be a central authority that certifies some attributes, each node should be able to evaluate the amount of trust it places in another node independently of the central authority. A popular approach to establish trust in self-organized networks is to build a web of trust based on recommendations by trusted nodes as proposed by Zimmermann in [Zim95] and the Pretty Good Privacy system. This approach yet requires time to propagate recommendations and some stability in the network so that nodes build trust relationships with other nodes that they often meet but it is unfortunately not adapted to opportunistic meetings with previously unknown nodes.

Since names do not have trust implications in opportunistic networks, an alternative is to mimic the human behavior: at first encounter, humans trust individuals based on their roles or the community they belong to rather than their names. The idea is therefore to characterize communities in opportunistic networks. Communities can be either inferred in an ad-hoc mode (e.g. all the people coming to a restaurant form this restaurant's community) as proposed by Hui et al. in [HYCC07], or they can be discovered based on preset attributes in nodes profiles (e.g. all people working in EURECOM form EURECOM's community). The first approach is appealing for autonomous systems, but the trust level implied by these ad-hoc communities is rather low. On the contrary, the latter implies a higher trust level as it concerns more stable attributes, and we adopt this approach in the sequel of this thesis.

The advantage of looking at nodes' profiles is that these profiles are already used in context-based forwarding and are thus readily available. Another advantage is that it is possible to build flexible trust policies based on these profiles. Examples of such policies are:

- **Binary Trust Policy:** if two nodes share one attribute then they fully trust each other and reveal all attributes to each other. This policy is rather permissive but it is simple to express and it is reasonable if the shared attribute is of significant importance (e.g. workplace).

- **Graduated Trust Policy:** nodes exchange their profiles and determine the number of shared attributes and the matching ratio r that follows. The trust level between these nodes is then r. This policy is much more flexible as it allows nodes to further add weights to attributes to express the relative importance of attributes in the computation of the trust level. Furthermore such policy allows for a definition of indirect trust by transitivity. For example, if node N_1 trusts node N_2 at level r_2 and node N_2 trusts node N_3 at level r_3 then it is possible to consider that node N_1 trusts node N_3 at level r_2r_3.

The use of node profiles thus provides a practical framework to assess trust level in newly encountered nodes in a decentralized way. Yet since so much importance is placed on attributes, these attributes themselves need to be trusted. In other terms it is not enough for a node to claim that it belongs to a given company, the node should be able to prove so. This naturally calls for attribute certificates that were first introduced by Bussard et al. in [BCC$^+$05]: attribute certificates are defined like identity certificates except that they certify the correctness of an attribute instead of an identity. Attribute certificates also require a security infrastructure, but only offline. Of course this leads to the question of trusting the security infrastructure, but then the reasoning follows the so called "chicken or egg" causality dilemma. At some point one has to subjectively decide to trust an entity as an anchor of trust, otherwise it becomes impossible to establish and propagate trust.

To summarize, it is possible to address the trust bootstrapping issue by considering communities based on shared attributes. An implicit underlying assumption is that nodes are concerned with their communities and do not adopt malicious behaviors with their community. We define this assumption as follows:

Definition 3.2.1 ***Trusted communities assumption:*** *a node does not attempt to harm another member of his community based only on the knowledge of their shared attributes (but it might adopt a malicious behavior if it discovers additional information).*

We illustrate this assumption by a simple caricatural example. Assume Alice is working at EURECOM and is member of the classical music addict group, while Bob who is also working at EURECOM is a member of the electro dance fans group. Under the trusted communities assumption, Alice will not adopt a malicious behavior against Bob if she only discovers their shared attributes, namely that they work at EURECOM, but she would collaborate with him on the contrary because that might benefit her as well. Yet, if Alice comes to know that Bob not only works at EURECOM but also is a member of the electro dance fans group then she might attempt to harm Bob's communication (dropping all messages to the electro dance fans group for example) without breaking the trusted communities assumption.

This assumption raises the requirement for an additional building block which is private matching: nodes should be able to discover each other's shared attributes but should not learn anything about attributes that are not shared.

Once trust is established among the members of the community, it is important to assure that the overall application carried out by the trusted parties takes place in a trusted manner in the face of malicious attacks or selfish behaviour. In other words, maintaining trust requires preventing potential attacks targeting the overall application carried out by the collaboration of community members. This second aspect of trust entails the design of security and cooperation mechanisms that enforce the trust relationship during the interactions between the members. Based on the security and cooperation requirements of the application, these security and cooperation mechanisms may include classical methods like data encryption and data integrity protection or incentive mechanisms (as presented in section 3.1).

In summary, trust is a general concept that eases secure communication between users, but trust establishment is a tedious task. Similarly to human behavior, establishing trust among peers in a self-organized way is indeed possible in case of a long-term relationship by monitoring the behavior of other nodes over a period of time and then trusting them in some tasks that they are used to perform reliably. Opportunistically taking advantage of arising contacts assumes short term encounters, and in that case, trust can be obtained transitively through a security infrastructure that gives some credentials to each node. These credentials are then used as anchors by nodes to translate trust in the security infrastructure into trust between nodes. Such a security infrastructure needs to be considered offline to be adapted to opportunistic communication.

It is important to define the trust assumptions for each scenario because it impacts the deployment of security solutions. When two nodes fully trust each other they do not require additional security mechanisms to perform operations on each other's data but the communication between them needs to be secured to deal with the activities of other malicious nodes overhearing the communication: this calls for mechanisms guaranteeing the authenticity, integrity and confidentiality of messages. Moreover nodes completely trusting each other are unusual in opportunistic networking as it requires a tightly controlled environment, hence intermediate trust assumptions need to be defined such as the trusted communities assumption. Under such assumptions one needs to also protect the confidentiality of the data and privacy of the users, as it will be discussed in section 3.4.

3.3 Authentication and Integrity

Authentication and integrity are two fundamental security services in any communication paradigm. In the following, we distinguish between two types of security services:

- **end-to-end security services** refer to services provided from the source of a message to its destination without being affetced by forwarding nodes,
- **hop-by-hop security services** protects the data only during a communication between two neighbors. These services hence need to be processed at each hop of the communication.

Authenticating the source of a message is a basic requirement in conversational communication such as oblivious forwarding, and it is classically provided thanks to signatures and identity certificates. In this form, source authentication is an end-to-end service that can also be provided in opportunistic networks in the same way (see Figure 3.2). Each node needs to generate public/private key pairs and then retrieve a certificate from a Certification Authority (CA which is a trusted entity) corresponding to this pair. The message is then signed with the private key and sent along with the certificate. The destination then verifies the signature thanks to the public key of the sender which can be found in the certificate. Alternatively it is possible to use identity-based signatures ([BLS01, CC03]) which

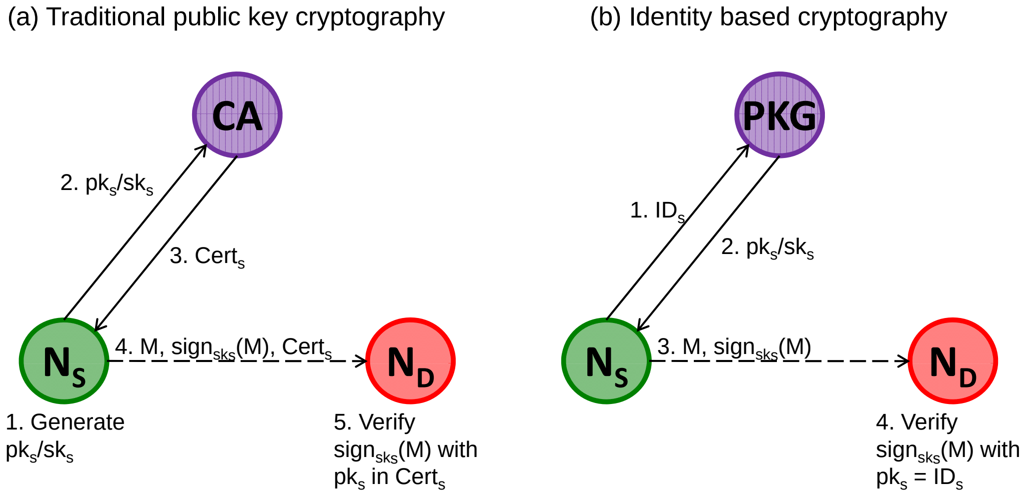

(a) Traditional public key cryptography
CA
2. pk_s/sk_s
3. $Cert_s$
N_S
4. M, $sign_{sk_s}$(M), $Cert_s$
N_D
1. Generate pk_s/sk_s
5. Verify $sign_{sk_s}$(M) with pk_s in $Cert_s$
(b) Identity based cryptography
PKG
1. ID_s
2. pk_s/sk_s
N_S
3. M, $sign_{sk_s}$(M)
N_D
4. Verify $sign_{sk_s}$(M) with pk_s = ID_s

Figure 3.2: Message integrity and source authentication with traditional public key cryptography (left) and with identity based cryptography (right). Plain arrows represent direct communication while dashed arrows represent opportunistic communication.

alleviates the need for certificates, as proposed by Asokan et al. in [AKG$^+$07]. In this case the public key of a node is its identifier, and the associated private key is computed by a Public Key Generator (PKG which is a trusted entity). The message is then signed with the private key and the destination verifies the signature by using the identity of the sender as public key. In both cases there is therefore a setup phase involving communication with a trusted entity (to retrieve the certificate or the private key respectively), but this phase is an offline step in that it needs to be performed once prior to communication; access to the trusted entities is not required during the opportunistic communication.

The advantage of using identity-based signatures is limited though, since basic identity-based signatures are subject to repudiation (because of the key-escrow possibility by the public-key generator), and the certificates required by traditional public-key signatures are sent along with the message to guarantee the authenticity of the public key: traditional public key cryptography can therefore readily be used and there is no clear advantage in using identity-based signatures. For example, the current DTN bundle security specification [SFWL10] includes a Payload Security Header (PSH) which supports authentication using a message authentication code, or a digital signature and it is possible to extend the specifications so that messages also carry the necessary certificates using which a verifier can validate a digital signature. The integrity of messages is usually provided along with authentication thanks to signatures. Classical authentication and integrity mechanisms apply well to opportunistic networks because the verification of a signature is an offline operation that can be performed asynchronously with respect to the communication and does only require information which is readily provided in the message. The challenge for such mechanisms is therefore finding more efficient signature primitives but there is no conceptual novelty, as the delay-tolerance characteristic of opportunistic networks does not have an impact on classical solutions, except from a revocation perspective.

Similarly hop-by-hop authentication to prevent tampering with messages by active attackers in the middle can be assured by adding a message authentication code or a signature as proposed by the DTNRG in the specification of the Bundle Authentication Header (BAH) ([SFWL10]).

New concepts are yet required when end-to-end authentication or integrity is not straightforward, because the messages need to be modified in their path from source to destination. This happens in two interesting cases that we describe hereafter.

First, oblivious opportunistic forwarding might require message fragmentation to accommodate specific requirements of the technology used for forwarding (e.g. bluetooth, WiFi). Therefore the message is divided into many fragments which need to be authenticated. A straightforward approach called the toilet-paper approach consists in signing each fragment such that each fragment is self-authenticating. This approach is yet undesirable because of the message size and computation overhead are linear in the number of fragments [SFWL06]. The challenge here is therefore to find a more efficient solution that enables fragments authentication prior to receiving all fragments. Asokan et al. propose an interesting approach in [AKK$^+$07] which is inspired by the Merkle hash tree concept that enables authentication of fragments with logarithmic message size and computation overhead.

Second, network coding based forwarding requires to modify packets at each node and to forward new linear combinations of received packets. The main threat in this protocol is called pollution attack: a node could inject a bogus packet in the network to prevent the correct decoding of the packets. This attack can be devastating [GR06]: introducing a single bogus packet can potentially harm the whole network. It is therefore critical to provide means to evaluate the correctness of linear combinations and the integrity of the packets. Such a problem thus requires the source to sign authentic blocks with a flexible signature scheme that enables intermediate nodes to compute a valid signature of a combination of blocks based on the signature of each block. Moreover the signature should encompass the whole linear combination which include the payload but also the coefficients used in the linear combination to guarantee the integrity of the coefficients as well. Such a scheme therefore calls for a homomorphic signature schemes as proposed first in [ÖSM07a, ÖSM07c] and later in [BFKW09].

Providing authenticity and integrity is therefore a classical issue for all communications, and classical solutions based on public-key cryptography are suitable in general. In some particular cases of oblivious forwarding though providing integrity and authenticity can be a compelling issue as it requires flexible signature scheme that support valid modification of the message: in that case integrity can not be ensured in an end-to-end fashion on the whole message. We now focus on the confidentiality and privacy issues inherent to rich forwarding mechanisms.

3.4 Confidentiality and Privacy

Similarly to authentication, confidentiality requirements can be divided in two main categories: hop-by-hop and end-to-end confidentiality.

Hop-by-hop confidentiality is required mainly to protect against eavesdropping which is especially easy to perform over wireless interfaces. It is therefore important to encrypt messages during a direct communication between neighbors. Providing such an encryption service is simple in a conversational approach where only two neighbors are involved: any cryptosystem can be used, even symmetric encryption provided that nodes first agree on a secret key. The problem is yet more complex in case several nodes are involved to take advantage of the broadcast nature of the wireless medium: in that case the sending node has to be aware of the legitimate neighbors that are interested in the message and to manage a multi-party computation to establish a group key. In all cases, hop-by-hop confidentiality can be performed interactively between a node and its neighbors while in communication range and is not affected by the opportunistic nature of the network.

End-to-end confidentiality of messages is traditionally achieved by encrypting messages either with a public key or with a symmetric key algorithm. In the latter case, an end-to-end shared key is first agreed upon through a key agreement mechanism which implies public key cryptography. When such a service is required in opportunistic networks (e.g. for a confidential conversation), applying encryption primitives used in legacy networks is not as straightforward as with signatures. The problem is that, in order to use traditional public key encryption, the sender of the message needs the public key of the receiver prior to sending the message. Fetching this public key requires access to either the destination or the Certification Authority and both cannot be assumed in case of intermittent connectivity: encryption with public key cryptography is possible only if destination's public key is already known by the source (this is the solution adopted by the DTN specification of the Confidentiality Header (CH) [SFWL10]). Traditional public key encryption mechanisms present therefore major limitations for a practical use in opportunistic networks, and one approach to solve this issue is the use of Identity-Based Encryption ([BF01]) as recommended by Asokan et al. in [AKG$^+$07]. In order to accommodate various authorities and delegation properties, Hierarchical Identity-Based Encryption ([GS02]) is investigated by Seth and Keshav in [SK05] and by Kate et al. in [KZH07]. Identity-Based Encryption uses the identity of the destination as public key in the encryption process and therefore knowledge of the public key of the destination does not require an additional communication. Contrary to authentication and integrity, the use of identity based cryptography presents major advantages over traditional public key cryptography to ensure end-to-end confidentiality (see Figure 3.3).

The previous approaches are suitable for oblivious forwarding approaches but not for the enriched forwarding proposals where the destination is not defined with a unique identifier at the source. In this case the confidentiality mechanism should be adapted to the specific way in which the destination is defined. To be more precise, in context-based approaches the destination is defined implicitly as a node which has a specific set of attributes described in the header. The encryption mechanism should therefore encompass this flex-

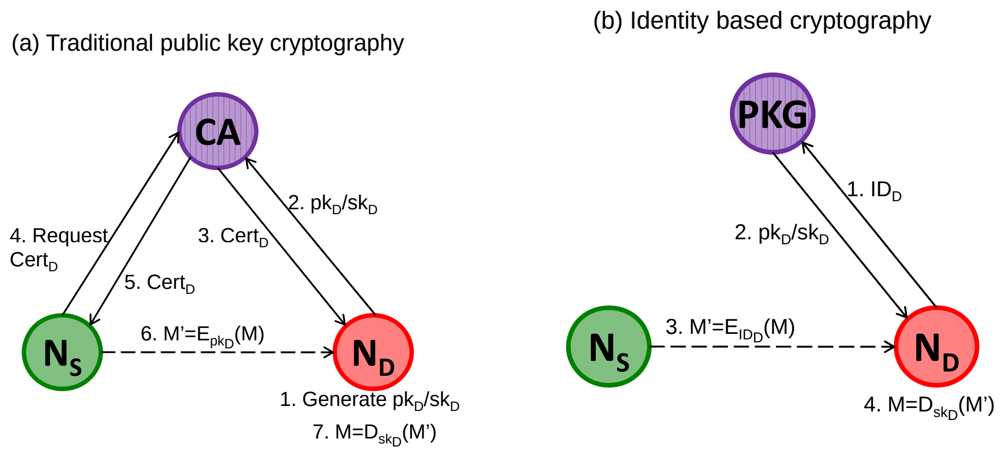


Figure 3.3: End-to-end confidentiality with traditional public key cryptography (left) and with identity based cryptography (right). The encryption operation with a key k is denoted by E_k while the decryption operation is denoted by D_k. Plain arrows represent direct communication while dashed arrows represent opportunistic communication. Step 4 and 5 on the left can be performed between the source and the CA or alternatively the destination or any other node which has the certificate $Cert_D$, but in all cases these steps are a major drawback to apply traditional public key encryption in opportunistic networks.

ible definition of the destination: it should enable any node to derive an encryption key corresponding to a set of conditions instead of a single identifier. The decryption mechanism should be possible only by nodes which verify all the conditions and therefore only nodes which verify all the conditions should be able to derive the decryption key. These requirements thus call for an extension of identity-based cryptography and we propose a solution adapted to this issue in chapter 6.

In the case of content-based communication confidentiality is not really a relevant issue because the essence of content-based communication is to disseminate a message to all interested nodes, but content-based communication raises interesting privacy issues.

Privacy is indeed expected to be a significant concern for acceptance of pervasive environments as pointed out by Opyrchal et al. in [OPA07]. Lilien et al. go even further in [LKBG06] as they believe that *"any opportunistic networking solution compromising on privacy protection is doomed to a total failure"* and that *"privacy protection is the "make it or break it" issue for opportunistic networks and pervasive computing in general."*

There is no commonly agreed-on definition of privacy as this is a fuzzy concept encompassing numerous aspects. We could globally state that information privacy is the ability for a user to control which information he keeps as secret and which information he reveals, and in the latter case the user should be able to control who can access that information. Privacy issue is therefore of great importance in databases management and companies or public administrations are required to have transparent privacy policies in accordance to national or international laws such as the european directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data [Dir95]. This aspect of privacy is yet out of the scope of this thesis as opportunistic networks are rather ephemeral networks aiming at enabling communication between mobile nodes in challenging conditions, therefore privacy of stored data is not of primary concern.

The main privacy concern in opportunistic networks is communication privacy. This includes confidentiality of exchanged messages that we described in the beginning of this section, but also privacy of the sender and receiver. A sender might indeed offer some sensitive content if he can remain anonymous while a receiver is legitimate in requiring that other nodes cannot determine what type of content he is downloading. Both sender and receiver (or any intermediate node) may probably not want that the communicating device can be linked to the human being using it. It makes a significant difference whether a node learns that "some node is interested in geek goodies" or "device-ip 123.189.157.46 is interested in geek goodies" or even "device-ip 123.189.157.46 belonging to Alice is interested in geek goodies". In general there are three distinguishable degrees to classify user identifiability [PH09]:

- **Identity:** A user that communicates with others and reveals any piece of information (e.g. full name, social security number) that can be used to clearly identify him is said to work under his identity.
- **Pseudonymity:** This is the ability to prove a consistent identity without revealing a user's real identity, instead using a pseudonym. Whether a pseudonym can be linked to the real identity of a user depends on the entity which is concerned and the frequency and variety of use of the pseudonym. The harder it is to reveal the pseudonym of a user, the closer we are to the state of not being identifiable at all, thus acting anonymously.
- **Anonymity:** Anonymity is the ability to remain unidentifiable within a set of users. A user acts anonymously if it is impossible to reveal his identity.

The level of identifiability and privacy depends on the application, the point of view (sender, receiver, intermediate node, outside observer) and the level of trust between entities. Based on our previous work in [SÖM09a], we define a general framework of privacy models with respect to a private data D_1 belonging to node N_1 and that needs to be processed by N_2 (for forwarding purposes for example) as follows:

- **model 1, privacy oblivious**: this model refers to the case where N_1 does not require privacy protection at all, N_2 has access to the whole content of D_1 in the process. This is a model that fits situations where the trust level is very high and therefore nodes trust each other in not exposing their privacy. In such cases nodes usually directly use their identity to further increase the trust level.
- **model 2, binary privacy**: in this model a node N_1 completly trusts some nodes and distrusts other. Therefore if N_2 belongs to the trusted group from the point of

view of N_1, N_2 has access to D_1 else N_2 is not granted access to D_1. This model hence refers to the binary trust case: for example, N_1 and N_2 they fully trust each other if they share one common attribute.

- **model 3, adaptable privacy**: in this model, the level of privacy depends on nodes' relationship: N_1 trusts N_2 partially. The trust relationship can for example be based on set of community memberships as explained in section 3.2. In this case, N_2 should be able to discover only part of the data D_1 based on the trust level. Since trust is not absolute in this case nodes use pseudonyms to prevent identification and thus privacy exposure.

- **model 4, full privacy**: as opposed to model 3, this model refers to the case where nodes do not trust each other at all. In this case N_2 should be able to process the data D_1 without having access to its content.

The nature of the private data D_1 (context or content information) as well as the level of trust and privacy achievable depends on the scenario considered.

In the case of context-based forwarding, the context of the message is directly linked to the profile of the destination and is therefore considered as private. The context should therefore be protected and only the information about shared context should be revealed: this calls for an encryption of the context. This issue raises the problem of computation on encrypted data: is it possible to design an encryption mechanism that enables encryption of the context while allowing the necessary computation on this encrypted context? In order to ensure the confidentiality of the message's context and thus privacy of the destination, nodes in context-based forwarding need indeed to be able to evaluate the amount of shared context between their profile and an encrypted context without knowledge of the encryption key and without decryption of the context. This requirement is an original and challenging open research problem that we tackle in chapter 7.

In the case of content-based forwarding, preserving the privacy of users mainly consists in protecting their interests: users want to receive content corresponding to their interests without revealing them. User privacy and forwarding present therefore conflicting requirements: the first requires encryption of the interests, while the second requires access to the filters. This raises again the problem of computation on encrypted data: is it possible for users to declare their interests in an encrypted way, such that other nodes cannot discover their interests but still correctly forward content? Note that the content should also be encrypted otherwise it would indirectly reveal the interest of users. Hence, the challenge for intermediate nodes is twofold:

- is it possible to build forwarding tables based on encrypted data?

- is it possible to perform look-up of encrypted content in the encrypted forwarding tables?

Privacy issues in rich forwarding approaches is the main focus of this thesis and therefore we expand more on this topic in the next two parts.

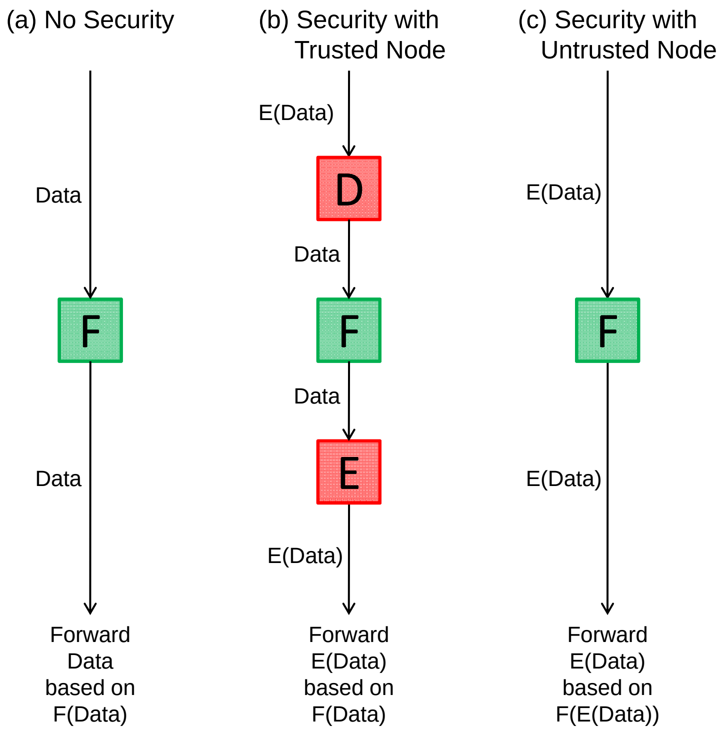

(a) No Security
(b) Security with Trusted Node
(c) Security with Untrusted Node
Data
F
Data
Forward Data based on F(Data)
E(Data)
D
Data
F
Data
E
E(Data)
Forward E(Data) based on F(Data)
E(Data)
F
E(Data)
Forward E(Data) based on F(E(Data))

Figure 3.4: Security, trust and processing of data. In figure (a), no security is required the data is processed in clear. In figure (b), the data is protected during communication between trusted nodes. Trusted nodes receive an encrypted data, decrypt it, process it, reencrypt it and forward it. Finally in figure (c), security is enforced and there is no trust relationship, therefore nodes receive an encrypted data, process it (with computation on encrypted data) and forward it without decrypting it at any time. The blocks D,E, and F stand respectively for decryption, encryption and forwarding algorithm.

3.5 Summary

Following the new forwarding concepts, all security aspects of secure communication need to be revisited. Providing support for secure communications between members of a community as well as incentive mechanisms to encourage peers to perform a fair share of basic networking operations like forwarding packets and requests are indeed key factors that need to be addressed in order to make the opportunistic networking paradigm realistic. Establishing trust and fairness among partners in the community however does not protect against attacks aiming at tampering or disclosing sensitive information being forwarded between peers or subverting the basic communication system such as denial of service attacks. While basic security services that provide message integrity and confidentiality can be based on traditional security mechanisms, forwarding requests and messages without accessing their content is a hard problem that calls for solutions based on techniques for computing with encrypted functions.

To be more precise, as a result of privacy and confidentiality requirements, data that is transferred by the communication mechanisms often needs to be encrypted in an end-to-end way. Most trust establishment and cooperation enforcement mechanisms also require hop-by-hop data encryption as means of preventing data access to untrusted or non-cooperating parties. Encryption on the other hand will hinder basic communication mechanisms such as packet forwarding that need to be able to parse at least the messages' headers, and more in the case of enriched forwarding strategies. The conflicts between security and communication functions are increasing with the amount of data required by the forwarding protocol: oblivious protocols are the easiest with this respect, followed by context-based protocols and finally content-based approaches which require access to the very content of messages. The potential conflicts can be solved by allowing networking mechanisms to operate on encrypted data without decrypting them, as presented in Figure 3.4. Computation on encrypted data is yet a challenging task, and existing solutions lack efficiency and flexibility. The focus of this thesis is therefore to propose practical solutions to enforce privacy and more generally secure communication in opportunistic networks with tailored mechanisms to perform computation on encrypted data, first in the framework of context-based forwarding, then in the even more complex case of content-based forwarding.

In the next chapter we present a practical implementation of some security services for a concrete architecture providing opportunistic communication.

Chapter 4

Haggle Architecture

Haggle is a European Union funded project in Situated and Autonomic Communications, which is one of the most notable initiative in the area of opportunistic networks. One of the main objectives of Haggle is to design a new autonomic networking architecture to enable communication in the presence of intermittent network connectivity, which exploits opportunistic communications (i.e. in the absence of end-to-end communication infrastructures).

In this chapter we present the Haggle node architecture and more particularly the security features that we designed and implemented.

4.1 Overview of the Haggle Node Architecture

The Haggle architecture was developed through incremental specifications: INFANT [Ge06], CHILD [DGN+07], YOUNG [BGM+08] and ADULT [TDG+09]. In this section we describe the key principles that are common to all Haggle versions, and then present an overview of the architecture of the ADULT Haggle node.

4.1.1 Architectural Invariants

The INFANT architecture ([Ge06]) identified a set of key principles of Haggle's operation when faced with a network environment featuring intermittent connectivity and opportunistic communications (in particular the absence of an end-to-end communication infrastructure). These key principles include:

- data- and people-centric communication,
- message switching instead of packet switching, i.e., using Application Data Units (ADUs) as the native data format,
- the reliance upon application layer information for determining how to forward information in the network,

- asynchronous and late-binding operation,
- disseminational communication by default (conversational communication between two parties is a special case of dissemination),
- transparent support for multiple transfer methods,
- resource management for prolonged untethered operation.

The above principles are the architectural invariants [ABE+04] that have guided the design of Haggle throughout its successive specifications. The following sections give an account of the major developments over the different incarnations of the Haggle architecture.

4.1.2 Main Components of the Haggle Architecture

The Haggle node architecture is a middleware that interacts directly with the application on one side, and the communication interface on the other. All the versions of the Haggle node depart from the traditional layered network and feature a collapsed architecture.

The first Haggle design (INFANT [Ge06]) specified a modular structure for the architecture, consisting of managers that act within predefined domains of responsibility. A manager solves specific tasks in the context of its domain, such as detecting neighbors, or exchanging data with them. A proof of concept of INFANT Haggle was implemented in Java.

The CHILD [DGN+07] architecture refined the core concepts of INFANT, in particular the interaction between the managers by introducing an event based interaction system, that mitigates the performance issues encountered in the INFANT Haggle implementation. To accommodate this rather fundamental change, the architecture was completely redesigned and subsequently reimplemented in the C++ language for multiple platforms (Windows/Windows mobile, Mac OS X, Linux). Instead of having the managers interact with each other directly, CHILD introduced an event model in which each manager generates and consumes a set of public events. A manager thus only knows about the events it is interested in, and is at the same time oblivious to who is generating them.

Whilst the step from INFANT to CHILD was quite large, the architectural changes in the subsequent versions are more incremental. The YOUNG [BGM+08] and ADULT [TDG+09], refined certain aspects of the CHILD Haggle architecture and introduced new concepts such as search-based networking. The resulting architecture consists at its core of four types of components (see Figure 4.1):

1. **HaggleKernel:** The HaggleKernel is a minimal event queue that coordinates the communication among Managers. The HaggleKernel also listens on socket interfaces for incoming data.
2. **Datastore:** A central repository with (application) data and information about nodes and their Interfaces. More precisely, the Datastore is built around four types of data sets:

- The *Data* set contains a virtual representation of persistent (application) Data Objects (see section 4.1.3).
- The *Node* set contains a virtual representation of a device or a user, depending on the node descriptions. The node description points at the node's associated attributes and communication interfaces.
- The *Interface* set represents the local node's physical communication interfaces.
- The *Attributes* set contains arbitrary name-value pairs that are associated to Data Objects or descriptions (i.e. in their metadata).

The Datastore is read-accessible by all Managers.

3. **Managers:** Managers are responsible for maintaining a certain part of the Datastore or specific tasks like managing communication interfaces or security related functions.

4. **Modules:** Managers can make use of Modules to implement specific functionality related to particular algorithms (for example, different forwarding algorithms, protocols, or communication interfaces).

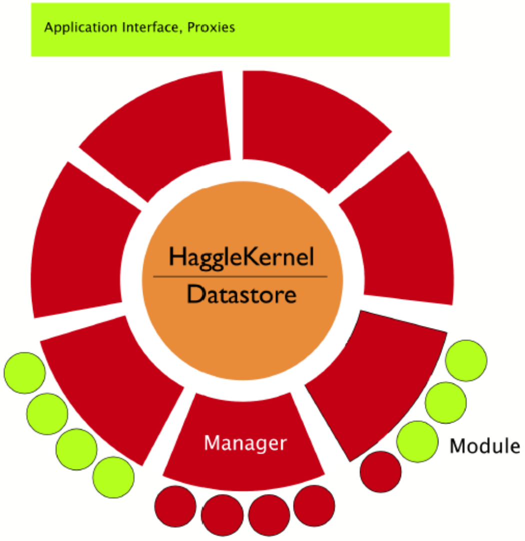


Figure 4.1: Overview of the Haggle architecture.

Communication with Haggle is done through socket interfaces, irrespective if it is another Haggle node or an application. The data format used by the Haggle node for the communication consists of well defined Data Objects that are described in the next section.

Table 4.1: Example of a Data Object

DO-ID	12
DO-Type	Data
Category	Picture
Location	Venise
Data	DSC18.jpg

4.1.3 Data Objects

The Haggle architecture is built around Data Objects (DO).

Data Objects are a general representation of all information inside and outside Haggle. They comprise a list of attributes, where an attribute is a type-value pair. Data Objects include:

- Application data (e.g. a binary stream or a text string), where attributes are used to annotate the application data: they can be keywords from the data content, or can also include an additional descriptive function (e.g., location where a picture is taken, data type, data size). DOs describing application data are persistent.
- Control messages, which are composed by attributes only and are non-persistent.

All Data Objects consist of:

1. a mandatory header (in XML) consisting of a number of attributes that represent the metadata part of a DO,
2. optional application data (payload).

Attributes that constitute metadata are typically identifier, data type, data size, keywords and addresses. To simplify, we thus consider in the sequel of the chapter that a DO can be represented in a table as depicted in Table 4.1.

4.1.4 Haggle Managers

Managers are responsible for maintaining a certain part of the Datastore or specific tasks like managing communication interfaces or security related functions.

A Manager registers at the HaggleKernel. It generates events and registers its interest in events. It can register a socket and read/write on it (e.g., for communication with other components or other nodes). Managers and applications can also register filters that "listen" for certain attributes (or links between attributes) and they are notified whenever a DataObject with the corresponding attribute is stored in the Datastore.

The structure of the Haggle node is generic, thus new managers can be created if needed. The default setup consists of the following managers:

1. The DataObject Manager, which is responsible for Data Objects and their Attributes, their storage and interpretation.

2. The Node Manager, which is responsible for maintaining the view (attributes' values) on other nodes. In particular it maintains the current status of other nodes (available or not, visible or not).

3. The Forwarding Manager, which is responsible for finding suitable next-hop nodes to forward Data Objects to (or to find Data Objects for which a specific node could serve as next hop). The Forwarding Manager can make use of different Forwarding Algorithms for that task.

4. The Connectivity Manager, which is responsible for managing and configuring the set of communication interfaces such as Bluetooth, WiFi, GPRS, etc. The Connectivity Manager also updates the Interface Datastore.

5. The Protocol Manager, which is responsible for sending and receiving data between Haggle nodes, and between the Haggle core and other components (e.g., Application Interface).

6. The Resource Manager, that monitors the resources (e.g. energy, storage) and the context, and issues resource policies (e.g. high, medium or low resources).

7. The Security Manager, which is responsible for all security related features and is described in more details in the next sections.

4.2 Security in the Haggle Node Architecture

As previously mentioned, the Haggle node architecture has been built in an incremental way. The early implementation of the INFANT Haggle, did not take security issues into consideration. Aware of the importance of security, we advocated for a dedicated Security Manager [ÖSM07a]. We implemented an early version of this manager as an extension of the INFANT Haggle in Java, and then ported it and improved it for the new event-based architecture in C++ [ÖSTB08, ÖS09]. The main difference is that the Security Manager based on the INFANT Haggle, was providing interfaces for each other Manager, while, in the new architecture, security related events are defined to achieve a more coordinated interaction between managers.

4.2.1 Design Considerations

Since the CHILD stage of development, Haggle nodes integrate a Security Manager, which is responsible of all security modules and services. The main security requirements raised by Haggle come in response to the security issues in Opportunistic networks described in chapter 3: cooperation enforcement, trust establishment, integrity and authenticity, confidentiality and privacy.

When considering the spectrum of security requirements in Haggle, we observe that there is a strong dependency between security and communication services:

- Haggle Managers and the application itself make requests to the Security Manager for some security services: the Managers are considered as users of the Security Manager. Each Manager may need different and dedicated security services. For example, in the case of a cooperation enforcement scheme based on reputation, the Forwarding Manager has to determine which of its neighbors is the most trustworthy to deliver the packet. To this extent, it requires an evaluation of the reputation of the neighbors from the security manager. There are also a lot of security primitives that are useful to many Managers if not all of them, such as a secure hash function.

- In order to achieve such security goals, the Security Manager in turn may require some communication services by other Managers in order to accomplish its security service. In the example illustrated in Figure 4.2, in order to encrypt a certain packet for the application, the Security Manager of a Haggle Node 1 may wish to communicate with the Security Manager of Haggle Node 2 in order to establish a shared key. In such case, other Managers should help the Security Manager to build the packet and forward it to Haggle Node 2: here, the Security Manager is considered as a user of the other Haggle Managers.

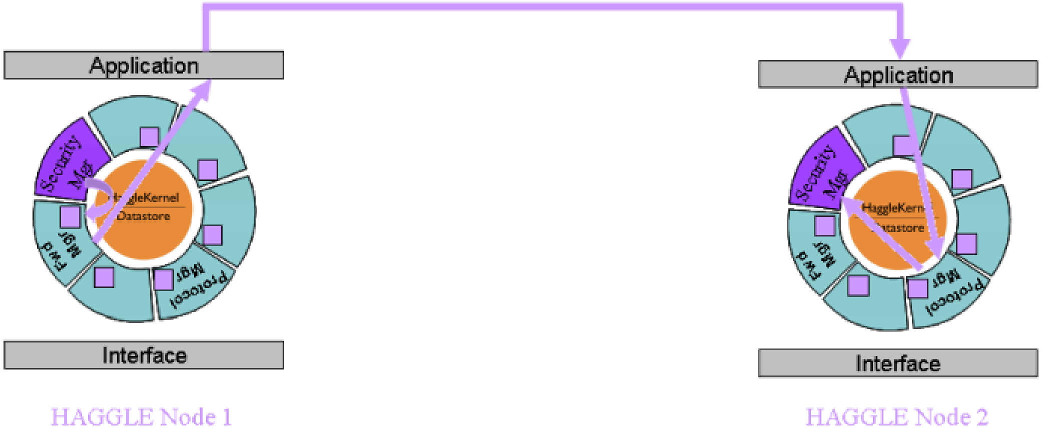


Figure 4.2: Security Manager as a user of the communication services.

This dependency between security and communication services is demanding from a task scheduling point of view. Thus implementing this architecture through well defined interactions between managers is difficult and could result in situations where managers are blocked waiting for a specific response thus considerably slowing the communication process. The event-driven architecture is more adapted as it runs the processes concurrently and without the requirement for a centralized scheduling unit, thus removing the possibility of dead-locks.

Moreover, concerning classical security requirements such as confidentiality and integrity, we distinguish between two security levels. As previously mentioned, data are indeed handled as Data Objects with several attributes. However, data ready to be sent are serialized in a byte array by the protocol manager before reaching the transmission interface. In a way, we can say that there are two levels of data and that's why we consider two levels of security as well. Applications or Managers may need to add a security feature to an attribute of a Data Object, but the Protocol Manager may also need the same features for serialized data. These are two distinct and very different problems that need to be addressed differently, hence the distinction between two security levels:

- **Application level**, where the security functions are applied to a certain subset of attributes.

- **Protocol level**, where the packet is protected as a whole on its road between nodes.

Note that legacy networking also distinguishes between several levels of security. For example, in the Internet, applications provide some security features, the transport layer includes an optional security mechanism (TLS), the network layer can add its own level (IPSec) independently of the previous ones and so on. The collapsed architecture of Haggle thus simplifies the management of security by regrouping all security features for the communication middleware and the application at the same level.

In the application level security, the data is protected from the source to the destination: this is an end-to-end security service. In the protocol level security the data is protected during its transmission between two consecutive hops (to avoid eavesdropping or man-in-the-middle nodes). Protocol level security is thus a hop-by-hop security service.

In the latter case, the Security Manager receives the serialized data from the Protocol Manager and modifies the received byte array in order to integrate additional security information. This process is an encapsulation as in IPSec tunnel mode: the resulting secured serialized data is retransmitted to the protocol manager for further forwarding. The neighbor node needs to de-encapsulate the received secured serialized data before being able to process it. Protocol level security thus simply requires classical security functions (encryption for confidentiality and MAC for integrity and authenticity) that have to be implemented in accordance to the serialized data format but it does not imply original security design as in application level security. In the sequel of this chapter we thus focus on application level security.

4.2.2 Attribute Certificates

In the Haggle communication model, Haggle nodes exchange a list of their attributes (node description) and look for some shared ones in order to learn more on context and make good forwarding decisions. Context might include some sensitive information that should be private to a certain degree. Therefore, only authorized nodes (for example nodes belonging to the same community or sharing the same attributes) should access the content of the packet. Moreover, attributes may also be falsified and therefore authentication becomes a strong requirement. Nodes should prove in a certain way that they hold the correct attributes or they belong to a certain community.

In order to cover these two important problems, we proposed the introduction of Attribute Certificates that allow nodes to prove their community membership and that may ensure some privacy within the community. An Attribute Certificate is similar to a classical identity certificate except that it proves the authenticity of an attribute instead of an identity (the identity is not even mentioned in Attribute Certificate). An attribute certificate thus allows a node to prove that it really has the declared attributes without revealing its identity.

4.2.2.1 Structure of Attribute Certificates in HAGGLE: HaggleCertificates

An Attribute Certificate, named a "HaggleCertificate" in the Haggle architecture, is defined as a Data Object that has the following attributes:

- **AttributeName** and **AttributeValue**: the type of the attribute and its value characterizing the node as it is defined in a generic DataObject,
- **PublicKey**: for each Attribute Certificate, a node stores a pair of public and private keys corresponding to this certificate. In order to verify the validity of a signature, the public key has to be part of the certificate,
- **Issuer**: the identity of the entity that guarantees that the node owns the claimed attributes, by signing this information,
- **Signature**: the signature of the Issuer generated over the other attributes of the Haggle-Certificate,
- **Validity**: the validity of the certificate that defines until when the certificate can be considered as valid if it is correctly verified.

Some other attributes enumerated in the following, are locally defined and are used by the nodes internally for purposes like certificate verification:

- Owner: this attribute is defined in order to be able to easily retrieve the certificates of a specific node. This attribute can be the MAC address of the node or its hashed identity as it is stored in the Node Description,
- Pubfilename and Privfilename: keys corresponding to a certificate are stored in files. Therefore, there should be a link between a certificate and the path where the corresponding keys are stored,
- Verified: this field is useful for the node that stores a certificate which is not his. It states if the certificate is already verified or if it is under verification process,
- Challenge: this field is again defined by the node: whenever the verification of the certificate is needed, the nodes need to define and store the challenge to be signed by the owner of the certificate.

The structure of Attribute Certificates are defined as a specific class inheriting the properties of the generic Data Object class in the HaggleCertificate.h and HaggleCertificate.cpp files to be used as a library.

4.2.2.2 Key Management for HaggleCertificates

Since the generation and the verification of Attribute Certificates are based on the use of some private and public keys, we propose to group these keys in three different folders:

- **personal keys:** Whenever a HAGGLE node requires a new HaggleCertificate, it first needs to generate a pair of public and private keys. These keys are stored in the folder *mykeys* and are named using the Owner's ID, the Attribute Name and the Attribute Value as defined in the HaggleCertificate. For example if a node "AAAA" would like to request an Attribute Certificate where the AttributeName and AttributeValue respectively are "EURECOM" and "Student", the generated public and private keys will be stored in the file "AAAA_EURECOM_Student.PubKey" and the file "AAAA_EURECOM_Student.PrivKey" respectively.

- **other nodes' keys:** HAGGLE nodes still need to store other nodes' public keys in order to be able to verify the validity of some signatures based on Attribute Certificates. Similarly to personal keys, these public keys are stored in the folder *otherkeys* and files are named using the ID of the owner of the certificate, the Attribute Name and the Attribute Value. For example, if the neighbor node's id is "BBBB" and if it claims to have the Attribute ("EURECOM", "Student"), the public key received together with its HaggleCertificate will be stored in the file "BBBB_EURECOM_Student.PubKey". There are no private keys for certificates of other nodes.

- **Issuers' keys:** In order to verify the validity of a HaggleCertificate, every node stores the public key of the Issuer who generated the certificate. These public keys are stored in the folder *issuerkeys* and the files are named according to the identity of the Issuer. Moreover, if the node itself is considered as an Issuer for some types of attributes, then it of course stores the private key in addition to its public key in a ".privKey" file.

The private keys are encrypted to protect them against malicious access by third parties. The encryption key is known to the Security Manager only, such that any operation requiring the private keys has necessarily to be performed by the Security Manager.

4.2.2.3 Storage of HaggleCertificates

In order to easily retrieve certificates whenever needed, a new online Certificate Repository named as *CertificateStore* is defined. Thanks to this new repository it is easy to retrieve any certificates with a given criteria. This repository is implemented in separate files called CertificateStore.h and CertificateStore.cpp.

DO-ID	12
DO-Type	Data
Category	Picture
Location	Venise
Data	DSC18.jpg

DO-ID	12
DO-Type	Data
Category	Picture
Location	$\mathcal{E}_{k_{12}}$(Venise)
Data	DSC18.jpg
Encrypted	Location
Key_ID	12

Table 4.2: Application level encryption. The security manager catches an event EVENT_TYPE_ENCRYPT for the Data Object on the left and the attribute Location. The security manager thus encrypts the Location with a key that is shared with the destination. It adds two attributes to the Data Object: an attribute to indicate which attribute was encrypted and an attribute that provides an identifier of the key which was used for the encryption. Both these attributes are useful for the Security Manager of the destination when performing decryption. The resulting updated Data Object (on the right) is then linked to en event EVENT_TYPE_DECRYPTED launched by the Security Manager.

The structure of this new CertificateStore is very similar to the one of the NodeStore defined for the retrieval of node descriptions in the main HAGGLE Architecture. Nodes can add or remove Attribute Certificates using the `add` or `remove` methods respectively. The `retrieve` method defined in the same file allows the retrieval of HaggleCertificates based on different sets of parameters: indeed, given an Attribute Name together with an Attribute Value, one can retrieve all HaggleCertificates whose AttributeName and AttributeValue attributes match with input values. Moreover, one can also retrieve all certificates based on a node ID thanks to the new `retrievebyNode` method.

This concludes the description of the functionalities of HaggleCertificates, and we now explain their integration and the practical implementation of the Security Manager.

4.2.3 Implementation of the Security Manager

In this section, we describe the main Security Manager implemented in the "SecurityManager.h" and the "SecurityManager.cpp" files.

4.2.3.1 Basic Functionalities

By basic functionalities of the Security Manager, we mean encryption and decryption for confidentiality, and signature and signature verification for integrity and authenticity. These operations are performed by the Security Manager on one attribute at a time, upon request from other Managers. To enable these capabilities we thus first defined related security events:

- EVENT_TYPE_ENCRYPT,

- EVENT_TYPE_DECRYPT,
- EVENT_TYPE_SIGN,
- EVENT_TYPE_CHECK.

Only the Security Manager is interested in these events. When such an event is raised, the Security Manager thus catches it, and performs the required operation on the Data Object linked to the event. Table 4.2 shows an example of processing of an EVENT_TYPE_ENCRYPT event. The core primitives (encryption, decryption, signature, verification) call OpenSSL [Ope99] functions with the appropriate parameters. Once the processing is finished and the Data Object updated, the Security Manager launches events corresponding to the operation, which are, respectively:

- EVENT_TYPE_ENCRYPTED,
- EVENT_TYPE_DECRYPTED,
- EVENT_TYPE_SIGNED,
- EVENT_TYPE_CHECKED.

These events are caught by interested Managers (Protocol, Data, Forwarding, and Node Manager) for further processing.

Note that in the previous description we assumed that the Security Manager already had a relevant key to perform the considered security operation. In the other case, the Security Manager needs to setup a key with the destination, hence the Security Manager creates a Data Object for key agreement or to obtain a certificate and requests the forwarding of this Data Object by launching a SEND_DO event (the Security Manager becomes a user of the communication middleware as explained previously).

The event-based interaction between the security manager and the other managers is summarized in Figure 4.3.

4.2.3.2 Management of Certificates

In order to securely discover neighboring nodes, a node first needs to own valid Attribute Certificates. In this section, we describe the generation of HaggleCertificate performed by Issuers and further provide details on how a HAGGLE node requests a HaggleCertificate. The process of obtaining a HaggleCertificate is as follows:

1. Whenever a node (the Requester) requests a certificate, the Security Manager first generates a pair of private/public keys using the function *generateCertificateKey*: first the pair of public/private keys are generated thanks to the inherent RSA generate key function defined in the OpenSSL [Ope99] cryptographic library, then these keys are stored as previously described in the folder *mykeys*. Then, a new Data Object is created and this Data Object includes the node's identity, the Attribute Name

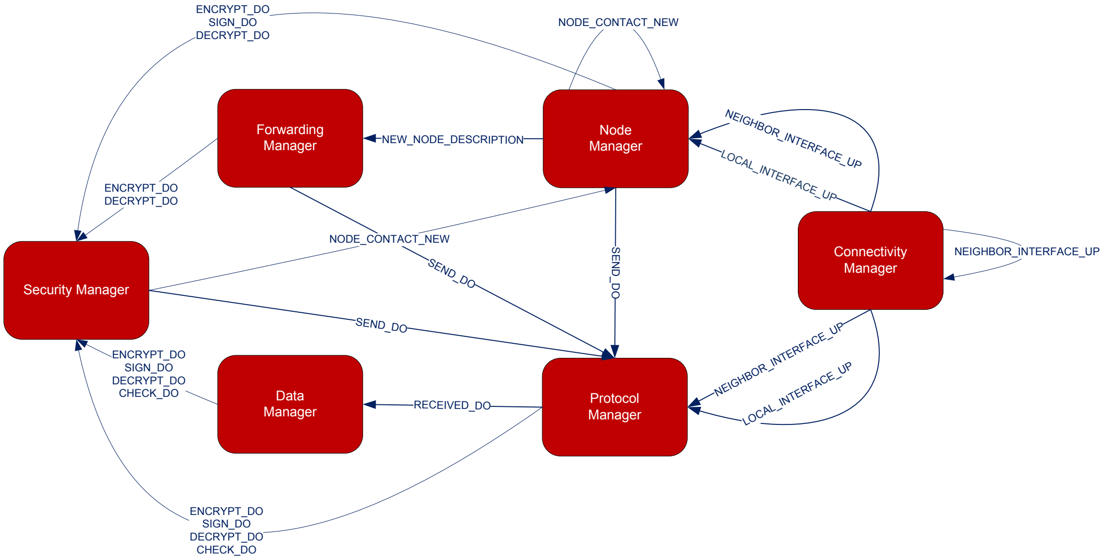


Figure 4.3: Interaction between managers for the basic security services.

and the Attribute Value corresponding to the requested HaggleCertificate and the generated public key. In order to differentiate this request from other Objects, the dedicated Data Object includes the attribute ("IsSecurity, CertificateRequest") as well. The Security Manager finally launches the EVENT_TYPE_SEND_DO event and other Managers ensure the correct forwarding of the request.

Moreover, in order for an Issuer to be able to differentiate certificate requests from other packets, its Security Manager adds a specific filter in its filter list in order to declare its interest on any Data Object with the ("IsSecurity, CertificateRequest") attribute. This filter is defined as FILTER_CERTIFICATE_REQUEST in the "SecurityManager.cpp" file.

2. When such a Data Object is received by the Issuer, the *onCertificateRequestReceived* function is executed to process the request. In particular, the SecurityManager verifies that the Requester owns the corresponding private key and then uses the *generateCertificate* function to generate a certificate.

   The verification that the requester owns the corresponding private key, is a classical challenge-response exchange: the Issuer sends a nonce (chosen randomly) to the Requester, the Requester signs this nonce with the private key and the Issuer verifies the signature with the corresponding public key.

   If this operation succeeds, the Issuer uses the *generateCertificate* function whose input parameters are the Attribute Name, the Attribute Value, the Public key corresponding to this attribute, the Issuer's ID, the Validity parameter and the name of the file where the Issuer's private key is stored. This function first concatenates the HaggleCertificate attributes, namely, {AttributeName, AttributeValue, PublicKey, Issuer, Validity}, then retrieves the private key stored in order to sign this string using the RSA signature algorithm. Since the basic RSA sign method defined in the OpenSSL cryptographic library does not take a string as an input parameter, a dedicated *SignString* function has been defined.

Moreover, the generation of the certificate can also be based on encrypted attributes. Indeed, in order to ensure privacy, attributes may sometimes be encrypted. In this case, the Issuer defines or retrieves the symmetric key corresponding to the attribute and encrypts the Attribute Name and the Attribute Value using this key. Then, the generation of the signature is performed over these encrypted information. This operation is implemented in the *generateSecureCertificate* method of the SecurityManager.

In both cases, at the end of the process, the Issuer outputs a HaggleCertificate corresponding to the (possibly encrypted) requested attribute. The Issuer then constructs a Data Object including the generated HaggleCertificated and the Attribute ("IsSecurity, CertificateReply"), and sends it back to the Requester by launching a EVENT_TYPE_SEND_DO event.

3. The Requester adds a FILTER_CERTIFICATE_REPLY filter to be warned when the reply to its request has arrived and to execute the function *onCertificateReplyReceived* that mainly ensures the verification of the validity of the certificate and its storage in case of success.

Note that a HaggleCertificate can be considered as valid only if the corresponding Issuer provides a correct signature. Thus the HaggleCertificate has a trust implication commensurate with the trust that nodes grant to the issuer. In particular a HaggleCertificate has no value for nodes which do not know the public key of the Issuer.

4.2.3.3 Securing Node Description

In Haggle, whenever two nodes meet, they first exchange their respective Node Descriptions. These Node Descriptions include attributes of nodes and thus need sometimes to be secured.

Whenever an application requires the protection of its attributes, the Node Manager launches the EVENT_TYPE_SECURE_NODE_DESCRIPTION event in which only the Security Manager is interested. The Security Manager then performs some security operations on the Node Description and then launches the EVENT_TYPE_SECURED_ NODE_DESCRIPTION event. The Node Manager catches this event and is then able to send this "secured" Node Description to its neighbors as defined in the reference architecture.

The security operations performed by the Security Manager depends on the authenticity requirements and the privacy model considered. The current implementation features two operations:

- authentication: the Security Manager can authenticate the attributes of the Node Description. To this extent, the Security Manager looks if Attribute Certificates exist for the corresponding attributes using the *retrievebyNode* function defined in the *CertificateStore*.

- privacy preservation: if required, the Security Manager encrypts the attributes in the Node Description if a corresponding "secure" HaggleCertificate exists.

The Security Manager thus selects relevant HaggleCertificates and integrates them in the Node Description. The Attribute ("IsSecurity, Certificates") is included as well.

When neighboring nodes receive Node Descriptions including HaggleCertificates, the Security Manager is warned thanks to the FILTER_CERTIFICATES filter which is activated whenever a DataObject includes a certificate. The Security Manager retrieves the certificates from the received DataObject and then verifies the validity of the certificates before starting the challenge-response protocol. A challenge-response protocol is needed in order to verify if the member is the one which claims to have the public keys corresponding to the received certificates. The corresponding messages are filtered thanks to the FILTER_CERTIFICATE_CHALLENGE and the FILTER_CERTIFICATE_RESPONSE filters. In fact, upon reception of a request, a node sends a nonce and further verifies the signature on this particular nonce. If the verification succeeds, then the node accepts and stores the certificates.

We now present a scenario that demonstrates the practical use of the aforementioned functions.

4.2.4 A practical Scenario

In this scenario we assume for the sake of simplicity that there is a unique certificate Issuer (a Trusted Third Party) and that the public key of this issuer is known by all other nodes.

This scenario revolves around the concept of trusted communities. We thus assume that nodes belong to trusted communities, where each community is defined by an attribute, or to be more precise by an Attribute Name, Attribute Value pair. Nodes fetch HaggleCertificates corresponding to their communities from the Issuer in a setup phase. There are two privacy levels for these communities:

- either the community is public, in which case the Attribute Name and Value are in clear in the HaggleCertificate which is generated through the *generateCertificate* function,

- or the community is private and in this case the Attribute Name and Value are encrypted using a key k_I known to the Issuer only and the certificate is generated through the *generateSecureCertificate* function.

During the opportunistic communication phase, we assume that nodes do not contact the Issuer further.

Table 4.3: Example of a Data Object forwarded only through the community of EURECOM staff

DO-ID	12
DO-Type	Data
Category	Picture
Location	Venise
Data	DSC18.jpg
Community	$(Workplace, EURECOM)$

When a node wants to send data it trusts only its community to forward it. Therefore the data is included in a data object and it specifies a required community to be forwarded. Building on the example presented in Table 4.1, the Data Object containing a picture of Venise would contain an additional attribute describing a community such as employees of EURECOM. The new Data Object is presented in Table 4.3.

This Data Object is forwarded only to employees of EURECOM. To be more precise, when a node encounters a neighbor, it requests from this neighbor a HaggleCertificate corresponding to the attribute $(Workplace, EURECOM)$. If the neighbor owns such a HaggleCertificate, then it sends it to the node which verifies its validity by:

- Verifying the signature of the Issuer in the HaggleCertificate thanks to the public key of the Issuer,

- Checking that the neighbor is the rightful owner of the HaggleCertificate by launching a challenge-response protocol.

If all the above succeeds, then the node forwards the Data Object to the neighbor otherwise the communication aborts. Note that at the protocol level, the communication is protected by protocol level integrity and confidentiality mechanisms, thus eavesdroppers cannot access the Data Object.

Moreover, in case the community of EURECOM employees is private, the difference is that, in the request for an attribute certificate, the attribute is encrypted. The node would therefore request a certificate corresponding to the attribute $\mathcal{E}_{k_I}(Workplace, EURECOM)$, where k_I is a key known by the Issuer only. The neighbor would therefore look-up in his *CertificateStore* whether he owns a certificate corresponding to $\mathcal{E}_{k_I}(Workplace, EURECOM)$ or not. From a privacy perspective, the difference is therefore that, in the public community case, the neighbor knows which attribute is requested, whereas in the private community case the neighbor learns which community is requested only if he belongs to the community himself.

In this simple scenario, nodes forward the message only to nodes belonging to the same community, whereas other nodes are excluded from the forwarding process. This scenario hence fits under the privacy model 2 decribed in section 3.4.

4.3 Conclusion

In this chapter we presented the architecture of the Haggle node. The Haggle node achieves opportunistic and autonomous communication thanks to a collapsed event-driven architecture. We described our security design and in particular we provided simple solutions to take into account the central notion of communities in Haggle.

We also implemented this design concretely on the Haggle platform by implementing a Security Manager and the required security events and filters. We also provided a HaggleCertificate library that enables the management of attribute certificates (from their generation to their storage). The core cryptographic functions are based on the OpenSSL cryptographic library, which we ported and compiled on a Windows Mobile 6 environment. The code of the Security Manager and of the whole Haggle node architecture can be retrieved at http://code.google.com/p/haggle/.

Finally we presented a simple secure opportunistic forwarding scenario. This scenario achieves privacy in the model 2 and is mainly used as a proof-of-concept of our security design. Achieving higher levels of privacy cannot however be provided by using existing solutions, which explains the difficulties raised by privacy-preserving forwarding in opportunistic communications.

In the sequel of this thesis, we present our research to provide privacy-preserving opportunistic forwarding protocols in the more challenging privacy models 3 and 4.

Part II

Secure Context-Based Forwarding

Chapter 5

Security Issues in Context-Based Forwarding

5.1 Introduction

As presented in section 2.2.3, context-based forwarding (e.g.[BCJP07, NGP07]) is an original communication paradigm, where messages are forwarded from source to destinations based on the context (e.g. location, workplace or social information) instead of explicit addresses. To be more precise, each message is associated with a message context which corresponds to the profile of its destination, and nodes make their forwarding decisions by comparing the context of the message with their own profile and the profile of their neighbors.

The assumption behind context-based forwarding is that the larger the context shared by two nodes, the higher the chances for these two nodes to meet one another (e.g. two persons working at the same company are highly likely to be in transmission range at some point). Thus, routing decisions in context-based forwarding are guided by the similarity between the contexts of encountered nodes and the destination's context.

Context-based forwarding is particularly adapted to challenged heterogeneous environments, like mobile opportunistic networks, where end-to-end connectivity is not guaranteed. Indeed, in such environments, classical routing mechanisms may be impractical, hence the transmission of messages should rely on opportunistic strategies.

However, context-based forwarding protocols present major security issues. Firstly, a classical security issue is data confidentiality: the data should only be accessible by the legitimate destination and therefore requires end-to-end encryption, as defined in section 3.4. This issue calls for innovative solutions in the context of opportunistic networks since such networks are delay-tolerant by nature and thus do not support end-to-end key agreement. Moreover, since the destination is defined implicitly through context information, even the identity of the destination itself is not necessarily known by the source: encryption can therefore not rely on a unique identifier of the destination and an associated key but should rather be flexible and encompass the implicit definition of the destination.

Second, user privacy is a crucial issue in such a protocol: the message context is essentially a subset of the profile of the destination, and the message is forwarded through various intermediate nodes that may not be trusted by the destination or the source. Moreover, trust relationships are loose in such a heterogeneous environment, therefore nodes want to keep a tight control over the access to their profile due to privacy reasons. Hence, intermediate nodes should be able to correctly make context-based forwarding decisions on encrypted messages. The encryption mechanism itself should not require prior contact with the destination (because of the lack of end-to-end connectivity) and should be public.

Furthermore, another consequence of the low level of trust is the requirement for a mechanism that provides assurance in the amount of shared context between encountered nodes and the destination. This amount is indeed a key element in the forwarding process and it is computed by neighboring nodes, therefore it is important to guarantee that the value claimed by the neighbors is not an incorrect value (e.g. to subvert the traffic). This issue is even more challenging with encrypted data, as providing a proof of computation correctness should not come at the expense of privacy exposure.

Our main contribution in this part are as follows:

- We study the problem of payload confidentiality and user privacy in context-based forwarding, and define the security primitives required to achieve privacy in the proposed protocol.
- We identify the issue of computation assurance, and the requirements to provide trust in the matching ratio while preserving privacy.
- We propose original solutions that can be efficiently combined in a secure context-based forwarding mechanism that features strong confidentiality and privacy enforcement along with proof of computation correctness.

This chapter introduces the security issues pertaining to context-based forwarding. In the next section, we describe the context-based forwarding model and then focus on the security requirements of the introduced model in section 5.3. We summarize these requirements and present an overview of our approach to securing context-based communication in section 5.4.

5.2 Context-Based Forwarding Settings

As explained in chapter 2, classical routing mechanisms are not well adapted to opportunistic and autonomic networks. This is in particular due to high node mobility which implies unstable network topology, and to the lack of end-to-end connectivity which results in unpredictable end-to-end delays. In such environments, the transmission of messages relies on dynamic forwarding strategies following the store, carry and forward principle described in section 2.2. In this chapter we investigate more particularly confidentiality and privacy issues in context-based forwarding strategies presented in section 2.2.3.

As presented in section 2.3, there is a conceptual gap between oblivious forwarding strategies and context-based ones in that the destination is not necessarily explicitly defined in the latter. The destination is rather implicitly defined as the collection of all nodes verifying a set of conditions or to put it in a simpler way, all nodes which own a given set of context attributes. This method of describing the destination is sometimes referred to as intentional naming, in particular in the DTN literature ([Fal03, KBM$^+$07, BKB08]), and is coupled with the late binding property. Late binding means that the mapping of an intentional name to the address described by the intentional name is not necessarily possible at the source (there is no online Domain Name System (DNS) service in DTNs): the mapping is performed progressively as the message is forwarded closer to the destination. In our context-based forwarding model we consider only very late binding in that only direct neighbors of the destination can completely map the intentional name with the address.

The assumption behind context-based forwarding is that the larger the context shared by two nodes, the higher the chances for these two nodes to meet one another (e.g. two persons working at the same company are highly likely to be in transmission range at some point). This assumption leads to an interesting forwarding strategy: the idea is that a node N_i forwards a message only to the neighbor with the highest number of attributes shared with the destination. If no neighbor shares more attributes with the destination than N_i itself, the node stores and carries the message until it meets a neighbor with a higher number of shared attributes. By adopting this strategy, the message is forwarded only to nodes with an ever increasing number of shared attributes until it eventually reaches the destination.

We follow in essence the model of HiBOp ([BCJP07]) or PROPICMAN ([NGP07]) without distinguishing between different types of context, and while considering only one hop neighbors. We define the settings more precisely in the following sections.

5.2.1 Network Settings

We consider a network composed of a set of n nodes $\{N_i\}_{1\leq i\leq n}$. The context of a node N_i is defined as a set of attributes $\{A_{i,j}\}_{1\leq j\leq m}$: $A_{i,j}$ is the j-th attribute of node N_i.

Definition 5.2.1 *An attribute is a couple (attribute name, attribute value). The j-th attribute of node N_i, denoted by $A_{i,j} = (E_j, V_{i,j})$ is composed of an attribute name E_j independent of N_i, and the value of the attribute $V_{i,j}$ which is relative to N_i.*

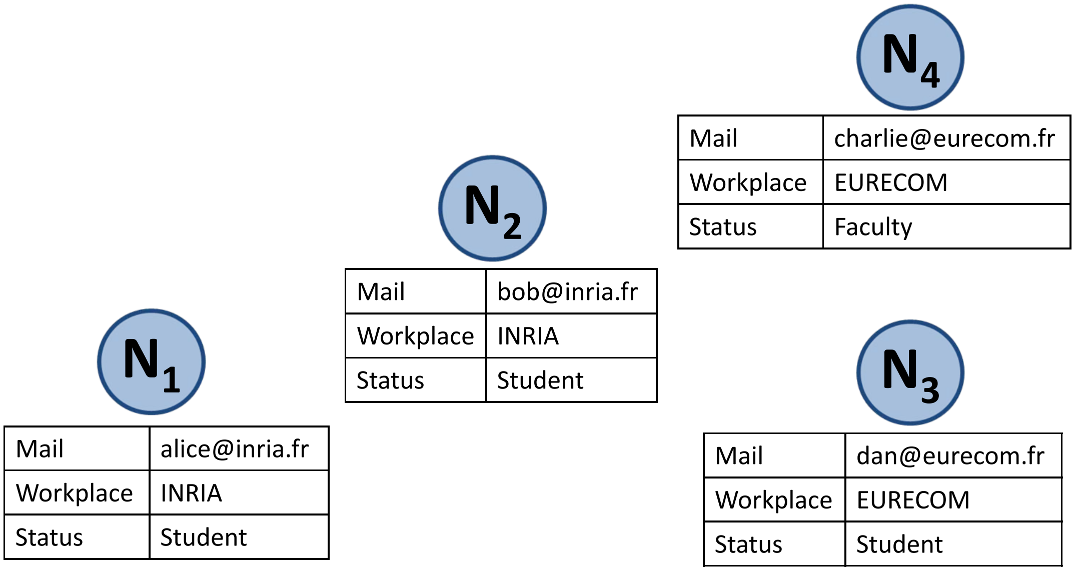
The attribute names are unique in that $E_{j_1} \neq E_{j_2}$ if $j_1 \neq j_2$.

All nodes have a set of m attributes, with the same attribute names E_j but the attribute values may vary between nodes.

Definition 5.2.2 *The profile $Prof(i)$ of node N_i is the concatenation of all its attributes: $Prof(i) = A_{i,1}||...||A_{i,m}$.*

The context of a node N_i is its profile $Prof(i)$.

This setting is illustrated by the small network given in figure 5.1. In this example there are $m = 3$ attribute names ($E_1 = Mail$,$E_2 = Workplace$ and $E_3 = Status$) and $n = 4$ nodes (N_1,N_2,N_3 and N_4). Each node's profile is an instantiation of the set of attributes (e.g. $Prof(1) = (Mail, alice@inria.fr)||(Workplace, INRIA)||(Status, student)$).



Mail	charlie@eurecom.fr
Workplace	EURECOM
Status	Faculty

Mail	bob@inria.fr
Workplace	INRIA
Status	Student

Mail	alice@inria.fr
Workplace	INRIA
Status	Student

Mail	dan@eurecom.fr
Workplace	EURECOM
Status	Student

Figure 5.1: Simple network composed of four nodes. Each node has a profile composed of three attributes represented below the node.

Finally we define the matching operation on attributes:

Definition 5.2.3 *We say that two attributes $A_{i_1,j_1} = (E_{j_1}, V_{i_1,j_1})$ and $A_{i_2,j_2} = (E_{j_2}, V_{i_2,j_2})$ match, and we write $A_{i_1,j_1} = A_{i_2,j_2}$, if and only if*

- *$E_{j_1} = E_{j_2}$, which implies $j_1 = j_2$,*
- *AND $V_{i_1,j_1} = V_{i_2,j_2}$.*

5.2.2 Message Format

A message M is a concatenation of two elements:

- the header $\mathcal{H}(M)$, which contains information about the destination of the message,
- the payload $\mathcal{PLD}(M)$, which contains the actual data sent to the destination.

Definition 5.2.4 *A message M is composed of a header $\mathcal{H}(M)$ and a payload $\mathcal{PLD}(M)$. The header is a concatenation of attributes:*

$$\mathcal{H}(M) = ||_{j \in L_M} A_{M,j},$$

where:

- $L_M \subset [1, m]$,
- $A_{M,j} = (E_j, V_{M,j})$ *is an attribute with the attribute name* E_j*, and a message dependent value* $V_{M,j}$*.*

The context of a message M corresponds to its header $\mathcal{H}(M)$. The header $\mathcal{H}(M)$ is used to perform networking operations as it defines the destination of the message implicitly: the destination of M is any node matching all the attributes in $\mathcal{H}(M)$ (with a straightforward extension of the definition 5.2.3).

Definition 5.2.5 *A node* N_i *is a destination of message* $M = \mathcal{H}(M)||\mathcal{PLD}(M) \iff \forall j \in L_M,\ A_{M,j} = A_{i,j}$*.*

5.2.3 Forwarding Model

When a node N_S ($1 \leq S \leq n$) wants to send a data, it constructs a message $M = \mathcal{H}(M)||\mathcal{PLD}(M)$ where:

- $\mathcal{PLD}(M)$ contains the data,
- $\mathcal{H}(M)$ is a set of conditions that have to be verified by the destination.

N_S then sends M to its neighbors.

When an intermediate node N_i receives the message M, it then needs to forward M closer to the destination. To this extent, N_i broadcasts $\mathcal{H}(M)$ to all its neighbors. Each of these neighbors N_k compares its own profile $Prof(k)$ with the header $\mathcal{H}(M)$ to evaluate the shared context between them.

Definition 5.2.6 *The shared context between a message* M *and a node* N_k*, corresponds to the attributes in* $\mathcal{H}(M)$ *that match attributes in* $Prof(k)$*.*

The matching set $L_{M,k} \subset L_M$ *corresponds to the indexes of attributes in the shared context between* M *and* N_k*, such that* $\forall j \in L_{M,k}, A_{M,j} = A_{k,j}$*.*

Definition 5.2.7 *The matching ratio* $p_k(M)$ *between a message* M *and a node* N_k *is defined as the number of attributes in the shared context between* M *and* N_k *divided by the number attributes in* $\mathcal{H}(M)$*.*

The matching ratio $p_k(M)$ *is thus computed as*

$$p_k(M) = \frac{|L_{M,k}|}{|L_M|},$$

where $|L_M|$ *and* $|L_{M,k}|$ *denote the cardinality of* L_M *and* $L_{M,k}$ *respectively.*

Definition 5.2.8 *We say that a message* M *and a node* N_k *present:*

- *no match, if* $p_k(M) = 0$,

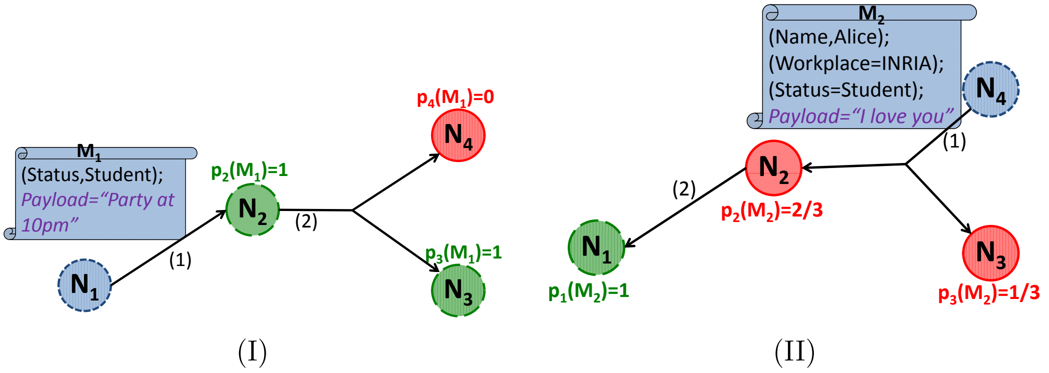


Figure 5.2: Two communication scenarios. In scenario (I), N_1 sends a message M_1 to all students (multiple destination) and in scenario (II) N_4 sends a message M_2 to N_1 (single destination). The nodes in dashed-blue are the source of messages (N_1 in scenario (I) and N_4 in scenario (II)), the nodes in long dashed-green the destinations (N_2 and N_3 in scenario (I) and N_1 in scenario (II)), and the nodes in regular-red are intermediate nodes that are not destinations (N_4 in scenario (I) and N_2 and N_3 in scenario (II)).

- *a partial match, if* $0 < p_k(M) < 1$,
- *a complete match, if* $p_k(M) = 1$.

In our forwarding model, the matching ratio $p_k(M)$ is interpreted as the probability that node N_k encounters a destination of M.

Hence, each node N_k replies to N_i by sending $p_k(M)$, and N_i selects the neighbor with highest matching ratio and forward the packet to this neighbor in unicast. If none of the neighbors has a matching ratio $p_k(M)$ higher than the matching ratio $p_i(M)$ of N_i itself, then N_i carries the packet and forwards it only when it encounters a neighbor with a higher matching ratio.

The matching ratio is also useful to determine whether a node is a destination of a message or not.

Property 5.2.9 *A node N_i is a destination of M if and only if M and N_i present a complete match, i.e.* $p_i(M) = 1$.

As previously mentioned, in this protocol, destinations of a message are implicitly defined, therefore a message can have multiple destinations. In the sequel of this part, destination thus refers to one node or to a set of nodes depending on the header of the message.

5.2.4 Examples

To illustrate the communication model we consider two scenarios depicted in figure 5.2.

5.2.4.1 Scenario (I)

In the first scenario, we assume that node N_1 wants to send a message to all students to advertise a party in the evening.

N_1 constructs the message M_1 with the payload:

$$\mathcal{PLD}(M_1) = \text{"}Party\ tonight\ at\ 10pm\text{"}$$

and a simple header composed of only one attribute:

$$\mathcal{H}(M_1) = (Status, student).$$

For this message, we have $L_{M_1} = \{3\}$.

N_1 broadcasts the header $\mathcal{H}(M_1)$ to its neighbor, N_2, which has a matching ratio of $p_2(M_1) = 1$. Therefore N_2 is a destination of M_1, and N_1 forwards the message M_1 to N_2 in unicast.

N_2 processes the packet in the same way: N_2 broadcasts the header $\mathcal{H}(M_1)$ to its neighbors:

- N_3 has a matching ratio of $p_3(M_1) = 1$
- N_4 has a matching ratio of $p_4(M_1) = 0$

Thus N_3 is a destination of M_1 whereas N_4 is not, and N_2 forwards the message M_1 to N_3 only in unicast.

5.2.4.2 Scenario (II)

As a second scenario, let us consider that N_4 wants to send a love declaration to N_1.

N_4 creates the message M_2 with payload:

$$\mathcal{PLD}(M_2) = \text{"}I\ love\ you\text{"}$$

and a header composed of three attributes

$$\mathcal{H}(M_2) = (Mail, alice@inria.fr)||(Workplace, INRIA)||(Status, student).$$

In this case $L_{M_2} = \{1, 2, 3\}$.

The header is broadcasted and received by N_2 and N_3.

- N_2 shares the workplace and status with the context of the message, thus $L_{M_2,2} = \{2, 3\}$ and $p_2(M_2) = 2/3$;
- N_3 shares only the status with the context of the message, thus $L_{M_2,3} = \{3\}$ and $p_3(M_2) = 1/3$.

Both N_2 and N_3 present partial matches with the message which indicates that none of them is a destination of M_2. However N_2 is more likely to encounter a destination of M_2 than N_3. Therefore N_4 sends in unicast the message M_2 to N_2.

N_2 then repeats the process and sends M_2 to N_1 which has a matching ratio $p_1(M_2) = 1$ and is therefore a destination of M_2.

In this scenario the destination is unique since $\mathcal{H}(M_2)$ contained a mail address which is a unique identifier. N_4 could therefore have constructed a message with one attribute (the mail address) only in the header, which would have resulted in the same set of destination for M_2. However, adding additional information about the destination increases the performance of the context-based forwarding strategy: using a unique attribute results in a situation where the destination has a complete match and other nodes no match, thus enabling only direct source-destination delivery, whereas multiple attributes enable other nodes to present partial matches and therefore forward the message closer to the destination through several hops.

We summarize the notations concerning the settings of context-based forwarding in Table 5.1.

The considered protocol takes forwarding decisions based on the message context, which is potentially sensitive information as it reveals a part of the profile of the destination. This protocol hence needs to be enhanced with security mechanisms from a confidentiality, privacy and reliability perspective. To this end, we first define the security requirements in the next section.

5.3 Security Requirements

As mentioned in section 3.2, the security requirements depend on the trust relationships among nodes. Trust is evaluated with respect to a specific operation. In the context-based protocol that we investigate in this part, trust is related to:

1. forwarding: will nodes forward the messages that they receive closer to the destination?
2. integrity: will nodes forward the messages without modifying it?
3. confidentiality: will nodes forward the messages without looking at their payload?
4. privacy: will nodes use the header only to take forwarding decisions?
5. assurance: will nodes correctly inform their neighbors on their matching ratio?

The issue of forwarding is very important in opportunistic networks as forwarding is performed by peers and not by an infrastructure. Therefore the trust level with respect to the forwarding task is lower compared with legacy networks relying on a routing infrastructure. This problem calls for cooperation enforcement solutions as presented in 3.1 and are out of the scope of this part. In this part, we therefore assume that most of the nodes are honest and take correct forwarding decisions based on the information they receive from

n	number of nodes
N_i	generic node number i
N_S	source node
N_D	destination node
m	number of attributes
$A_{i,j}$	j-th attribute of node N_i
E_j	Name of j-th attribute
$V_{i,j}$	Value of j-th attribute at node N_i
$Prof(i)$	Profile of node N_i
M	a message
$\mathcal{H}(M)$	header of message M
$\mathcal{PLD}(M)$	payload of message M
$A_{M,j}$	attribute of message M which name is E_j
$V_{M,j}$	value of attribute $A_{M,j}$
L_M	set of indexes of attributes of M
$\|X\|$	cardinal of set X
$L_{M,k}$	matching set between context of message M and node N_k
$p_k(M)$	matching ratio between context of message M and node N_k

Table 5.1: Settings of the context-based forwarding model

their neighbors. Few might be malicious and forward messages erroneously or not forward them at all, but adding redundancy prevents them from disrupting the communication.

The issue of integrity is important as well. Malicious nodes could indeed:

- modify the header to change the destination of a message,
- modify the payload to provide the destination with wrong data.

In addition, independently of nodes, the communication channel is prone to transmission errors. There is therefore a requirement for an integrity mechanism that allows a destination N_D to verify that:

- the header was set by the source N_S, and thus that N_D is a legitimate recipient of the message,
- the payload that N_D receives corresponds to the payload sent by N_S.

Classical solutions for end-to-end message integrity and authenticity can be transposed to the problem of header and payload integrity as explained in section 3.3. These solutions mainly consist in a cryptographic signature of the whole message by the source N_S. Unless the source requires anonymity, the problem of message integrity is not difficult to solve, and we do not investigate this issue further in this part.

Concerning the three remaining issues, confidentiality is required unless all nodes are trusted to be honest. This assumption is not reasonable in mobile opportunistic networks, where we consider that most of the nodes are honest but curious: we assume that most nodes are honest in performing the forwarding operation (thanks to cooperation enforcement scheme for example) but they will access the messages they receive if they can. We thus need to provide a mechanism for payload confidentiality such that only the destination of a message can access it.

The issue of privacy is similar: since we assume that nodes are honest but curious, they use the information in the header to correctly forward the messages but they might also use it to profile the destination. This raises the requirement of a mechanism to protect the confidentiality of the header in order to preserve user privacy.

Finally the issue of assurance, is similar to the general issue of trusting nodes with forwarding with a subtle difference. In the case of forwarding the question was whether a node takes a correct forwarding decision or not, and this can be enforced with cooperation enforcement solutions. Here the question is whether neighbors correctly inform the node about their matching ratio or not. This issue cannot be solved with cooperation enforcement scheme. We assumed that most nodes are honest concerning the forwarding operation but few might be malicious. We need to guarantee that these few malicious nodes cannot heavily affect the traffic, which raises the issue of providing a node with confidence in the matching ratio of its neighbors.

The three last problems are the main issues that we tackle in this part, and we therefore detail further the requirements to solve each of these issues in the following sections.

5.3.1 Payload Confidentiality

As for classical communication mechanisms, data confidentiality is one of the first security requirements that should be taken into account. Indeed, access to the content of any message should only be authorized to destined nodes. Data confidentiality is usually ensured by using cryptographic encryption algorithms. However, contrary to many existing solutions based on the establishment of secure tunnels between source and destination (e.g. SSL/TLS [Res00] or IPSec [KS05]), context-based communication protocols cannot rely on an end-to-end key management mechanism. Therefore source nodes should be able to encrypt the content of the message without a priori sharing any key with the destination node(s).

Moreover, since the identity of destination node(s) may be unknown by the source (such as in scenario (I) presented in section 5.2.4.1), the key used for the new encryption mechanism should be based on the set of attributes included in the context of the message. Only nodes which own the correct set of attributes should be able to access the content of the message. There is thus a very strong link between decryption keys and attributes used to encrypt the message. Only authorized nodes, i.e. those which own the corresponding attributes, should receive the required decryption keys. The decryption keys should therefore also depend on the set of attributes included in the context of the message.

The encryption mechanism should also be flexible in the sense that a node should be able to encrypt a message with any set of attributes. However, only nodes owning all attributes that were used to encrypt the message should be able to decrypt this message using the combination of keys corresponding to each of the attributes. For example, in the scenario (II) presented in section 5.2.4.2, the encryption key should be derived from the three attributes in the header ($(Mail, alice@inria.fr)$,$(Workplace, INRIA)$ and $(Status, student)$) and only N_1 should have the keys to decrypt the payload, but N_4 should not need to agree on an end-to-end key with N_1 beforehand.

Hence, in order to ensure payload confidentiality and only let authorized nodes access the payload, the two following security primitives have to be formally defined:

- ENCRYPT_PAYLOAD: used by the source to encrypt the payload of the message for the destination. This function should be public and the encryption key should be based on the attributes of the destination.
- DECRYPT_PAYLOAD: used by the destination node to decrypt the encrypted payload of the message. This function should be private: since messages are encrypted with respect to attributes, only nodes who actually have those attributes should receive the decryption keying material.

5.3.2 User Privacy

As opposed to classical forwarding or routing algorithms, context-based forwarding algorithms allow nodes to take forwarding decisions based on the context information instead of a specific address. However, since the context of a node reveals information on the user characteristics which is sensitive, such protocols raise new privacy concerns which conflict with the communication protocol.

As defined in section 3.4, depending on the trust level, several privacy models can be considered.

In the privacy model 1, all nodes are fully trusted with respect to each other's personal information. This means either nodes do not care about their privacy, or they trust other nodes in using the context to take forwarding decision and not to make malicious use of the context (for e.g. to profile the destination). This model, thus, encompasses completely honest nodes and is not realistic in opportunistic networks.

In the privacy model 2, some nodes are trusted and some are not. The trusted nodes are considered completely honest as in the previous model, and can therefore access the context. The network is therefore divided in two: the trusted nodes, which can access the context of other nodes and take forwarding decisions, and untrusted nodes which should not be able to access context information and cannot take context-based decision. Untrusted nodes can thus not take part in the context-based forwarding process, they can only take context-oblivious decisions as in epidemic forwarding. This model can be used by a controlled group of nodes (e.g. a military unit) in an adversarial environment. As presented in the second scenario of figure 3.4, the solution consists in encrypting the context such that only the group of trusted nodes can access it. These trusted nodes take the forwarding decision based on the context of the message and then encrypt the message before sending it. Untrusted nodes cannot access the context of the message but they can still serve as relays to forward the message in an epidemic way until the message reaches another member of the trusted group. This privacy model thus calls for classical mechanisms of group key management [BD94, CHL04, KLL04, KP05]. Furthermore, the applicability of this model is restricted to controlled groups in hostile environments and is not adapted to generic opportunistic networks, we thus do not discuss this privacy model further.

The two remaining privacy models require to keep the context of the message secret from forwarding nodes for privacy reasons. Yet, while being kept secret from any other node, the message still needs to reach the correct destination nodes. Therefore, forwarding nodes should be able to compare their profile to the context included in the header of the message without having access to unshared context information. Thus, the security requirements differ from those in payload confidentiality: while payload confidentiality is an end-to-end service where only nodes with a complete match (with matching ratio equal to one) can decrypt, the dedicated encryption function should still enable nodes with partial matches to discover shared attributes and hence to correctly forward packets.

It is impossible to preserve privacy in the model 4 (full privacy) defined in section 3.4 as intermediate nodes need to compare the context of the destination with their own and therefore learn at least the shared attributes to be able to perform context-based forwarding. Thus we propose an adaptable privacy (model 3) solution based on the Trusted Communities Assumption first presented in definition 3.2.1. In this protocol, we define communities based on attributes.

Definition 5.3.1 *Let $A_{M,j} = (E_j, V_{M,j})$ be an attribute. The community $\mathcal{COM}(A_{M,j})$ of attribute $A_{M,j}$ is composed by all nodes matching $A_{M,j}$, hence:*

$$\mathcal{COM}(A_{M,j}) = \{N_i | A_{i,j} = A_{M,j}\}.$$

In the example of figure 5.1, nodes N_1, N_2 and N_3 form the community of $(Status, student)$ (or in short the community of students as long as it is unambiguous). In the trusted communities assumption, nodes which belong to the same community trust each other and have common interests. Hence, revealing to another node a shared attribute is acceptable from a privacy perspective, but attributes that do not match should remain secret.

In the example of scenario (II), this means that node N_2 should be able to discover that the header $\mathcal{H}(M_2)$ of M_2 contains the attributes $(Workplace, INRIA)$ and $(Status, student)$. However, if N_2 manages to discover that $\mathcal{H}(M_2)$ contains the attribute $(Mail, alice@inria.fr)$ we consider that N_2 invades the privacy of N_1.

Finally, as for payload confidentiality, encryption of the header cannot rely on an end-to-end key management mechanism because of the opportunistic nature of the communication medium. Moreover, a node N_S, should be able to send a secret message to any other node N_D even if N_S does not share a single attribute with N_D. Therefore, the encryption function should be public and should not require a secret related to the attributes of the destination.

To sum up, in order to ensure user privacy in the model 3, context-based forwarding protocols require the definition of the following two security primitives:

- ENCRYPT_HEADER: used by the source to encrypt the context information. This function should be public and should enable forwarding nodes to compare their profile with the encrypted context in order to correctly forward packets.
- MATCH_HEADER: used by any forwarding node to determine whether the encrypted header includes some shared attributes of its profile. This function should not however reveal any additional information on non-matching attributes.

The primitive ENCRYPT_HEADER denotes a secure encoding of the header. It is different from traditional encryption schemes (e.g. the payload encryption) in that decryption is not required. However, we use the term "encrypt" in both cases for the sake of simplicity.

5.3.3 Computation Assurance

On top of privacy and confidentiality issues, the presented model raises an important security issue in the correctness of the matching ratio computation. Indeed, the context-based forwarding principle requires that the neighbor N_k of a node N_i computes the matching ratio $p_k(M)$ between the context of message M and its profile $Prof(k)$. The forwarding decision at node N_i solely depends on the correctness of the matching ratios of the neighbors, as N_i forwards the message to the neighbor with the highest matching ratio.

The security issue here is linked to a different scope of trust: can a node N_i trust its neighbor N_k in providing an accurate value of the matching ratio $p_k(M)$?

If node N_k is honest or honest-but-curious as most of the nodes, then the answer is yes, N_k complies with the protocol and does not provide fake matching ratio.

In the case where N_k is malicious though, N_k can stray from the protocol as it likes, and in particular provide false matching ratios. We previously mentioned that it was possible to deal with nodes dropping messages by adding some redundancy, thus if the number of nodes dropping all messages is low, communication would not be disrupted. However, if N_k provides erroneous high matching ratios on top of dropping messages, it can attract all messages forwarded by its neighbors and effectively disrupt communication in the neighborhood resulting in a Denial of Service attack. We call this attack a black-hole attack as it is close to the black hole attack studied in MANETs [YLY$^+$04, DLA02, Lun00, ASYP04]. The analogy in IP networks is a malicious router advertising distance 1 for all network addresses: this router attracts all communication in the network and drops the packets to bring the network down.

To prevent such attacks it is necessary to design a mechanism that guarantees that the matching ratio $p_k(M)$ advertised by N_k to N_i corresponds to the real ratio computed with the attributes in the header of the message $\mathcal{H}(M)$ and the profile $Prof(k)$.

The computation assurance requirement is a priori independent of privacy requirements, but privacy protection makes the problem more difficult for two reasons:

- In order to protect the privacy of N_k, N_i should not access the profile $Prof(k)$ of N_j and should not learn any information on the context of N_k, with the notable exception of shared attributes between N_k and N_i.
- The privacy protection of the destination node N_D implies that N_i does not have access to the whole header $\mathcal{H}(M)$ but only to the attributes shared between $Prof(i)$ and $\mathcal{H}(M)$. The same goes for N_k which should discover only attributes shared between $Prof(k)$ and $\mathcal{H}(M)$. The difficulty here comes from the fact that some of those attributes are unknown to N_i and therefore N_i cannot perform a direct verification on those attributes. For example, in scenario (II), N_2 sends the header $\mathcal{H}(M_2)$ to N_1, which shares attributes $Workplace$, $Status$ and $Name$ with the destination (in this particular example N_1 is the destination in fact). But N_2 shares only $Workplace$ and $Status$ with the destination but does not share $Name$ and therefore if a privacy-preserving mechanism is deployed, the attribute value corresponding to $Name$ is encrypted and unknown to N_2. So N_2 should have a method to verify that two encrypted data match without accessing these data.

To provide both privacy protection and computation assurance, the ENCRYPT_HEADER and MATCH_HEADER primitives should thus be enhanced and complemented to enable verification by intermediate nodes of the correctness of the matching ratio. The complementary security primitive is the following:

- VERIFY_RATIO: used by any forwarding node to determine whether the matching ratio advertised by a neighbor was correctly computed based on the encrypted header of the message. This function should not reveal any information on attributes which are not shared between nodes.

5.3.4 Trusting the Attributes

Attributes play a crucial role in the context-based communication model, and they are central elements in our security analysis: trusted communities are based on shared attributes. There is thus a strong link between each node's profile and the keying material that is required by security solutions.

The attributes should thus be verified and certified authentic by a trusted entity in order to provide an anchor of trust.

In the sequel of this part, we assume the existence of a Trusted Third Party (TTP) which is in charge of verifying that nodes' profile are correct and of distributing the related keying materials if so. However, the correct execution of the forwarding protocol should not rely on the presence of this TTP: because of the delay-tolerant nature of the network, the TTP only plays a role on the preliminary distribution of the keying material and is considered offline during the forwarding of the data.

5.4 Overview of the Proposed Security Solution

To summarize, we identified three security requirements which call for innovative security solutions in context-based forwarding: payload confidentiality, user privacy and computation assurance.

To cope with these requirements, a source node N_S uses ENCRYPT_PAYLOAD and ENCRYPT_HEADER to encrypt respectively the payload and the header of the message. Whenever an intermediate node N_i receives an encrypted message, it broadcasts the header to its neighbors. The neighbors use MATCH_HEADER in order to perform a privacy-preserving computation of the matching ratio and return the result to N_i. N_i uses VERIFY_RATIO to verify the correctness of the received ratio and then takes a forwarding decision. If the message reaches a node with a complete match, then this node is a destination of the message and it performs DECRYPT_PAYLOAD on the message to access the payload.

In the following chapters, we address each of these requirements separately.

In order to provide an anchor of trust while being compliant with the delay-tolerant nature of opportunistic networks, all our solutions include two phases:

- the setup phase, during which nodes contact the TTP to retrieve the keying material as well as the global parameters of the system,
- the runtime phase, during which the communication occurs opportunistically without access to the TTP.

The main characteristics of our solutions are as follows:

1. All nodes have the same set of m attribute names, but the value of an attribute may vary from one node to another. Nodes also get private keys corresponding to their attributes' values from an offline TTP.
2. Payload confidentiality:
    (a) The payload of each message is encrypted using an Identity-based Encryption (IBE) function.
    (b) A node can decrypt the message payload using an Identity-based Decryption (IBD) function, only if its profile presents a complete match with the message.
    (c) The main idea in implementing these end-to-end confidentiality functions is to use a sum of $|L_M|$ arguments instead of a single argument in IBD and IBE.
3. User Privacy:

(a) Each message's header contains $|L_M| \leq m$ attributes, where the values are encrypted with Public key Encryption with Keyword Search (PEKS) functions.

(b) A node can match its attributes to that in the header of a message, again using PEKS functions and its private keys.

(c) The forwarding of a message by an intermediate node is based on the matching ratio.

(d) The main idea to provide the matching capability to intermediate nodes is to modify the instantiation of PEKS, by introducing an offline TTP.

4. Computation Reliability:

(a) The privacy-preserving matching computation requires computation of some pseudorandom value.

(b) This pseudorandom value is hashed and is known only by nodes with matching attributes; exhibiting this value is thus a proof that the attributes match.

(c) The matching attributes are inserted in a counting Bloom filter to protect the privacy of the node.

(d) The counting Bloom filter computed by the node is compared with a counting Bloom filter computed by the source to determine the correct matching ratio.

We now present in detail the solutions for payload confidentiality in chapter 6, user privacy in chapter 7 and computation assurance in chapter 8, and we analyze the global solution in chapter 9.

Chapter 6

Solution for Payload Confidentiality

In this chapter we focus on providing a solution that meets the requirements of end-to-end confidentiality in context-based forwarding in Opportunistic Networks. As presented in section 5.3.1, such a solution relies on two main primitives:

- an encryption primitive that enables a node N_S to encrypt a payload to any destination without a shared key, and which should therefore be public,
- a decryption primitive that allows only the destination of a message (nodes which present a complete match) to access the payload and should therefore be private.

Nodes with shared attributes are able to match these attributes in the header (partial match) but only the destination can decrypt the payload (complete match).

Public encryption functions call for asymmetric cryptosystems. However, the implementation of classical schemes like RSA [RSA78], requires the existence of a public key infrastructure in order for a node to prove its identity: a node would fetch a destination's certificate before sending a message. Such solutions are unfortunately not practical in an opportunistic environment.

Identity-based cryptography is a good candidate for opportunistic environments since it avoids the use of certificates while being asymmetric. Therefore, we propose a solution based on refinements of identity-based cryptography to allow any node to compute an encrypted version of the message.

6.1 Related Work

Related work in this area is scarce because context-based forwarding is an emerging concept and existing security solutions do not fit the issues presented above. In [LKBG06], Lilien et al. present several challenges in privacy and security of opportunistic networks, and in particular the need for end-to-end confidentiality but they do not propose concrete solutions.

In the neighboring area of Delay Tolerant Networks (DTN), the DTNRG defined a Bundle Security Protocol (BSP [SFWL10]) to secure communications in DTN. BSP includes in particular a confidentiality block that only enables the encryption of the entire payload at the source and its decryption at the final destination. The solution is based on traditional public-key cryptography and the source requires the certificate of the destination prior to encryption. Furthermore BSP only support conversational communication between one source and one destination. The security features proposed by BSP are therefore not flexible enough to fit the requirements of context based forwarding.

More recently, Chuah et al. presented in [CRS10] a security solution adapted to descriptive messaging in DTN, which addresses the data confidentiality issue. Their solution is based on Ciphertext Policy Attribute-Based Encryption (CP-ABE) which is also an extension of identity-based encryption. CP-ABE is yet an expensive mechanism as it encompasses all kinds of logical expressions, and in particular the logical operators AND and OR which is not required in context-based forwarding. For this reason our tailored confidentiality solution is more efficient than CP-ABE.

6.2 Cryptographical Background

In this section we recall the definition of two useful cryptographic tools: hash functions and bilinear maps, and further recall classical notions related to formal security proofs.

6.2.1 Cryptographic Hash Functions

A hash function is a function that maps a block of data of arbitrary length to an element in a finite group. A hash function is therefore a deterministic encoding function whose output is called the hash value or digest.

A cryptographic hash function $hash : X \rightarrow Y$ ([Sti95]) is a hash function with the following three additional requirements:

- **First preimage resistance:** given a digest $h \in Y$ it should be hard to find any message $M \in X$ such that $h = hash(M)$.
- **Second preimage resistance:** given an input $M_1 \in X$, it should be hard to find another input, $M_2 \in X$ different from M_1 such that $hash(M_1) = hash(M_2)$.
- **Collision resistance:** it should be hard to find any two different messages $M_1 \in X$ and $M_2 \in X$ such that $hash(M_1) = hash(M_2)$.

Classical cryptographic hash functions include MD5 ([Riv92]) which is still in use in spite of the discovered vulnerabilities, SHA-1 and SHA-2 ([SHA08]). The National Institute of Standards and Technology (NIST) also launched a competition to select a new standard cryptographic hash function that will be denoted as SHA-3 ([SHA12]).

Cryptographic hash functions are usually modeled as random oracles [BR93]:

Definition 6.2.1 *A random oracle $RO : X \rightarrow Y$ is a map chosen by selecting each bit of $RO(x)$ uniformly and independently for every $x \in X$.*

Random oracles are useful tools to perform security proofs of cryptographic schemes using hash functions.

6.2.2 Bilinear Pairings

The cryptographic primitives used in our solution rely on cryptographic bilinear maps (also called bilinear pairings) [Sil86, BSS99], therefore we briefly define them in this section. We consider two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of same large prime order q. $\mathbb{G}_1$ is denoted additively and $\mathbb{G}_2$ multiplicatively. We denote by P a generator of $\mathbb{G}_1$.

Definition 6.2.2 *A cryptographic bilinear map is a function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ which satisfies the following properties:*

- *Bilinearity:*

$$\begin{aligned} \forall P_1, P_2, P_3 \in \mathbb{G}_1,\ & \hat{e}(P_1 + P_2, P_3) = \hat{e}(P_1, P_3)\hat{e}(P_2, P_3) \\ & \textit{and } \hat{e}(P_1, P_2 + P_3) = \hat{e}(P_1, P_2)\hat{e}(P_1, P_3), \end{aligned}$$

- *Non-degeneracy: $\hat{e}(P, P) \neq 1$, which means that $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$,*
- *Computable: There exists an efficient algorithm to compute $\hat{e}(P_1, P_2)$ for all $P_1, P_2 \in \mathbb{G}_1$.*

By corollary, bilinearity also implies the following useful property :

Proposition 6.2.3 *Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a cryptographic bilinear map. We have:*

$$\forall P_1, P_2 \in \mathbb{G}_1, k \in \mathbb{Z}_q^*, \hat{e}(kP_1, P_2) = \hat{e}(P_1, P_2)^k = \hat{e}(P_1, kP_2).$$

Examples of bilinear pairings include the Weil ([MOV93]) and Tate pairings ([FR94]).

In the sequel of this chapter, we assume the existence of a group and bilinear map generator $\mathcal{G}$ which proceeds as follows:

1. $\mathcal{G}$ takes a security parameter $sp \in \mathbb{Z}^+$ and outputs a prime number q,
2. $\mathcal{G}$ generates two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q,
3. $\mathcal{G}$ outputs the description of a cryptographic bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

We assume that $\mathcal{G}$ runs in polynomial time in *sp*. $\mathcal{G}$ should be randomized, and the number of bits of randomness depends on *sp*.

Definition 6.2.4 *A randomized algorithm or function produces an output depending on an internal random number on top of the input: the output of the function changes for successive runs on the same input.*

An example of such a group and bilinear map generator is given in [BF01].

6.2.3 Bilinear Diffie-Hellman Assumption

The BDH problem is a variant of the well-known Computational Diffie-Hellman (CDH) problem of a group $\mathbb{G}$, which consists in computing abP given $P, aP, bP \in \mathbb{G}$. We define the BDH problem formally.

Definition 6.2.5 *Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of prime order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a cryptographic bilinear map, and let P be a generator of $\mathbb{G}_1$. The BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is the following:*

Given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, compute $\hat{e}(P,P)^{abc}$.

We consider a randomized BDH parameter generator $\mathcal{G}$. On input of a security parameter sp, $\mathcal{G}$ outputs a tuple: $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$.

An algorithm $\mathcal{A}$ is said to have advantage $\epsilon(sp)$ in solving the BDH problem for $\mathcal{G}$ if for sufficiently large sp:

$$\mathcal{P}[\mathcal{A}(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P,P)^{abc}] \geq \epsilon(sp).$$

The probability is computed over the random generation of $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ by $\mathcal{G}$, the random choices of $a, b, c \in \mathbb{Z}_q^*$ and random choice of $P \in \mathbb{G}_1^*$.

$\mathcal{G}$ is said to satisfy the BDH assumption if $\epsilon(sp)$ is a negligible function in the sense that:

$$\forall \kappa \in \mathbb{Z}^+, \exists A \in \mathbb{Z}^+ | \forall sp > A, \epsilon(sp) < sp^{\kappa}.$$

When $\mathcal{G}$ satisfies the BDH assumption, the BDH problem is said to be hard in groups generated by $\mathcal{G}$.

It has been proved that the BDH problem is no harder than the CDH problem in $\langle q, \mathbb{G}_1 \rangle$ or $\langle q, \mathbb{G}_2 \rangle$. Therefore, assuming that the CDH problem is hard in $\langle q, \mathbb{G}_1 \rangle$ and $\langle q, \mathbb{G}_2 \rangle$ implies the hardness of the BDH problem in $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$.

6.2.4 `IND-CPA` Security

Security of public key encryption schemes is analyzed formally by defining security models. One common model is the semantic security against a chosen plaintext attack first as defined in [GM84, BDPR98].

For a classical public key system, semantic security is defined using the following game:

1. the adversary is given a random public key generated by the challenger

2. the adversary outputs two equal lengths messages M_0 and M_1 and receives the encryption of M_b from the challenger, where b is chosen at random in $\{0, 1\}$,

3. the adversary outputs $b' \in \{0, 1\}$ and wins the game if $b = b'$.

A public key system is said to be semantically secure if no polynomial time probabilistic adversary can win the game with a non-negligible advantage.

Intuitively, semantic security means that an adversary which is given a ciphertext learns nothing about the corresponding plaintext.

We now recall the definition and construction of the identity-based encryption scheme proposed by Boneh et al. in [BF01].

6.3 Identity-Based Encryption

Identity-based cryptography is a type of asymmetric key cryptography in which the public key of a node is the node's identity. Identity-based encryption allows a node to encrypt and send a message without previously receiving the destination node's public key. The first practical identity-based encryption scheme was proposed by Boneh et al. in [BF01]. In this section we remind the definition and the construction of an identity-based encryption system as proposed in [BF01].

6.3.1 Definition

An identity-based encryption scheme $\mathcal{IBE}$ is specified by the following four randomized algorithms:

1. `IB-Setup:` takes a security parameter *sp* and returns *params* (system parameters) and $master - key$. The system parameters include a description of a finite message space and a description of a finite ciphertext space.

2. `IB-Extract:` takes as input *params*, $master - key$, and an arbitrary $ID \in \{0,1\}^*$, and returns a private key d_{ID}.

3. `IB-Encrypt:` takes as input params, ID and $M \in \mathcal{M}$. It returns a ciphertext $M' \in \mathcal{C}$.

4. `IB-Decrypt:` takes as input *params*, $M' \in \mathcal{C}$ and a private key d_{ID}. It returns $M \in \mathcal{M}$.

These algorithms must be consistent:

$$\forall ID \in \{0,1\}^*, \forall M \in \mathcal{M}, \texttt{IB-Decrypt}(params, M', d_{ID}) = M,$$

where:

$$M' = \texttt{IB-Encrypt}(params, ID, M),$$

and

$$d_{ID} = \texttt{IB-Extract}(params, master - key, ID).$$

6.3.2 Construction

Following the definition of section 6.3.1, Boneh et al. proposed in [BF01] a construction of a basic identity-based encryption scheme based on bilinear maps. We recall their construction by describing the construction they proposed for each of the four algorithms:

1. `IB-Setup:` Given a security parameter $sp \in \mathbb{Z}^+$ the algorithm works as follows:
   (a) Run $\mathcal{G}$ on input sp, generate a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a cryptographic bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$.
   (b) Pick a random $s \in \mathbb{Z}_q^*$, and set $P_{pub} = sP \in \mathbb{G}_1$.
   (c) Choose two cryptographic hash functions:
      - $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$,
      - $H_2 : \mathbb{G}_2 \to \{0,1\}^\nu$ for some $\nu \in \mathbb{Z}^+$.

   (d) The message space is $\{0,1\}^\nu$, and the ciphertext space is $\mathbb{G}_1^* \times \{0,1\}^\nu$. The system parameters are then defined as:

   $$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, H_1, H_2 \rangle ,$$

   and the $master - key$ is $s \in \mathbb{Z}_q^*$.

2. `IB-Extract:` For a given string $ID \in \{0,1\}^*$ the algorithm proceeds as follows:
   (a) Compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$,
   (b) Set $d_{ID} = sQ_{ID}$.

3. `IB-Encrypt:` To encrypt $M \in \{0,1\}^\nu$ under the public key ID:
   (a) Compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$,
   (b) Choose a random $r \in \mathbb{Z}_q^*$,
   (c) Set $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in \mathbb{G}_2$
   (d) The ciphertext M' is the tuple:

   $$M' = \langle rP, M \oplus H_2(g_{ID}^r) \rangle .$$

4. `IB-Decrypt:` Let $M' = \langle U, V \rangle$ be a ciphertext encrypted under ID. If $U \notin \mathbb{G}_1^*$ reject the ciphertext. To decrypt M' using the private key $d_{ID} \in \mathbb{G}_1^*$ compute:

   $$V \oplus H_2(\hat{e}(d_{ID}, U)) = M.$$

This completes the description of the basic identity-based encryption scheme proposed in [BF01].

We note that `IB-Encrypt` is a randomized algorithm and therefore the ciphertext M' changes at each execution even if the inputs are the same.

In practice, `IB-Setup` is run by a trusted authority called the Public Key Generator (PKG). PKG is the only entity with knowledge of $master - key$, hence only PKG can run `IB-Extract` and produce private keys associated with identities.

Identity-based cryptography allows a node to encrypt a message with only the knowledge of the destination's identity. The destination needs to fetch the decryption key corresponding to its identity from the PKG. The destination is then able to decrypt the messages. The encryption mechanism is public and randomized, while the decryption mechanism is private (it requires the private key of the destination).

Identity based encryption is therefore a good candidate in order to achieve payload confidentiality in our protocol. However, since a source node may not know the destination identity in advance, and since the destination is defined as a set of attributes (in the header of the message), our protocol requires an extended version of identity based encryption that takes a set of attributes as input. We therefore propose an extension of identity-based encryption that we call multiple identity-based encryption in order to provide encryption based on the conjunction of attributes instead of a single identity.

6.4 Multiple Identity-Based Encryption

In this section we describe our proposal of multiple identity based encryption.

6.4.1 Definition

A multiple identity-based encryption scheme $\mathcal{MIBE}$ is specified by the following four randomized algorithms:

1. `MIB-Setup`: takes a security parameter *sp* and returns *params* (system parameters) and $master - key$. The system parameters include a description of a finite message space and a description of a finite ciphertext space.

2. `MIB-Extract`: takes as input $params$, $master - key$, and an arbitrary $ID \in \{0,1\}^*$, and returns a private key d_{ID}.

3. `MIB-Encrypt`: takes as input $params$, *a set of identities* $\{ID_i\}_{1 \leq i \leq \mu}$ (where $\mu \in \mathbb{Z}^+$) and $M \in \mathcal{M}$. It returns a ciphertext $M' \in \mathcal{C}$.

4. `MIB-Decrypt`: takes as input $params$, $M' \in \mathcal{C}$ and *a set of private keys* $\{d_{ID_i}\}_{1 \leq i \leq \mu}$. It returns $M \in \mathcal{M}$.

These algorithms must be consistent:

$$\forall \mu \in \mathbb{Z}^+, \forall \{ID_i\}_{1 \leq i \leq \mu} \in (\{0,1\}^*)^\mu, \forall M \in \mathcal{M}, \texttt{MIB-Decrypt}(params, M', \{d_{ID_i}\}_{1 \leq i \leq \mu}) = M,$$

where:

$$M' = \texttt{MIB-Encrypt}(params, \{ID_i\}_{1 \leq i \leq \mu}, M),$$

and

$$d_{ID_i} = \texttt{MIB-Extract}(params, master - key, ID_i).$$

We now explain our construction of a multiple identity-based encryption scheme by describing the four algorithms that we use.

6.4.2 Construction

This scheme is essentially the same as the original identity-based scheme [BF01], except that we use a sum of identities and private keys instead of a single one. This is possible thanks to the bilinearity of the bilinear map $\hat{e}$.

1. `MIB-Setup`: Given a security parameter $sp \in \mathbb{Z}^+$ the algorithm works as follows:
   (a) Run $\mathcal{G}$ on input sp, generate a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a cryptographic bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$.
   (b) Pick a random $s \in \mathbb{Z}_q^*$, and set $P_{pub} = sP \in \mathbb{G}_1$.
   (c) Choose two cryptographic hash functions:
      - $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$,
      - $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^\nu$ for some $\nu \in \mathbb{Z}^+$.
   (d) The message space is $\{0,1\}^\nu$, and the ciphertext space is $\mathbb{G}_1^* \times \{0,1\}^\nu$. The system parameters are then defined as:
   $$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, H_1, H_2 \rangle ,$$
   and the $master - key$ is $s \in \mathbb{Z}_q^*$.
2. `MIB-Extract`: For a given string $ID \in \{0,1\}^*$ the algorithm proceeds as follows:
   (a) Compute $Q_{ID} = H_1(ID) \in \mathbb{G}_1^*$,
   (b) Set $d_{ID} = sQ_{ID}$.
3. `MIB-Encrypt`: To encrypt $M\ in \{0,1\}^\nu$ under *the set of* μ *public keys* $\{ID_i\}_{1 \leq i \leq \mu}$:
   (a) For each $1 \leq i \leq \mu$, compute $Q_{ID_i} = H_1(ID_i) \in \mathbb{G}_1^*$,
   (b) Choose a random $r \in \mathbb{Z}_q^*$,
   (c) Set $g_{\{ID_i\}_{1 \leq i \leq \mu}} = \hat{e}(\sum_{i=1}^{\mu} Q_{ID_i}, P_{pub}) \in \mathbb{G}_2$
   (d) The ciphertext M' is the tuple:
   $$M' = \left\langle rP, M \oplus H_2(g^r_{\{ID_i\}_{1 \leq i \leq \mu}}) \right\rangle .$$
4. `MIB-Decrypt`: Let $M' = \langle U, V \rangle$ be a ciphertext encrypted under *the set of* μ *public keys* $\{ID_i\}_{1 \leq i \leq \mu}$. If $U \notin \mathbb{G}_1^*$ reject the ciphertext. To decrypt M' using *all the* μ *private keys* $d_{ID_i} \in \mathbb{G}_1^*$ *(for* $1 \leq i \leq \mu$*)* compute:
   $$V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} d_{ID_i}, U)) = M$$

This completes the description of the proposed multiple identity-based encryption scheme.

We prove the consistency of this new scheme:

Theorem 6.4.1 *The proposed construction of a multiple identity-based encryption scheme is consistent, i.e.:*

$$\forall \mu \in \mathbb{Z}^+, \forall \{ID_i\}_{1\leq i\leq\mu} \in (\{0,1\}^*)^\mu, \forall M \in \mathcal{M}, \texttt{MIB-Decrypt}(params, M', \{d_{ID_i}\}_{1\leq i\leq\mu}) = M,$$

where:

$$M' = \texttt{MIB-Encrypt}(params, \{ID_i\}_{1\leq i\leq\mu}, M),$$

and

$$d_{ID_i} = \texttt{MIB-Extract}(params, master - key, ID_i).$$

Proof Let $\{ID_i\}_{1\leq i\leq\mu}$ be a set of μ identities, and $M \in \mathcal{M}$.

For $1 \leq i \leq \mu$, let $d_{ID_i} = \texttt{MIB-Extract}(params, master - key, ID_i)$.

Let $M' = \texttt{MIB-Encrypt}(params, \{ID_i\}_{1\leq i\leq\mu}, M) = \langle U, V\rangle$.

We only need to prove that $V \oplus H_2(e(\sum_{i=1}^{\mu} d_{ID_i}, U)) = M$ to deduce that $\texttt{MIB-Decrypt}(params, M', \{d_{ID_i}\}_{1\leq i\leq\mu}) = M$.

$$\begin{aligned}
V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} d_{ID_i}, U)) &= V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} sQ_{ID_i}, rP)) \\
&= V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} Q_{ID_i}, sP)^r) \\
&= V \oplus H_2(\hat{e}(\sum_{i=1}^{\mu} Q_{ID_i}, P_{pub})^r) \\
&= M \oplus H_2(g^r_{\{ID_i\}_{1\leq i\leq\mu}}) \oplus H_2(g^r_{\{ID_i\}_{1\leq i\leq\mu}}) \\
&= M
\end{aligned}$$

∎

We now introduce a new security model that is adapted to multiple identity-based encryption, in order to prove the security of our payload confidentiality solution.

6.4.3 The IND-MID-CPA Security Model

In [BF01], Boneh et al. defined an improved (with respect to IND-CPA) semantic security model for identity based schemes denoted by IND-ID-CPA, where the adversary is allowed to issue private key extraction queries. The adversary is also challenged on the ID of its choice, instead of a random public key.

Similarly we define an improved model IND-MID-CPA to capture semantic security for our $\mathcal{MIBE}$ scheme. Our model captures the fact that the adversary is challenged on the set of identities of her choice instead of a single one.

We say that a multiple identity-based encryption scheme is semantically secure against an adaptative chosen plaintext attack (IND-MID-CPA) if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage against the Challenger in the following IND-MID-CPA game:

1. **Setup:** The challenger takes a security parameter sp, and runs the MIB-Setup algorithm. It gives the adversary the resulting system parameters $params$. It keeps $master - key$ to itself.

2. **Phase 1:** The adversary issues private key extraction queries $q_1, ..., q_{a_1}$ for identities ID_1,...ID_{a_1}. To answer q_i (with$1 \leq i \leq a_1$) the challenger computes:

   $$d_{ID_i} = \texttt{MIB-Extract}(params, master - key, ID_i).$$

   The challenger sends d_{ID_i} to the adversary.

   These queries may be asked adaptively, which means that each query q_i may depend on the replies to q_1,...q_{i-1}.

3. **Challenge:** Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts M_0,$M_1 \in \mathcal{M}$ and a set of identities $\{ID_j\}_{1 \leq j \leq \mu}$ for some $\mu \in \mathbb{Z}^+$ on which it wishes to be challenged. The only constraint is that at least one of the identities ID_j did not appear in any private key extraction query in Phase 1. The set of indexes of identities that were not previously queried is denoted by CU.

   The challenger picks a random bit $b \in \{0, 1\}$ and sets

   $$M' = \texttt{MIB-Encrypt}(params, \{ID_j\}_{1 \leq j \leq \mu}, M_b).$$

   It sends M' as the challenge to the adversary.

4. **Phase 2:** The adversary issues more queries q_{a_1+1},...,q_{a_2} for identities ID_{a_1+1},...ID_{a_2}.

   The only constraint is that these new queries do not correspond to any identity of the set $\{ID_i\}_{i \in CU}$.

   The Challenger responds as in Phase 1. These queries may be asked adaptively as in Phase 1.

5. **Guess:** The adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We refer to such an adversary $\mathcal{A}$ an IND-MID-CPA adversary. The adversary's advantage in attacking the multiple identity-based scheme $\mathcal{MIBE}$ is a function of the security parameter sp:

$$Adv_{\mathcal{MIBE},\mathcal{A}}(sp) = |\mathcal{P}[b = b'] - \frac{1}{2}|.$$

Definition 6.4.2 *The multiple identity-based scheme $\mathcal{MIBE}$ is said to be semantically secure against an adaptative chosen plaintext attack or simply* `IND-MID-CPA` *secure if for any polynomial time* `IND-MID-CPA` *adversary $\mathcal{A}$, the function $Adv_{\mathcal{MIBE},\mathcal{A}}(sp)$ is negligible.*

We now explain how to apply the multiple identity-based encryption scheme $\mathcal{MIBE}$ to meet the requirements of payload confidentiality in context-based forwarding.

6.5 Application of the $\mathcal{MIBE}$ Scheme to Payload Confidentiality

As described in section 5.3.1, payload confidentiality requires an end-to-end encryption mechanism between source and destination. The destination of a message is defined as a node which profile contains all the attributes included in the header of the message.

We therefore propose to use multiple identity-based encryption by using the attributes of the destination as the set of identities used for encryption.

As explained in section 5.4, we propose a solution featuring two phases: a setup phase featuring easy access to an authority, and a runtime phase featuring the actual opportunistic communication.

6.5.1 Setup Phase

During the setup phase, we assume the availability of a Public Key Generator (PKG). However, because of the delay-tolerant nature of the network, the PKG should only have the role of distributing keying materials. It thus needs to be considered as offline during the real execution of the communication protocol. The PKG is assumed to be a trusted authority.

The PKG first runs the algorithm `MIB-Setup`: it obtains $params$ which are public parameters of $\mathcal{MIBE}$ and a $master-key$, s, as described in section 6.4.2. s is kept secret by PKG and is never revealed to any other party.

During the setup phase, the nodes contact the PKG to fetch their private keys. We assume that each node N_i communicates with the PKG through a secure communication channel, and we do not detail the process of establishing such a secure channel. The communication process is the following:

1. The PKG sends $params$ to N_i.
2. N_i sends its profile $Prof(i) = ||_{1\leq j\leq m} A_{i,j}$ to the PKG.
3. The PKG:
   (a) verifies the validity of $Prof(i)$,
   (b) computes $A_{priv_{i,j}} = \texttt{MIB-Extract}(params, master-key, A_{i,j})$ for $1 \leq j \leq m$,
   (c) sends the set of private keys $\{A_{priv_{i,j}}\}_{1\leq j\leq m}$ to N_i.

Note that both the attribute name and attribute value are used in the private key extraction. The reason is that different attributes might have the same value (e.g the age of N_i as well as its office number could have the same value 29). By using both name and value, we guarantee that private keys corresponding to different attributes are different.

Each node N_i is required to enter setup phase before being able to communicate with its peers, but this does not imply that nodes need to synchronize to enter setup phase at the same time. Before joining the opportunistic network, node N_i needs to enter setup phase by contacting the PKG.

At the end of the setup phase, N_i has the public parameters *params* and m secrets $A_{priv_{i,j}}$ for $1 \leq j \leq m$. N_i can then communicate with all nodes that already performed the setup phase, and does not need to contact the PKG anymore.

6.5.2 Runtime Phase

We assume that all nodes involved in this phase have already performed the setup phase, and we describe the encryption and decryption processes.

The encryption of the payload $\mathcal{PLD}(M)$ of the message M with header $\mathcal{H}(M) = ||_{j\in L_M} A_{M,j}$ (see section 5.2) is the encryption with the $\mathcal{MIBE}$ scheme where the context of the message is used as the set of identities:

$$\begin{aligned}\mathcal{PLD}(M') &= ENCRYPT\_PAYLOAD(M)\\ &= \texttt{MIB-Encrypt}(params, \{A_{M,j}\}_{j\in L_M}, \mathcal{PLD}(M)).\end{aligned}$$

The destination profile includes, by definition 5.2.5, $A_{M,j}$ for $j \in L_M$, and therefore the destination has the corresponding private keys: $A_{priv_{M,j}}$. The destination can thus decrypt the encrypted payload in message M' by running the algorithm `MIB-Decrypt` with the set of private keys $\{A_{M,j}\}_{j\in L_M}$:

$$\begin{aligned}DECRYPT\_PAYLOAD(M') &= \texttt{MIB-Decrypt}(params, \mathcal{PLD}(M'), \{A_{priv_{M,j}}\}_{j\in L_M})\\ &= \mathcal{PLD}(M).\end{aligned}$$

The main characteristic of this scheme is that a node N_S can encrypt the payload of a message with any set of attributes and thus to any destination, even if N_S does not share these attributes. Conversely, only nodes which own all the attributes that were used to encrypt the payload can perform the decryption operation, and these nodes are by definition the destination of the message. Other nodes cannot decrypt the message even if they share some of the attributes. This solution therefore provides end-to-end confidentiality of the payload between source and destination without the need to establish an end-to-end key or a group key beforehand.

The proof of consistency of the scheme was given in section 6.4.2, and we focus now on the security analysis.

6.6 Security Evaluation

In this section we formally analyze the security of our proposed $\mathcal{MIBE}$ scheme.

In [BF01], Boneh et al. prove that the basic $\mathcal{IBE}$ scheme is semantically secure against a chosen plaintext attack (IND-ID-CPA) in the random oracle model assuming that the BDH problem is hard in groups generated by $\mathcal{G}$.

We follow their strategy to prove that $\mathcal{MIBE}$ is IND-MID-CPA. We thus define a public encryption (not identity-based) scheme BasicPub that we use afterwards to show that an IND-MID-CPA adversary on the $\mathcal{MIBE}$ scheme can be converted to an IND-CPA attacker on BasicPub.

6.6.1 BasicPub

We use the public key encryption scheme BasicPub defined in [BF01].

BasicPub is described by three algorithms: BP-keygen, BP-encrypt, BP-decrypt.

- BP-keygen: Given a security parameter $sp \in \mathbb{Z}^+$, the algorithm works as follows:
  1. Run $\mathcal{G}$ on input sp to generate two prime order groups $\mathbb{G}_1$, $\mathbb{G}_2$ and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let q be the order of $\mathbb{G}_1$, $\mathbb{G}_2$. Choose a random generator $P \in \mathbb{G}_1$.
  2. Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Pick a random $Q_{ID} \in \mathbb{G}_1^*$.
  3. Choose a cryptographic hash function $H_2 : \mathbb{G}_2 \to \{0,1\}^\nu$ for some ν.
  4. The public key is $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, Q_{ID}, H_2 \rangle$. The private key is $d_{ID} = sQ_{ID} \in \mathbb{G}_1^*$.
- BP-encrypt: To encrypt $M \in \{0,1\}^\nu$ choose a random $r \in \mathbb{Z}_q^*$ and set the ciphertext to be:
  $$M' = \langle rP, M \oplus H_2(g^r) \rangle \,,$$
  where $g = \hat{e}(Q_{ID}, P_{pub}) \in \mathbb{G}_2^*$.
- BP-decrypt: Let $M' = \langle U, V \rangle$ be a ciphertext created by using the public key. To decrypt M' using the private key $d_{ID} \in \mathbb{G}_1^*$ compute:
  $$V \oplus H_2(\hat{e}(d_{ID}, U)) = M.$$

This completes the description BasicPub. We now show that an IND-MID-CPA attack on the $\mathcal{MIBE}$ scheme can be converted to a IND-CPA attack on BasicPub.

6.6.2 Reduction of IND-MID-CPA to IND-CPA Security

Theorem 6.6.1 *Let H_1 be a random oracle from $\{0,1\}^*$ to $\mathbb{G}_1^*$. Let $\mathcal{A}$ be an IND-MID-CPA adversary that has advantage $\epsilon(sp)$ against the $\mathcal{MIBE}$ scheme. Suppose $\mathcal{A}$ makes at most $q_E > 0$ private key extraction queries. Then there exists an IND-CPA adversary $\mathcal{B}$ that has advantage at least $\frac{\epsilon(sp)}{e(1+q_E)}$ against BasicPub. Its running time is $O(time\mathcal{A})$.*

Our proof is similar to the proof presented in [BF01] with the notable difference that we need to modify the behavior of the random oracle H_1 to take into account the multiple identities used during the challenge.

Proof We show how to construct an IND-CPA adversary $\mathcal{B}$ that uses $\mathcal{A}$ to gain advantage $\epsilon/e(1+q_E)$ against BasicPub.

The game between the challenger and the adversary $\mathcal{B}$ starts with the challenger first generating a random public key by running algorithm BP-keygen of BasicPub. The result is a public key

$$K_{pub} = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, Q_{ID}, H_2 \rangle$$

and a private key $d_{ID} = sQ_{ID}$. The challenger gives K_{pub} to algorithm $\mathcal{B}$. Algorithm $\mathcal{B}$ is supposed to output two messages M_0 and M_1 and expects to receive back the BasicPub encryption of M_b under K_{pub} where $b \in \{0,1\}$. Then algorithm $\mathcal{B}$ outputs its guess $b' \in \{0,1\}$ for b.

Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in an IND-MID-CPA game as follows ($\mathcal{B}$ simulates the challenger for $\mathcal{A}$):

- **Setup:** Algorithm $\mathcal{B}$ gives $\mathcal{A}$ the system parameters:

  $$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, H_1, H_2 \rangle ,$$

  where H_1 is a random oracle controlled by $\mathcal{B}$ as described below and the other parameters were generated by the challenger.

- **H_1-queries:** At any time, algorithm $\mathcal{A}$ can query the random oracle H_1. To respond to these queries, algorithm $\mathcal{B}$ maintains a list of tuples $\langle ID_i, Q_i, b_i, c_i \rangle$ as explained below. We refer to this list as the H_1^{list}. The list is initially empty. When $\mathcal{A}$ queries the oracle H_1 at a point ID_i algorithm $\mathcal{B}$ responds as follows:

  1. If the query ID already appears on H_1^{list} as a tuple $\langle ID_i, Q_i, b_i, c_i \rangle$, then algorithm $\mathcal{B}$ responds with $Q_i \in \mathbb{G}_1^*$.
  2. Otherwise, $\mathcal{B}$ generates a random $c_i \in \{0,1\}$ so that $\mathcal{P}[c_i = 0] = \delta$ for some δ that will be determined later.
  3. Algorithm $\mathcal{B}$ picks a random $b_i \in \mathbb{Z}_q^*$. Then,
     - If $c_i = 0$ compute $Q_i = b_i P \in \mathbb{G}_1^*$.
     - If $c_i = 1$ compute $Q_i = b_i Q_{ID} \in \mathbb{G}_1$.
  4. Algorithm $\mathcal{B}$ adds the tuple the tuple $\langle ID_i, Q_i, b_i, c_i \rangle$ on the H_1^{list} and responds to $\mathcal{A}$ with Q_i.

- **Phase 1:** Let ID_i be a private key extraction query issued by algorithm $\mathcal{A}$. Algorithm $\mathcal{B}$ responds to this query as follows:

1. Run the above algorithm for responding to H_1-queries and obtain a $Q_i \in \mathbb{G}_1^*$ such that $H_1(ID_i) = Q_i$. Let $\langle ID_i, Q_i, b_i, c_i \rangle$ be the corresponding tuple on the H_1^{list}. If $c_i = 1$ then $\mathcal{B}$ reports failure and terminates. The attack on `BasicPub` failed.
2. We know $c_i = 0$ and hence $Q_i = b_i P$. We define $d_i = b_i P_{pub} \in \mathbb{G}_1^*$: $d_i = sQ_i$ is the private key associated to the public key ID_i. Give d_i to $\mathcal{A}$.

- **Challenge:** Once algorithm $\mathcal{A}$ decides that Phase 1 is over it outputs a set of public keys $\{ID_{ch_j}\}_{1 \le j \le \mu_{ch}}$ and two messages M_0, M_1 on which it wishes to be challenged. The set of public keys contains some (possibly zero) keys which where queried in Phase 1 and some (at least one) keys which where not queried. By reordering the set, with thus assume that:

  - the first μ_{kn} keys $\{ID_{ch_j}\}_{1 \le j \le \mu_{kn}}$ where queried in phase 1 (with $0 \le \mu_{kn} < \mu_{ch}$,
  - the last $\mu_{ch} - \mu_{kn}$ keys $\{ID_{ch_j}\}_{\mu_{kn}+1 \le j \le \mu_{ch}}$ keys where not queried during phase 1.

  Algorithm $\mathcal{B}$ responds as follows:

  1. Algorithm $\mathcal{B}$ gives its challenger the messages M_0 and M_1. The challenger responds with a `BasicPub` ciphertext $M_c' = \langle U, V \rangle$ which is the encryption of M_c for a random $c \in \{0, 1\}$.
  2. Next, $\mathcal{B}$ runs the algorithm for responding to H_1-queries for the public keys $\{ID_{ch_j}\}_{\mu_{kn}+1 \le j \le \mu_{ch}-1}$ and obtains tuples $\left\langle ID_{ch_j}, Q_{ch_j}, b_{ch_j}, c_{ch_j} \right\rangle$ for $\mu_{kn} + 1 \le j \le \mu_{ch} - 1$.
  3. For the last unknown public key $ID_{ch_{\mu_{ch}}}$, $\mathcal{B}$ proceeds to a slight modification in the H_1-query: $\mathcal{B}$ performs the first three steps of the H_1 query as usual and obtains a tuple $\left\langle ID_{ch_{\mu_{ch}}}, Q_{ch_{\mu_{ch}}}, b_{ch_{\mu_{ch}}}, c_{ch_{\mu_{ch}}} \right\rangle$. This tuple is **not** added to the H_1^{list}. $\mathcal{B}$ rather adds the following tuple:

  $$\left\langle ID_{ch_{\mu_{ch}}}, Q_{ch_{\mu_{ch}}} - \sum_{j=1}^{\mu_{ch}-1} Q_j, b_{ch_{\mu_{ch}}}, c_{ch_{\mu_{ch}}} \right\rangle$$

     to the H_1^{list}. Thus the H_1-query of $ID_{ch_{\mu_{ch}}}$ returns $Q_{ch_{\mu_{ch}}} - \sum_{j=1}^{\mu_{ch}-1} Q_j$. Note that this value remains uniform in $\mathbb{G}_1^*$.
  4. If $c_{ch_{\mu_{ch}}} = 0$ then $\mathcal{B}$ reports failure and terminates. The attack on `BasicPub` failed.
  5. We know that $c_{ch_{\mu_{ch}}} = 1$ and therefore $Q_{ch_{\mu_{ch}}} = bQ_{ID}$. Recall that when $M_c' = \langle U, V \rangle$, $U \in \mathbb{G}_1^*$. $\mathcal{B}$ then sets $M' = \left\langle b_{ch_{\mu_{ch}}}^{-1} U, V \right\rangle$, where $b_{ch_{\mu_{ch}}}^{-1}$ is the inverse of $b_{ch_{\mu_{ch}}}$ mod q. Finally Algorithm $\mathcal{B}$ responds to $\mathcal{A}$ with the challenge ciphertext M'.

Note that M' is a proper `MIB-encryption` of M_c under the set of public keys $\{ID_{ch_j}\}_{1\leq j\leq\mu_{ch}}$ as required. To see this, first observe that

$$\sum_{j=1}^{\mu_{ch}} H_1(ID_{ch_j}) = Q_{ch_{\mu_{ch}}}.$$

Thus the set of private keys $\{d_{ID_{ch_j}}\}_{1\leq j\leq\mu_{ch}}$ verifies:

$$\sum_{j=1}^{\mu_{ch}} d_{ID_{ch_j}} = \sum_{j=1}^{\mu_{ch}} sH_1(ID_{ch_j}) = s\sum_{j=1}^{\mu_{ch}} H_1(ID_{ch_j}) = sQ_{ch_{\mu_{ch}}}.$$

Second, observe that:

$$\hat{e}(b^{-1}U, \sum_{j=1}^{\mu_{ch}} d_{ID_{ch_j}}) = \hat{e}(b^{-1}U, sQ_{ch_{\mu_{ch}}}) = \hat{e}(U, sb^{-1}Q_{ch_{\mu_{ch}}}) = \hat{e}(U, sQ_{ID}) = \hat{e}(U, d_{ID}).$$

Hence, the $\mathcal{MIBE}$ decryption primitive `MIB-Decrypt` of M' using the private keys corresponding to $\{ID_{ch_j}\}_{1\leq j\leq\mu_{ch}}$ is the same as the `BasicPub` decryption of M'_c using d_{ID}.

- **Phase 2:** Algorithm $\mathcal{B}$ responds to private key extraction queries as in Phase 1.
- **Guess:** Eventually algorithm $\mathcal{A}$ outputs a guess c' for c. Algorithm $\mathcal{B}$ outputs c' as its guess for c.

The algorithm $\mathcal{A}$'s view is identical to its view in a real attack as each response to H_1-queries are uniformly and independently distributed in $\mathbb{G}_1^*$. Furthermore, all responses to private key extraction queries are valid.

By definition of algorithm $\mathcal{A}$, $|\mathcal{P}[c = c'] - \frac{1}{2}| \geq \epsilon(sp)$, therefore if $\mathcal{B}$ does not abort then $\mathcal{B}$ has the same advantage $|\mathcal{P}[c = c'] - \frac{1}{2}| \geq \epsilon(sp)$.

It remains to compute the probability $p_{ab}^{q_E}(\delta)$ that $\mathcal{B}$ aborts during the simulation. If $\mathcal{A}$ makes a total of q_E private key extraction queries, then the probability that $\mathcal{B}$ does not abort in phases 1 or 2 is δ^{q_E}, while during the challenge the probability is $1-\delta$; the total probability $p_{ab}^{q_E}(\delta)$ is hence $\delta^{q_E}(1-\delta)$. The derivative of the function $p_{ab}^{q_E}$ is:

$${p_{ab}^{q_E}}'(\delta) = q_E\delta^{q_E-1}(1-\delta) - \delta^{q_E} = \delta^{q_E-1}(q_E - (1+q_E)\delta).$$

$p_{ab}^{q_E}(\delta)$ is therefore maximized for δ_{opt} verifying:

$$q_E - (1+q_E)\delta_{opt} = 0,$$

$$\Rightarrow \delta_{opt} = \frac{q_E}{1+q_E} = 1 - \frac{1}{1+q_E}.$$

The probability $p_{ab}^{q_E}$ at δ_{opt} is thus:

$$\begin{aligned} p_{ab}^{q_E}(\delta_{opt}) &= (1-\frac{1}{1+q_E})^{q_E}\frac{1}{1+q_E} \\ &= e^{q_E ln(1-\frac{1}{1+q_E})}\frac{1}{1+q_E} \\ p_{ab}^{q_E}(\delta_{opt}) &\geq e^{-\frac{q_E}{1+q_E}}\frac{1}{1+q_E} \\ &\geq \frac{1}{(1+q_E)e} \end{aligned}$$

By choosing $\mathcal{P}[c=0]=\delta_{opt}$, $\mathcal{B}$ is hence assured to have advantage at least $\frac{\epsilon(sp)}{e(1+q_E)}$ in the `IND-MID-CPA` game. ∎

To end the security proof of the $\mathcal{MIBE}$ scheme we use a reduction of the IND-CPA attacker on `BasicPub` to the BDH problem.

6.6.3 Semantic Security of the $\mathcal{MIBE}$ Scheme

In [BF01], Boneh et al. also prove the following result:

Lemma 6.6.2 *Let H_2 be a random oracle from $\mathbb{G}_2$ to $\{0,1\}^\nu$. Let $\mathcal{A}$ be an `IND-CPA` adversary that has advantage $\epsilon(sp)$ against `BasicPub`. Suppose $\mathcal{A}$ makes a total of $q_{H_2}>0$ queries to H_2. Then there is an algorithm $\mathcal{B}$ that solves the BDH problem for $\mathcal{G}$ with advantage at least $\frac{2\epsilon(sp)}{q_{H_2}}$ and a running time $O(time(\mathcal{A}))$.*

Theorem 6.6.3 *Suppose the hash functions H_1, H_2 are random oracles. Then our $\mathcal{MIBE}$ scheme is a semantically secure identity based encryption scheme (`IND-MID-CPA`) assuming BDH is hard in groups generated by $\mathcal{G}$.*

Proof The theorem follows directly from Theorem 6.6.1 and Lemma 6.6.2. Composing both reductions shows that an `IND-MID-CPA` adversary on our $\mathcal{MIBE}$ scheme with advantage $\epsilon(sp)$ gives a BDH algorithm for $\mathcal{G}$ with advantage at least $\frac{2\epsilon(sp)}{e(1+q_E)q_{H_2}}$. ∎

6.7 Summary

The fact that $\mathcal{MIBE}$ scheme is `IND-MID-CPA` secure essentially means that an attacker is unable to decrypt a payload $\mathcal{PLD}(M)$ of a message M unless it owns all the private keys corresponding to the attributes in the header $\mathcal{H}(M)$. Even if an attacker has some of the required keys, it cannot deduce any information on $\mathcal{PLD}$ due to semantic security.

Hence, the payload confidentiality proposal provides end-to-end confidentiality of the payload without requiring the source and destination to share keys prior to communication. The mechanism is flexible as the key used for encryption is derived from the very definition of the destination through the attributes included in the header of the message. Any node can therefore encrypt the payload to any destination, and only the intended destination can decrypt the payload. The $\mathcal{MIBE}$ scheme thus meets the requirements of payload confidentiality in context-based forwarding.

This completes the presentation of the confidentiality solution and we focus, in the next chapter, on the mechanisms to enforce user privacy.

Chapter 7

Proposal for User Privacy

7.1 Introduction

As mentioned in section 5.3.2, privacy is an important requirement in context-based forwarding where forwarding decisions are taken based on the context. The context of a node N_i contains private information as it reflects the characteristics of the user who owns N_i. The context of the message M itself also is private as it reveals the attributes of the destination of M. Hence, preserving privacy and taking forwarding decisions based on context present conflicting requirements: while the former requires context protection, the latter needs access to the context.

In this chapter we present a solution (section 7.4) to protect user privacy in the model 3 (as defined in section 3.4) based on the Trusted Communities Assumption presented in definition 3.2.1: this solution enables intermediate nodes to compute the matching ratio required for taking context-based forwarding decisions by discovering matching attributes only. The intermediate nodes do not learn any information on non-matching attributes though. Our solution, which was first described in [SÖM09b], defines two security primitives as defined in section 5.3.2:

- ENCRYPT_HEADER: used by the source to encrypt the context information.
- MATCH_HEADER: used by any forwarding node to determine whether the encrypted header includes some shared attributes of its profile.

The proposed solution is based on a cryptographic mechanism called Public Key Encryption with Keyword Search (PEKS) that was introduced by Boneh et al. in [BCOP04]. We recall their definition and construction in section 7.3.

We start by analyzing a first approach proposed by Nguyen et al. in [NGP07] and draw lessons for the design of a satisfying solution.

7.2 Basic Approach Based on Hash Functions

7.2.1 Solution Sketch

The first idea to solve the privacy issue in context-based forwarding is to use cryptographic hash functions, as proposed in [NGP07]. We gave the definition and main properties of cryptographic hash functions in section 6.2.1.

The idea proposed in [NGP07] is to use a cryptographic hash function *hash* to implement the ENCRYPT_HEADER primitive.

To be more precise, a node N_S which wants to send a message M simply computes the hash of all the values of the header $\mathcal{H}(M)$, thus obtaining:

$$\mathcal{H}(M) = ||_{j\in L}(E_j, hash(V_{M,j})).$$

As a counterpart, an intermediate node N_i implements MATCH_HEADER as follows:

- N_i first hashes its profile with the same hash function *hash*,
- N_i then tests whether one of its hashed attributes $(E_j, hash(V_{i,j}))$ is equal to an attribute $(E_j, hash(V_{M,j}))$ of the received header.

The security argument is based on the preimage resistance properties of *hash*:

- First preimage resistance ensures that given en encoded attribute $(E_j, hash(V_{M,j}))$, it is hard to find the original value $V_{M,j}$. Therefore, a node is unable to discover the value of non-matching attributes.
- Second preimage resistance ensures that given an encoded attribute $(E_j, hash(V_{M,j}))$, it is hard to find a different value $V_{i,j} \neq V_{M,j}$, such that:

  $$(E_j, hash(V_{i,j})) = (E_j, hash(V_{M,j})).$$

  Hence, the attributes where the equality holds are shared attributes, and the ones where it does not hold are non-matching attributes.

This idea seems attractive because it requires only a public function, which is *hash*. Furthermore, hash functions are widely available, and they are efficient to compute. Moreover this solution does not involve encryption keys, hence neither key management nor any kind of centralized authority is required. The solution thus looks adapted to MobiOpps.

7.2.2 Security Concerns

Although the idea of using hash functions seems attractive because these functions are public and efficient in terms of computation, this solution unfortunately does not meet the security requirements defined in section 5.3: this solution is namely prone to dictionary attacks and such attacks have a strong impact on the privacy of users.

Definition 7.2.1 *A dictionary attack on a cryptosystem consists in successively trying all the words in an exhaustive list called a dictionary as input of the cryptosystem.*

Dictionary attack [Shi07] is thus a kind of brute force attack where the guesses of the adversary are focused on a small message space (the dictionary) consisting of possibilities which are most likely to succeed.

In context based communications, attributes are not pseudo-random sequences, they are rather well formated and have a meaning: the space of realistic values is therefore rather small. Therefore, an intermediate node can compute the hash of each word in a dictionary (containing all the possible values of attributes) and then simply looks up the values of the message header in the hashed dictionary to discover the values of all (matching and non-matching) attributes of the header. For example, the attribute *Status* in the example presented in Figure 5.1 can only take three values in a university: *student* or *faculty* or *staff*.

Note that the dictionary attack does not contradict the second preimage resistance property. The second preimage property assumes indeed that the size of input and output space is large enough to prevent brute force guessing. For example the output space of SHA-1 is 160 bits while the input space for one block in SHA-1 is 512 bits (but SHA-1 processes messages of up to 2^{64} bits). SHA-1 verifies the second preimage property for arbitrary inputs, yet if the input space is reduced due to external constraints then the property does not hold anymore. In the example of the attribute *Status*, the message space has size 3 and therefore SHA-1 restricted to this space does not have preimage resistance properties. In this particular example the message space is extremely small on purpose, but the attack is efficient on much larger spaces as is it possible to compute around 2^{20} SHA-1 hashes of 512 bits inputs per second with a normal desktop computer (which is greater than the number of words in an english dictionary).

Since the hash function is public, any node can launch a dictionary attack on an intercepted message. Dictionary attacks can thus easily and efficiently be launched and hash functions, as they are used here, do not provide privacy.

7.2.3 Preventing Dictionary Attacks

In order to avoid dictionary attacks against the mechanism protecting user privacy, two important properties are required:

- MATCH_HEADER, the counterpart of ENCRYPT_HEADER, should be private. This property means that a node should only be able to match the values of attributes in its profile and no other values. This was already mentioned in section 5.3.2, but the basic solution does not verify this property. This property also implies that ENCRYPT_HEADER and MATCH_HEADER should be different functions.

- The output of ENCRYPT_HEADER should be randomized in the sense of definition 6.2.4, which means that the output of ENCRYPT_HEADER should be different at each execution, even if the input does not change. Indeed, all nodes need to be able to compute the primitive ENCRYPT_HEADER on any input in order to be able to send a message to any destination, therefore ENCRYPT_HEADER has to be implemented by a public function as mentioned in section 5.3.2. If the output of ENCRYPT_HEADER is deterministic, nodes can launch a dictionary attack on all possible inputs as explained above, but this attack cannot be launched if the output of ENCRYPT_HEADER is randomized.

In summary, the problem of preserving privacy in context-based forwarding cannot be solved simply with hash functions. We propose an elaborate solution based on searchable encryption, with strong security properties. In the next section, we present a searchable encryption mechanism called Public key Encryption with Keyword Search (PEKS [BCOP04]) which is one of the main components of our privacy-preserving solution.

7.3 Public key Encryption with Keyword Search (PEKS)

Public key Encryption with Keyword Search (PEKS) -or in short searchable encryption- introduced by Boneh et al. in [BCOP04] is a cryptographic mechanism based on bilinear maps (see section 6.2.2). PEKS allows a node to search for some keyword in some encrypted data without being able to retrieve any additional information from the data. This test can be performed only if the node has previously received a trapdoor corresponding to the keyword. Thanks to PEKS, only authorized nodes can perform the keyword search test and they cannot learn any information apart from the occurrence or not of the keyword in the encrypted data.

7.3.1 Definition

In [BCOP04], Boneh et al. define a non-interactive public key encryption with keyword search scheme $\mathcal{PEKS}$ as a scheme consisting of polynomial time randomized algorithms as follows (we slightly modified the definition of [BCOP04] to make it more consistent with the definitions of the previous chapter):

1. $\texttt{SE-Setup}(sp)$: takes a security parameter sp, and outputs $params$ (system parameters).

2. $\texttt{SE-KeyGen}(params)$: takes as input $params$ and generates a public/private key pair pk_A, sk_A.

3. $\texttt{SE-PEKS}(params, pk_A, W)$: for a public key pk_A and a keyword W, this algorithm produces S_W a searchable encryption of W.

4. $\texttt{SE-Trapdoor}(params, sk_A, W)$: given a private key sk_A and a keyword W produces a trapdoor T_W.

5. $\texttt{SE-Test}(params, S_{W_1}, T_{W_2})$: this algorithm takes:

- a searchable encryption $S_{W_1} = \texttt{SE-PEKS}(params, pk_A, W_1)$,
- and a trapdoor $T_{W_2} = \texttt{SE-Trapdoor}(params, sk_A, W_2)$.

It outputs 1 if $W_1 = W_2$ and 0 otherwise.

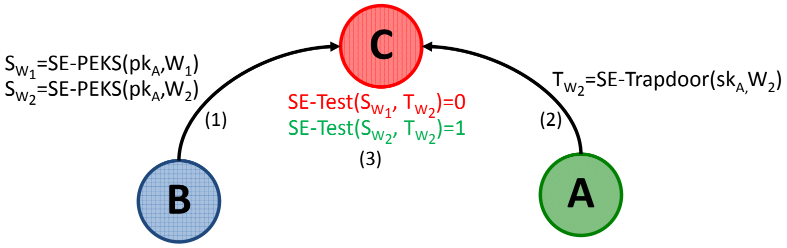

S_{W_1}=SE-PEKS(pk_A,W_1)
S_{W_2}=SE-PEKS(pk_A,W_2)
(1)
C
SE-Test(S_{W_1}, T_{W_2})=0
SE-Test(S_{W_2}, T_{W_2})=1
(3)
T_{W_2}=SE-Trapdoor(sk_A,W_2)
(2)
B
A

Figure 7.1: Functional description of PEKS. B is sending two searchable encryptions S_{W_1} and S_{W_2} corresponding respectively to W_1 and W_2. A gives C the trapdoor T_{W_2} corresponding to W_2. C can then test the PEKS values received with the trapdoor and detects that S_{W_2} corresponds to the trapdoor T_{W_2} whereas S_{W_1} does not.

Figure 7.1 illustrates PEKS in a classical scenario with three nodes. In this scenario, node B is sending a message to node A through node C. Node A wants to retrieve the message from node C only if they correspond to a certain keyword.

- Node B uses `SE-PEKS` to generate searchable encryptions S_{W_1} and S_{W_2} of words W_1 and W_2 respectively, and we assume $W_1 \neq W_2$. Node B sends S_{W_1} and S_{W_2} to node C.

- We assume node A is only interested in keyword W_2. To this extent, node A generates T_{W_2}, a trapdoor for W_2. Only A can generate such a trapdoor as the `SE-Trapdoor` algorithm requires the private key sk_A of A. Node A sends T_{W_1} to node C.

- Node C compares the trapdoor and the searchable keywords it received with the `SE-Test` function. Nodes C learns that T_{W_2} matches S_{W_2} but does not match S_{W_1}. Node C does not learn the value of either W_1 or W_2 though.

We now present an efficient construction based on bilinear maps to implement the PEKS algorithms presented in [BCOP04].

7.3.2 Construction

The construction proposed by Boneh et al. proceeds as follows:

1. `SE-Setup`: Given a security parameter $sp \in \mathbb{Z}^+$ the algorithm works as follows:

   (a) Run $\mathcal{G}$ on input sp, generate a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a cryptographic bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

   (b) Choose a generator P of $\mathbb{G}_1$.

   (c) Choose two cryptographic hash functions:

   - $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$,
   - $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^{\log q}$.

   Return the system parameters:

$$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, H_1, H_3 \rangle \, .$$

2. `SE-KeyGen`: Given the system parameters $params$, the algorithm picks a random $s \in \mathbb{Z}_q^*$. It outputs $pk_A = sP$ and $sk_A = s$.

3. `SE-PEKS`$(params, pk_A, W)$: Given a keyword W and a public key $pk_A = sP$, the algorithm randomly chooses $r \in \mathbb{Z}_q^*$ and returns the tuple:

$$S_W = \langle rP, H_3(\hat{e}(H_1(W), rpk_A)) \rangle \, .$$

4. `SE-Trapdoor`$(params, sk_A, W)$: given a private key $sk_A = s$ and a keyword W the algorithm returns:

$$T_W = sk_A H_1(W).$$

5. `SE-Test`$(params, S_{W_1}, T_{W_2})$: given $S_{W_1} = \langle U, V \rangle$ and T_{W_2} the algorithm outputs

   - 1 if $H_3(\hat{e}(T_{W_2}, U)) = V$,
   - 0 if $H_3(\hat{e}(T_{W_2}, U)) \neq V$.

We summarize the notations used in this section in Table 7.3.2.

We can easily verify that the scheme is consistent in that:

- if $S_{W_1} =$ `SE-PEKS`$(params, pk_A, W_1)$,
- and $T_{W_2} =$ `SE-Trapdoor`$(params, sk_A, W_2)$,
- and $W_1 = W_2$,

then `SE-Test`$(params, S_{W_1}, T_{W_2}) = 1$.

Conversely, if $W_1 \neq W_2$ `SE-Test`$(params, S_{W_1}, T_{W_2}) = 0$ with overwhelming probability. `SE-Test` would indeed return 1 in case there is a collusion on H_1 (probability $2^{-\frac{q}{2}}$) or on H_3 (probability $2^{-\frac{\log q}{2}}$). Given that a typical value for $\log q$ is at least 160, those probabilities are negligible.

The PEKS scheme being described, we present our specific instantiation to meet the requirements of user privacy in context-based forwarding in the next section.

sp	security parameter
$params$	system parameters
$\mathcal{G}$	a group and bilinear map generator
q	a prime number
$\mathbb{G}_1, \mathbb{G}_2$	groups of order q
P	a generator of $\mathbb{G}_1$
$\hat{e}$	a bilinear map
H_1, H_3	cryptographic hash functions
A, B, C	parties involved in the protocol
r, s	random numbers in $\mathbb{Z}_q^*$
pk_A	public key of A
sk_A	private key of A
W, W_1, W_2	keywords
S_W	searchable encryption of W
T_W	trapdoor for W

Table 7.1: Notations used in the description of PEKS

7.4 Searchable Encryption for User Privacy in MobiOpps

As described in section 5.3.2, while the protection of user privacy requires the encryption of the header, intermediate nodes should still be able to compare their profile to the context of the message in order to correctly execute the protocol. PEKS enables intermediate nodes to perform searches on encrypted data without accessing the data. While PEKS seems a good candidate for user privacy, this scheme cannot directly be implemented since, in the original version of PEKS, the sender needs to retrieve the public key of the destination in order to compute `SE-PEKS` which is impractical in opportunistic networks. Furthermore, `SE-Test` requires a trapdoor T_W that can only be computed by the destination. In context-based forwarding, the destination is not known through its identity though. Furthermore, even if we define the destination through its context, requiring that the destination computes the trapdoors for its profile and send it to all the nodes is not only impractical but also defeats the purpose of protecting the destination's privacy.

We therefore propose to adapt PEKS functions to the environment defined in section 5.2 in a way inspired by identity-based encryption. The latter alleviates the need for the destination's public key required in classical asymmetric schemes by replacing the destination's public key by the public key of a Trusted Third Party (TTP) -called private key generator-, and the identity of the destination. Similarly, we propose to use PEKS with the public key of a TTP instead of the public key of the destination. As a consequence, neither the public key of the destination nor the destination's identity itself need to be known. The source can generate searchable encryptions of the attributes in the header with the knowledge of the public key of TTP only. The TTP gives each node trapdoors corresponding to their profile so that they can detect matches between their profile and the header of the message while non-matching attributes remain secret: the trapdoors are matching capabilities.

An example of this new instantiation of PEKS is sketched in figure 7.2 and the notations are summarized in table 7.4.

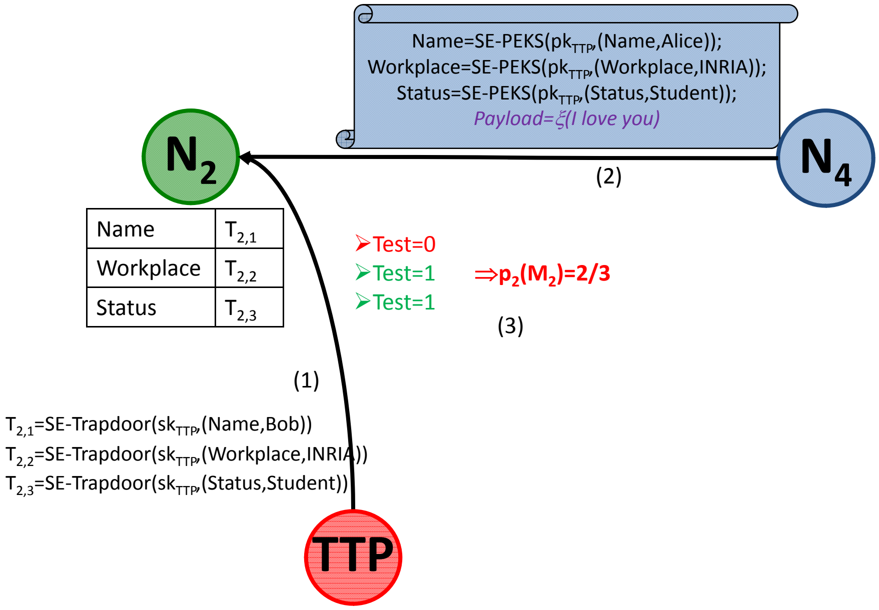

Name=SE-PEKS(pk$_{TTP}$,(Name,Alice));
Workplace=SE-PEKS(pk$_{TTP}$,(Workplace,INRIA));
Status=SE-PEKS(pk$_{TTP}$,(Status,Student));
Payload=ξ(I love you)
N$_2$
N$_4$
(2)
Name T$_{2,1}$
Workplace T$_{2,2}$
Status T$_{2,3}$
➢Test=0
➢Test=1
➢Test=1
⇒p$_2$(M$_2$)=2/3
(3)
(1)
T$_{2,1}$=SE-Trapdoor(sk$_{TTP}$,(Name,Bob))
T$_{2,2}$=SE-Trapdoor(sk$_{TTP}$,(Workplace,INRIA))
T$_{2,3}$=SE-Trapdoor(sk$_{TTP}$,(Status,Student))
TTP

Figure 7.2: Example of user privacy mechanism in scenario (II). The TTP gives nodes (here N_2) the trapdoors associated with their profile. The sender N_4 computes searchable encryptions of the attributes of the header based on the public key of the TTP. Intermediate nodes like N_2 can then test the encrypted header with the trapdoors they own to compute the matching ratio $p_2(M_2)$. ξ denotes the PAYLOAD_ENCRYPTION primitive.

Our solution features two phases as described in section 5.4 and we describe these two phases in detail in the sequel of this section.

7.4.1 Setup Phase

During the setup phase, we assume the availability of a Trusted Third Party TTP which main task is key management.

The TTP first runs the algorithm `SE-Setup`: it obtains *params* which are system parameters. The TTP then runs `SE-KeyGen` and generates a public/private key pair denoted

sp	security parameter
$params$	system parameters
TTP	Trusted Third Party
pk_{TTP}	public key of the TTP
sk_{TTP}	private key of the TTP
$A_{i,j}$	j-th attribute of node N_i
$T_{i,j}$	trapdoor for $A_{i,j}$
L_M	set of indexes of attributes of message M
$A_{M,j}$	attribute of message M with index j
$S_{M,j}$	searchable encryption of $A_{M,j}$
$p_i(M)$	matching ratio of node N_i with message M

Table 7.2: Notations used in our instantiation of PEKS

by pk_{TTP}/sk_{TTP}.

During the setup phase, nodes contact the TTP to fetch the trapdoors corresponding to their profile. We assume that each node N_i communicates with the TTP through a secure communication channel, and we do not detail the process of establishing such a secure channel. The communication process is the following:

1. The TTP sends $params$ and pk_{TTP} to N_i.

2. N_i sends its profile $Prof(i) = ||_{1 \leq j \leq m} A_{i,j}$ to the TTP.

3. The TTP:

   (a) verifies the validity of $Prof(i)$,

   (b) computes trapdoors $T_{i,j} = \texttt{SE-Trapdoor}(params, sk_{TTP}, A_{i,j})$ for $1 \leq j \leq m$,

   (c) sends the set of trapdoors $\{T_{i,j}\}_{1 \leq j \leq m}$ to N_i.

In our instantiation of PEKS, the trapdoors generated by the TTP are secrets like private keys: nodes should not reveal them to other parties.

Similarly to the payload encryption solution, the keyword used as an input in `SE-Trapdoor` is the concatenation of both the attribute name and attribute value. This guarantees that trapdoors corresponding to different attributes are different.

At the end of the setup phase, N_i receives the system parameters $params$, the public key of the TTP pk_{pub} and m secrets $T_{i,j}$ for $1 \leq j \leq m$. N_i does not need to contact the TTP further.

7.4.2 Runtime Phase

We assume that all nodes involved in this phase have already performed the setup phase and that they have the trapdoors associated with their profile.

The encryption of the header $\mathcal{H}(M) = ||_{j \in L_M} A_{M,j}$ uses the `SE-PEKS` primitive of $\mathcal{PEKS}$ to produce searchable encryptions $S_{M,j}$ of each attribute $A_{M,j}$ in the header with the public key of the TTP. The source node N_S hence constructs an encrypted header $\mathcal{H}(M')$ as follows:

$$\mathcal{H}(M') = ENCRYPT\_HEADER(M) = ||_{j \in L_M}(E_j, S_{M',j}),$$

where, for each $j \in L_M$,

$$S_{M',j} = \texttt{SE-PEKS}(params, pk_{TTP}, A_{M,j}).$$

In order for an intermediate node N_i to compute the matching ratio, N_i uses the `SE-Test` function on the attributes of the header. The other required input in `SE-Test` are the trapdoors which were received during the setup phase. Then, N_i implements MATCH_HEADER by computing for each $j \in L_M$:

$$\texttt{SE-Test}(S_{M',j}, T_{i,j}),$$

which outputs:

- 1, if $A_{M,j} = A_{i,j}$,
- 0, if $A_{M,j} \neq A_{i,j}$.

N_i is then able to construct the set $L_{M,i}$ of indexes of attributes shared between M and N_i and to compute the matching ratio $p_i(M) = \frac{|L_{M,i}|}{|L_M|}$.

7.5 Evaluation

In this section we show that the proposed specific instantiation of PEKS meets the requirements of user privacy in context-based forwarding.

Firstly, the proposed solution features a TTP only during the setup phase. This TTP is required to generate system parameters and distribute trapdoors to each node. However the the runtime phase does not require the online availability of the TTP, hence does not imply a conflict with the opportunistic nature of the communication.

Secondly, the solution enables any node to compute an encrypted version of the header of the message with any attribute. The encryption function uses `SE-PEKS` which requires only the public key of the TTP which is distributed during the setup phase and known by all nodes: the encryption function is thus public. Furthermore, `SE-PEKS` is a randomized function as presented in section 7.3.2: `SE-PEKS` internally generates and uses a random number to produce different outputs at each execution, even if the input does not change. The privacy mechanism thus avoids the dictionary attack following the requirements described in section 7.2.3.

Another advantage of randomization is protection of the privacy of the source as well. The privacy of the source here does not refer to the attributes of the source, but to the nodes with which the source communicates. Indeed, if the header of all messages addressed to a given destination are always the same, then it is possible to deduce the frequency of communication between a source and a destination simply by observing the headers of successive messages and linking similar headers together. Randomization of the headers prevent this attack: headers of successive messages to the same destination are different each time, which helps protecting another aspect of privacy.

Finally, intermediate nodes are able to compute the matching ratio while the privacy of the destination is preserved. Indeed, even though the `SE-Test` function is public, it requires trapdoors as input, and each node only has the trapdoors corresponding to its own profile. Furthermore nodes cannot compute other trapdoors, because the `SE-Trapdoor` function requires the private key sk_{TTP} of the TTP.

To be more precise, Boneh et al. proved in [BCOP04] that their construction is semantically secure against a chosen keyword attack in the random oracle model, assuming that the Bilinear Diffie-Hellman problem is hard. This means in particular that it is unfeasible for a node to discover the value of an attribute unless it knows the corresponding trapdoor. If the node does not have the corresponding trapdoor, it can in fact not distinguish the searchable encryption from a random string. Furthermore, since only the TTP knows the private key sk_{TTP}, nodes cannot forge trapdoors. Recovering the private key sk_{TTP} amounts to a discrete logarithm computation in $\mathbb{G}_1$, where discrete logarithm computation are assumed to be hard.

Hence only the TTP can compute and distribute the trapdoors. Since the TTP provides a node N_i only with the trapdoors corresponding to its profile, this implies that N_i can only discover attributes of the header corresponding to its profile by using the `SE-Test` function. Hence, each node can only match attributes with its own profile as required in section 7.2.3.

Note that, in our setting, this primitive reveals the value of the matching attributes, because intermediate nodes know the value of the attributes of their profile and the corresponding trapdoors. This is different from the original PEKS setting where an intermediate node would receive a trapdoor and a searchable encryption and perform a matching test that does not reveal any information on the value of the involved keywords. This solution therefore does not achieve full privacy but fits under the adaptable privacy model (model 3) described in section 3.4 where the trust level relies on the trusted community assumption introduced in definition 3.2.1.

7.6 Summary

In summary, the proposed solution enables to compute the matching ratio and to take forwarding decisions while preserving destination's privacy: a node N_k can compute its matching ratio $p_k(M)$ without discovering non-matching attributes and send the result to N_i, which then takes a forwarding decision based on $p_k(M)$. This raises another problem which is not addressed by the privacy-preserving solution namely the issue of trusting the value $p_k(M)$ advertised by N_k. To this extent, we enhance the privacy preserving mechanism in the next chapter with a proof of computation in order to ensure the correctness of the advertised matching ratio.

Chapter 8

Computation Assurance Proposal

8.1 Introduction

In context-based forwarding, a node N_i takes forwarding decisions based on the matching ratio $p_k(M)$ received from its neighbor N_k. Hence, it is important to ensure that the matching ratios advertised by neighbors are correct. If the neighbor N_k of N_i is honest-but-curious then there is no problem of computation assurance, as N_k does not stray from the communication protocol, thus it advertises a correct matching ratio. If N_k is malicious however, N_k can advertise a high matching ratio to attract messages towards him to subvert the communication or disrupt communication by dropping all messages, thus acting as a black-hole. Guaranteeing the correctness of the matching ratio is thus important for the network resilience against this type of Denial of Service (DoS) attack.

A simple way of avoiding such maliciousness is to modify the context based forwarding protocol: instead of requiring the matching ratio $p_k(M)$ from N_k, N_i could request the profile $Prof(k)$ of its neighbor N_k. N_i could then compute the matching ratio $p_k(M)$ locally and would not have to trust N_k. This simple solution is not satisfactory however, as it deeply modifies the protocol and raises additional privacy concerns because N_i would then access $Prof(k)$.

Hence, computation assurance is a separate issue from user privacy, but it still has an impact on it.

In this chapter we propose an original approach to provide a node with the assurance that the matching ratio advertised by its neighbor is correct, without exposing the privacy of the neighbors. We first present a basic idea (section 8.2) inspired by commitment schemes (section 8.2.1): we use cryptographic hash functions and more precisely the preimage of a digest as a proof of a match computation. Then, in section 8.3, we propose to enhance the privacy and efficiency of the solution by using counting Bloom filters [FCAB00] in an original way to enable the computation of the proved global matching ratio. Finally, we analyze the security of this solution in section 8.4.

8.2 Basic Idea

8.2.1 Commitment Schemes

The first idea consists in using cryptographic hash functions to give a proof of knowledge similarly to commitment schemes which were first formalized by Brassard et al. in [BCC88]. The basic scenario is a two-party (Alice and Bob) protocol composed of two phases: a commit and a reveal phase. In this scenario, Alice knows a secret information $info$ and she wants to prove to Bob that she knows this $info$ without revealing it, until the reveal phase.

Definition 8.2.1 *A commitment scheme is a protocol between two parties Alice, who knows some information $info$, and Bob, who wants to verify that Alice indeed knows $info$. The protocol takes place in two phases:*

- ***Commit phase:*** *Alice sends a* **commitment** *message depending on $info$ to Bob.*
- ***Reveal phase:*** *Alice sends an* **opening** *message, which enables Bob to verify the commitment.*

It is essential that $info$ cannot be discovered by the receiver before the Reveal phase (this is called the hiding property). A simple Reveal phase would consist of a single message, the opening, from the sender to the receiver, followed by a check performed by the receiver. The value chosen during the commit phase must be the only one that the sender can compute and that validates (i.e. that can be verified by the receiver) during the reveal phase (this is called the binding property). Therefore, the commitment chosen by *Alice* have to verify two essential properties:

Property 8.2.2 ***Hiding property:*** *The commitment is hiding: Bob cannot retrieve $info$ from the commitment.*

Property 8.2.3 ***Binding property:*** *The commitment must be the only value that Alice can compute from $info$ which validates during the reveal phase.*

We describe a simple example of commitment scheme. Assume *Alice* randomly chooses some information $secret \in \{0,1\}^*$, and that *Alice* and *Bob* know a cryptographic hash function *hash* (see section 6.2.1). *Alice* interacts with *Bob* as follows:

- Commit phase: *Alice* sends $hash(secret)$ to Bob. The commitment is $hash(secret)$.
- Reveal phase: *Alice* sends *secret* to *Bob*. The opening message is *secret* itself.

In this example, the commitment is:

- Hiding: the first preimage resistance property of *hash* implies that *Bob* cannot retrieve *secret* from the commitment. Furthermore we stated that *secret* was a randomly chosen bit string, hence dictionary attack is not possible.

M'	encrypted message
$\mathcal{H}(M')$	encrypted header of the message
L_M	Set of indexes of attributes in $\mathcal{H}(M')$
$A_{M,j}$	j-th attribute of M
$S_{M',j}$	tuple representing a searchable encryption of $A_{M,j}$
$r_{M',j}$	random number used in $S_{M',j}$
H_3	cryptographic hash function
$x_{M',j}$	preimage (under H_3) of the second element of the tuple $S_{M',j}$
N_k	node N_k, neighbor of N_i
$T_{k,j}$	trapdoor of node N_k for the j-th attribute
$L_{M,k}$	set of indexes of attributes shared between N_k and M

Table 8.1: Notations used in the description of the basic computation assurance scheme

- Binding: the collusion resistance property of *hash* implies that *Alice* cannot compute two messages with the same digest.

Hence at the end of the protocol *Bob* is convinced that *Alice* knew *secret* before sending the commitment. The commitment scheme therefore provides a solution to prove knowledge of a secret without revealing the secret in a first step.

8.2.2 A Fundamental Building Block for Computation Assurance

We propose an approach inspired by commitment scheme to prove the correctness of a match (shared attributes) without revealing the attribute. We summarize the notations used in this section in Table 8.2.2.

In our scenario, a node N_i has an encrypted message M' with an encrypted header $\mathcal{H}(M')$. N_i sends to its neighbor N_k (see section 7.4.2):

$$\mathcal{H}(M') = ENCRYPT\_HEADER(M) = ||_{j \in L_M}(E_j, S_{M',j}),$$

where, for each $j \in L_M$,

$$S_{M',j} = \texttt{SE-PEKS}(params, pk_{TTP}, A_{M,j}).$$

N_k then computes the matching ratio $p_k(M')$ corresponding to the number of shared attributes between $Prof(k)$ and $\mathcal{H}(M')$ with the `SE-Test` function.

Our goal now is to define how N_i verifies that the matching ratio $p_k(M')$ advertised by N_k indeed corresponds to the number of shared attributes between $Prof(k)$ and $\mathcal{H}(M')$.

To this extent, we first look in the details of the construction of the PEKS primitives described in section 7.3.2 and make several useful observations.

First, the searchable encryption of an attribute $A_{M,j}$ is a tuple:

$$S_{M',j} = \left\langle r_{M',j}P, H_3(\hat{e}(H_1(A_{M,j}), r_{M',j}pk_{TTP}))\right\rangle = \left\langle r_{M',j}P, H_3(x_{M',j})\right\rangle,$$

where $x_{M',j} = \hat{e}(H_1(A_{M,j}), r_{M',j}pk_{TTP}) \in \mathbb{G}_2^*$.

The node N_k, which owns the trapdoor $T_{k,j}$, uses the `SE-Test` function on $S_{M',j}$. This function checks whether:

$$H_3(\hat{e}(T_{k,j}, r_{M',j}P)){=}H_3(x_{M',j})$$

or not. This check is equivalent to checking whether:

$$\hat{e}(T_{k,j}, r_{M',j}P) = x_{M',j}$$

or not.

If N_k shares the j-th attribute with M, $A_{k,j} = A_{M,j}$, then N_k is able to compute $x_{M',j}$ by using $T_{k,j}$ and $S_{M',j}$. Moreover, since H_3 is a cryptographic hash function and hence is first preimage resistant, nodes which do not share the attribute cannot retrieve $x_{M',j}$ from $S_{M',j}$. Contrary to the attribute $A_{M,j}$, $x_{M',j}$ is an element of $\mathbb{G}_2$ which is randomized with $r_{M',j}$ and is therefore not prone to dictionary attacks.

Based on these observations we now present the idea of the protocol:

1. N_i sends the header $\mathcal{H}(M')$ of the message M' to N_k: $\mathcal{H}(M')$ is the commitment.

2. For each $j \in L_{M,k}$, the matching set of N_k and M', N_k is able to retrieve $x_{M',j}$. N_k sends the set $\{x_{M',j}\}_{j \in L_{M,k}}$ to N_i: this set is the opening.

3. For each $j \in L_{M,k}$, N_i computes $H_3(x_{M',j})$ and simply checks that it is equal to the second element in the tuple $S_{M',j}$. N_i then computes the guaranteed matching ratio $p_k(M) = \frac{|L_{M,k}|}{|L_M|}$.

This protocol is inspired by the commitment scheme with some modification: the commitment and openings are provided by different entities. N_i commits the digest of a secret $x_{M',j}$ (N_i might not even know this secret) and N_k reveals that he knows this secret for matching attributes.

8.2.3 Analysis

The protocol presented in the previous section is based on the use of cryptographic hash function H_3 and pseudorandom values $x_{M',j}$. By exhibiting $x_{M',j}$, N_k proves to N_i that the j-th attribute is shared between M and N_k. Thus, this protocol solves the computation assurance problem.

There are yet two issues with this approach. First, from a privacy perspective, N_i now discovers separately which attributes N_k shares with M. This means that even though N_i

does not know what the attribute values are, N_i knows which attribute names are shared between N_j and M. This comes from the fact that each match is verified separately instead of verifying the whole matching ratio at once.

Second, from a communication overhead perspective, the amount of information sent by N_k rises from just a ratio (one floating number) to all matching preimages $x_{M',j} \in \mathbb{G}_2^*$ which require $\log_2 q$ bits each. In some situations this can be a severe drawback, and it would be interesting to have a homomorphic-like property such that N_k would send only one compact proof to pledge for its honesty.

Based on the previously described simple commitment scheme and considering these two issues, we propose, in the next section, a new scheme which combines PEKS with counting Bloom filter.

8.3 Efficient Solution

In this section we propose an efficient scheme that provides privacy-preserving computation assurance. We first define the properties of counting Bloom filters, then we describe our proposal and finally we prove its security. The notations used in this section are summarized in Table 8.3.

8.3.1 Counting Bloom Filters

A Bloom filter is a probabilistic data structure which was first introduced by Burton Bloom ([Blo70]). The classical use of Bloom filters is to test whether an element is a member of a set in a space-efficient way. Bloom filters support dynamic addition of an element in the set represented by the Bloom filter but the deletion is impossible without reconstructing the whole Bloom filter from scratch. The only drawback of Bloom filters is the false positive rate: a false positive occurs when the Bloom filter test is positive for an element which does not belong to the set represented in the Bloom filter.

Broder and Mitzenmacher present a comprehensive survey of the applications of Bloom filters in networking in [BM02], where they aptly state the Bloom filter principle as: *"Wherever a list or set is used, and space is a consideration, a Bloom filter should be considered. When using a Bloom filter, consider the potential effects of false positives."*

We focus on an extension of Bloom filters called counting Bloom filters that were proposed by Fan et al. in [FCAB00] to support the dynamic deletion of an element.

Definition 8.3.1 ***A counting Bloom filter*** *CBF is an array of ϕ positions (also called buckets) used to represent a set $\mathcal{X}$. Counting Bloom filters implement the following functions:*

- *`Query`(x,CBF): on input of an element x, and a counting Bloom filter representing $\mathcal{X}$, returns 1 if $x \in \mathcal{X}$ and 0 otherwise.*
- *`Insert`(x,CBF): on input of an element x, modifies CBF such that it represents $\mathcal{X} \cup x$.*

- `Delete`*(x,CBF): on input of an element $x \in \mathcal{X}$, modifies CBF such that it represents $\mathcal{X} - x$.*

All those functions require a predefined set of t hash functions, denoted by h_1,...,h_t.

We use the counting Bloom filter structure to represent a set of preimages $x_{M',j}$, which are elements of $\mathbb{G}_2^*$.

CBF is initialized to zero: $CBF[i] = 0$ for $0 \leq i \leq \phi - 1$.

The functions $h_1, ..., h_t$: $\mathbb{G}_2 \rightarrow [0, \phi - 1]$ are hash functions which map an element of $\mathbb{G}_2$ to one of the ϕ array positions with a uniform random distribution. These hash functions are not necessarily cryptographic hash functions.

To insert an element x in CBF, we compute the digest of the element from each of the t hash functions and increment the value of the filter at these positions:

$$\texttt{Insert}(x, CBF)\text{: for } 1 \leq i \leq t\text{, } CBF[h_i(x)] \leftarrow CBF[h_i(x)] + 1.$$

Conversely, the delete operation consists of decrementing the value of each of the respective buckets:

$$\texttt{Delete}(x, CBF)\text{: for } 1 \leq i \leq t\text{, } CBF[h_i(x)] \leftarrow CBF[h_i(x)] - 1.$$

To query for an element x, we compute the t digests $h_i(x)$ for $1 \leq i \leq t$. If any of the values $CBF[h_i(x)]$ at these positions are 0, the element is not in the set. If all are non-zero, then either the element is in the set, or the bits have been incremented during the insertion of other elements (false positive case).

$$\texttt{Query}(x, CBF)\text{: if } \exists i \in [1, t]/CBF[h_i(x)] = 0 \text{ return } 0 \text{ else return } 1.$$

The false positive ratio can be made as small as required by carefully choosing the array size ϕ and the number t of hash functions in function of the number of elements in the set.

In Figure 8.1 we present a simple example of a counting Bloom filter with three elements.

We also define the weight w_{CBF} of a counting Bloom filter CBF as the sum of the values of all positions: $w_{CBF} = \sum_{0 \leq i \leq r-1} CBF[i]$.

Property 8.3.2 *Let CBF be a counting Bloom filter constructed with t hash functions and representing a set $\mathcal{X}$. Then, the weight w_{CBF} of CBF is linearly dependent on the cardinal $|\mathcal{X}|$ of $\mathcal{X}$. To be more precise, we have:*

$$w_{CBF} = t|\mathcal{X}|.$$

The property follows directly from the construction of counting Bloom filters and the definition of their weight. Note that this property is not true however in classical Bloom filters. In fact the main reason why we choose counting Bloom filters over classical ones is to obtain this property.

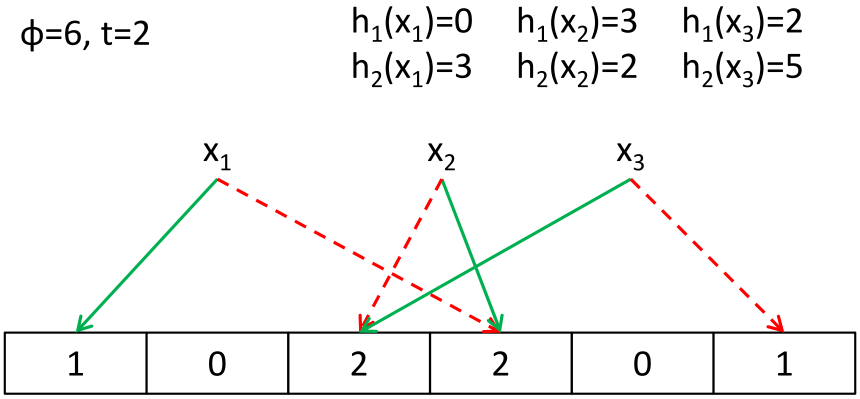


Figure 8.1: Example of a simple counting Bloom filter. The size of the array is $\phi = 6$, and the number of hash functions is $t = 2$. In this example we inserted three elements (x_1, x_2, and x_3) in the counting Bloom filter. The plain green arrow represent the hash function h_1, while the dotted red arrow represents the hash function h_2.

Finally, we define a relation of partial order between counting Bloom filters. Let CBF_1 and CBF_2 be two counting Bloom filters of same size ϕ. We say that CBF_1 is smaller than CBF_2, and we denote this relation as $CBF_1 \prec CBF_2$ if at all positions the value of CBF_1 is smaller than the value of CBF_2. To be more precise:

$$CBF_1 \prec CBF_2 \Leftrightarrow \forall 0 \leq i \leq \phi - 1, \;\; CBF_1[i] \leq CBF_2[i].$$

With this definition of partial order we have the following property:

Property 8.3.3 *Let CBF_1 and CBF_2 be two counting Bloom filters verifying $CBF_1 \prec CBF_2$. Then $w_{CBF_1} \leq w_{CBF_2}$.*

This concludes the overview of the main functionalities and parameters of counting Bloom filters and we now focus on the original use of this structure to enhance privacy and efficiency on the computation assurance solution.

8.3.2 Combining PEKS and Counting Bloom Filters

To improve the basic scheme of section 8.2.2, the idea is that, whenever N_i sends $\mathcal{H}(M')$ to N_k, N_k inserts all the $x_{M',j}$ it manages to compute in a counting Bloom filter instead of sending them separately. By doing so N_i will only compute the matching ratio without even discovering which attribute names are shared between the message and N_k. However, since $x_{M',j}$ is included in a counting Bloom filter and thus is not in clear anymore, a counting Bloom filter, computed by the source, is forwarded along with the message in order to allow N_i to perform the required verification operations. The purpose of the counting Bloom filters is not to verify that an element belongs to a set as in the classical use, but to verify the cardinality of a set intersection.

To be more precise, the source N_S constructs the encrypted header of the message M' as (see section 7.4.2):

$$\mathcal{H}(M') = ENCRYPT\_HEADER(M) = ||_{j \in L_M}(E_j, S_{M',j}),$$

where, for each $j \in L_M$,

$$S_{M',j} = \texttt{SE-PEKS}(params, pk_A, A_{M,j}).$$

Additionally, N_S constructs a counting Bloom filter $CBF_S(M')$ representing the set of preimages $\{x_{M',j}\}_{j \in L_M}$. To this extent, N_S creates an empty counting Bloom filter $CBF_S(M')$ and then performs:

$$\texttt{Insert}(x_{M',j}, CBF_S(M')),$$

for each $j \in L_M$.

The parameters of the counting Bloom filter are public (hash functions used and their number t, array size ϕ) but the inserted data is not. There are two options to define the general parameters ϕ and the t hash functions:

- either each source node decides to set up these parameters independently, and then the parameters are sent along with the message: this solution offers flexibility as it enables the source node to choose t, which is a security parameter as will be explained in 8.4.3, but it incurs a non-negligible communication overhead,
- or a Trusted Third Party generates the parameters and distributes them during the offline setup phase: this solution is less flexible but more efficient from a communication perspective.

The counting Bloom filter $CBF_S(M')$ serves as a matching reference and N_S sends $CBF_S(M')$ along with the message M'.

We now assume that the message M' reaches N_i along with the counting Bloom filter $CBF_S(M')$. We also assume that N_k does not know the counting Bloom filter $CBF_S(M')$ created by the source otherwise there is no challenge for N_k, since N_k could simply piggyback $CBF_k(M') = CBF_S(M')$. Therefore it is important to protect communication between neighbors with simple hop-by-hop encryption mechanisms to prevent eavesdroppers from overhearing the counting Bloom filters exchanged and then being able to claim a matching ratio of 1.

The communication process carries on as follows:

1. N_i sends the header $\mathcal{H}(M')$ to N_k. In addition, N_i also informs N_k about the public parameters of $CBF_S(M')$ but it does not send $CBF_S(M')$.
2. N_k constructs a new empty counting Bloom filter $CBF_k(M')$ with the same parameters as the one constructed by N_S. Then, for each $j \in L_{M,k}$:

- N_k computes the preimages $x_{M',j}$,
- N_k adds the preimage to $CBF_k(M')$ by performing $\texttt{Insert}(x_{M',j}, CBF_k(M'))$.

N_j then sends $CBF_k(M')$ to N_i.

3. N_i verifies the consistency of $CBF_k(M')$ sent by N_k with respect to the matching reference $CBF_S(M')$ by checking that:

   - $CBF_k(M') \prec CBF_S(M')$,
   - the weight $w_{CBF_k(M')}$ of $CBF_k(M')$ is a multiple of t.

   If both verifications succeed then N_i accepts the answer of N_k as being valid and computes the matching ratio as $\frac{w_{CBF_k(M')}}{w_{CBF_S(M')}}$.

Proposition 8.3.4 *If $CBF_j(M')$ is generated as specified in the protocol, then*

$$p_k(M') = \frac{w_{CBF_j(M')}}{w_{CBF_S(M')}}.$$

Proof If $CBF_j(M')$ is generated according to the protocol, it contains $L_{M,k}$ elements, thus by Property 8.3.2, $w_{CBF_k(M')} = t|L_{M,k}|$. The matching reference $CBF_S(M')$ is a counting Bloom filter containing L_M elements, hence $w_{CBF_S(M')} = t|L_M|$. Finally:

$$\frac{w_{CBF_k(M')}}{w_{CBF_S(M')}} = \frac{t|L_{M,k}|}{t|L_M|} = p_k(M').$$

▮

With this solution, N_i is now able to verify the matching ratio without discovering which attribute names match, and the answer of N_k is more space efficient (we evaluate the communication overhead in section 9.2.2).

N_i can thus choose a next hop N_k based on reliable matching ratio: N_i then sends to N_k the missing part of M' which is the payload $\mathcal{PLD}(M')$ and the matching reference $CBF_S(M')$ generated by the source N_S so that N_k can repeat the process of reliably verifying the matching ratio of its neighbors. We now evaluate the security of the proposed solution.

8.4 Security Assessment

This scheme deals with two security aspects:

- **User Privacy:** preserving the privacy of the destination is ensured by the security mechanism presented in section 7.4: this mechanism is not modified. The novelty of the reliability protocol is that the node N_k proves the matching ratio $p_k(M')$ to node N_i which might incur a privacy risk on N_k as explained in section 8.2.3. Hence, it should be hard for N_i to discover which attributes are shared between N_k and M'.

- **Computation Assurance:** node N_i should be able to verify that the matching ratio claimed by N_k is consistent with the information provided in the header $\mathcal{H}(M')$ of the message.

We analyze these two issues successively.

8.4.1 User Privacy

From a user privacy perspective, the threat is that node N_i or an eavesdropper discovers which attribute names are shared between N_k and M: the attacker is the curious N_i and the target is N_k. We do not consider the case of the eavesdropper further, as it is a weaker attacker than N_i.

We argue that the knowledge of counting Bloom Filter $CBF_k(M')$ does not enable node N_i to recover the words $x_{M',k}$ inserted in it. Thus N_i cannot dissociate the different attributes inserted and know which attributes are shared between N_k and M'.

Indeed, let us examine the easiest case for N_i: N_k inserted a single element (e.g. $x_{M',1}$) in $CBF_k(M')$. In that case the positions $h_1(x_{M',1}); ...; h_t(x_{M',1})$ are incremented in $CBF_j(M')$. The security argument is based on two main observations:

- The first observation is that the hash functions $h_1, ..., h_t$ are not invertible, even though they are not necessarily cryptographic hash functions. The reason is that those functions map elements of $\mathbb{G}_2$ (a group of order q) to a small set (the integers smaller than ϕ). Therefore there are many preimages corresponding to the output of each function: if the hash functions have a uniformly distributed output then each output has $\frac{q}{\phi}$ preimages. Said more formally, for all $1 \leq i \leq t$ and all $0 \leq y \leq \phi - 1$, the equation $h_i(x) = y$ has $\frac{q}{\phi}$ solutions. If we combine the t equations corresponding to the t hash functions in an equation system, the number of inputs verifying simultaneously t conditions on their digest is $\frac{q}{\phi^t}$ or more formally stated:

$$\forall (y_1, ..., y_t) \in [0, \phi - 1]^t, |\{x \in \mathbb{G}_2 / h_1(x) = y_1, ..., h_t(x) = y_t\}| = \frac{q}{\phi^t}.$$

  This result indicates that the size of the set of solutions to the system of t equations, that can be derived from the counting Bloom filter is still large since $q >> \phi^t$ (see section 9.2.4 for numerical results).

- The second observation is that the order of the hash functions is lost once the element is inserted in the counting Bloom filter: there is an information loss in the construction of this structure. Therefore, it is impossible to know which hash function resulted in the incrementation of a given position in the filter and this further increases the size of the potential preimages: it is possible to set many systems of equations for the same counting Bloom filter. The number of systems of equations for a counting Bloom filter containing only one element $x_{M',1}$ varies from 1 (if all the hash functions incremented the same position, that is to say if $h_1(x_{M',1}) = h_2(x_{M',1}) = ... = h_t(x_{M',1})$) to $t!$ (if the hash functions output different positions each). Out of all these different system of equations only one leads to the right set of possible preimages.

As a result, the set of possible preimages corresponding to a counting Bloom filter containing a single element is at least $\frac{q}{\phi^t}$. This is a lower bound on the set of preimages but the actual result can be multiplied by a factor of up to $t!$ depending on the outputs of the hash functions. And if more than one element is inserted in the counting Bloom filter the set of preimages is multiplied even further.

This result does not even take into account the computational complexity required to find the corresponding set of preimages: Even in the eventuality of the existence of an efficient method for finding the set of preimages of the hash functions $h_1, ...h_t$, it would still be impossible to single out $x_{D,1}$ from $CBF_k(M')$ but only to find a set of $\frac{q}{\phi^t}$ preimages that would lead to the same counting Bloom filter.

From the perspective of a brute force attacker, being able to solve the equations would lead to an advantage as it reduces the size of the space of possibilities from q down to $\frac{q}{\phi^t}$. However, careful setting of the parameters q, ϕ and t, makes the size of this set large enough to prevent brute force guessing (see section 9.2.4).

Finally, an important point to keep in mind is that $x_{D,1}$ is a pseudorandom string that has no semantic: the randomness involved in the construction of SE-PEKS guarantees that even the computation of $x_{D,1}$ for an attribute $A_{D,1}$ changes at each execution of the protocol. It is therefore infeasible to link the entries of the counting Bloom filter based on successive observations.

In summary, the counting Bloom filter cannot be reversed to obtain the entries that were inserted in it, which guarantees that the computation assurance scheme does not affect the privacy of node N_k.

We now focus on the analysis of the computation assurance property.

8.4.2 Computation Assurance

Concerning computation assurance, the roles are reversed with respect to the attacker model defined in the previous section: the attacker becomes N_k, neighbor of N_i. N_i wants to verify if the matching ratio claimed by N_k corresponds to the legitimate matching ratio between N_k and M'. On the contrary N_k wants to convince N_i that its matching ratio is higher than the legitimate matching ratio.

To clarify the matter we introduce the following notations. We denote by:

- $p_{k_{legit}}(M')$, the legitimate matching ratio between M' and N_k, which is computed as $\frac{|L_{M,k}|}{|L_M|}$.
- $p_{k_{claim}}(M')$, the matching ratio claimed by N_k.

With these new notations, the goal of N_i is to verify that $p_{k_{claim}}(M') \leq p_{k_{legit}}(M')$, while the goal of a malicious node N_k is to claim a matching ratio $p_{k_{claim}}(M')$ greater than $p_{k_{legit}}(M')$. Note that N_i verifies that the claimed matching ratio is smaller than the legitimate one, but it cannot verify that the matching ratio is exactly equal. N_k can always pretend to have no match with any message. However, in this case N_k does not receive messages, hence this behavior does not subvert traffic. We thus only consider attackers who aim at claiming a higher matching ratio.

We show that the proposed solution for computation assurance, guarantees with high probability that $p_{k_{claim}}(M')$ is smaller than $p_{k_{legit}}(M')$. In fact the attacker N_k does not know whether its attempt is successful or not but the success rate is very low. And if the attack fails then N_i is aware of the attempt of N_k to cheat, which suggests the possibility of implementing reputation mechanisms to ban misbehaving nodes.

8.4.2.1 Attacker Model

We assume that N_k does not know the counting Bloom filter $CBF_S(M')$, thus the only information known by N_k on the matching reference $CBF_S(M')$ are the global parameters of the counting Bloom filter: the hash functions used h_1,...,h_t and the size ϕ. N_k also knows $\mathcal{H}(M')$ and therefore knows that the number of elements $x_{M,j}$ inserted in $CBF_S(M')$ is $|L_M|$.

The goal of the malicious N_k is to claim a matching ratio $p_{k_{claim}}(M')$ higher than $p_{k_{legit}}(M')$: with the proposed solution the matching ratio is computed by N_i based on the counting Bloom filter $CBF_k(M')$ received from N_k. To achieve its goal, N_k thus needs to claim a counting Bloom filter that leads to $p_{k_{claim}}(M')$. We denote by:

- $CBF_{k_{legit}}(M')$ the legitimate counting Bloom filter that can be constructed by N_k based on the information included in $\mathcal{H}(M')$ and the trapdoors $\{T_{k,j}\}_{1\leq j\leq m}$ owned by N_k. $CBF_{k_{legit}}(M')$ allows N_i to compute $p_{k_{legit}}(M')$.
- $CBF_{k_{claim}}(M')$ the counting Bloom filter claimed by N_k, which leads N_i to compute $p_{k_{claim}}(M')$.

For N_k to be successful, the claimed counting Bloom filter $CBF_{k_{claim}}(M')$ has to:

1. be considered valid by N_i, as required by the last step of the protocol described in section 8.3.2, which implies that:
   - $CBF_{k_{claim}}(M') \prec CBF_S(M')$,
   - the weight $w_{CBF_{k_{claim}}(M')}$ of $CBF_{k_{claim}}(M')$ is a multiple of t,
2. lead to $p_{k_{claim}}(M') > p_{k_{legit}}(M')$, hence the weight of $CBF_{k_{claim}}(M')$ needs to verify $w_{CBF_{k_{claim}}(M')} > w_{CBF_{k_{legit}}(M')}$.

Note that $CBF_{k_{claim}}(M')$, does not need to be constructed as a real counting Bloom filter by inserting elements in it with hash functions. $CBF_{k_{claim}}(M')$ simply needs to look like a legitimate counting Bloom filter, hence $CBF_{k_{claim}}(M')$ is essentially an array of size ϕ filled with integers.

In summary, we say that N_k launches a successful attack if, given h_1,...,h_t, ϕ and $|L_M|$, N_k outputs an array of integers $CBF_{k_{claim}}(M')$ of size ϕ, verifying the three following properties:

1. $CBF_{k_{claim}}(M') \prec CBF_S(M')$,
2. $w_{CBF_{k_{claim}}(M')}$ is a multiple of t,
3. $w_{CBF_{k_{claim}}(M')} > w_{CBF_{k_{legit}}(M')}$.

N_k cannot know whether the first property is met or not as N_k does not know $CBF_S(M')$. N_k can only make guesses based on the general parameters of $CBF_S(M')$. We thus present, in the next section, a probabilistic model of counting Bloom filters in order to evaluate the probability of having the three aforementioned properties validated without the knowledge of $CBF_S(M')$.

8.4.2.2 Probabilistic Modeling of Counting Bloom Filters

The goal of this section is to derive properties on the distribution of a random counting Bloom filters with known parameters.

In the sequel of this section, we denote by CBF a counting Bloom filter of size ϕ, which contains λ (unknown) elements. Each element is inserted in CBF by using the `Insert` primitive. This primitive uses t hash functions denoted h_1,...,h_t. The output of these hash functions is assumed to be uniformly distributed over $[0, \phi-1]$ (the set of indexes of CBF).

Proposition 8.4.1 *The probability distribution of the values in CBF follows a binomial distribution at each position. To be more precise:*

$$\begin{aligned}\forall 0 \leq i_1 \leq \phi-1, \forall 0 \leq i_2 \leq \lambda t, \mathcal{P}_{i_1}(i_2) &= \mathcal{P}[CBF[i_1] = i_2] \\ &= \binom{\lambda t}{i_2}\left(1-\frac{1}{\phi}\right)^{\lambda t - i_2}\left(\frac{1}{\phi}\right)^{i_2} \\ &= \mathcal{P}_{i_1}(0)\binom{\lambda t}{i_2}\frac{1}{(\phi-1)^{i_2}}.\end{aligned}$$

Proof We consider $0 \leq i_1 \leq \phi - 1$.

Since the output of the hash functions $h_1, ..., h_t$ is uniformly distributed, the probability that the output of any of these hash functions is i_1 is $\frac{1}{\phi}$. On the contrary the probability that a hash function does not output i_1 is $1 - \frac{1}{\phi}$.

There are λ elements and each element is inserted in CBF using t hash functions, therefore the weight of CBF is $w_{CBF} = \lambda t$. For $i_2 \in [1, \lambda t]$, $CBF[i_1] = i_2$ means that exactly i_2 out of the λt outputs are i_1 and the other $\lambda t - i_2$ outputs are different.

This leads to the classical binomial distribution and therefore the probability that $CBF[i] = j$ is

$$\mathcal{P}_{i_1}(i_2) = \binom{\lambda t}{i_2}\left(1-\frac{1}{\phi}\right)^{\lambda t - i_2}\left(\frac{1}{\phi}\right)^{i_2}.$$

The second part of the equality follows by rearranging the terms. We first observe that $\mathcal{P}_{i_1}(0) = (1 - \frac{1}{\phi})^{\lambda t}$. Then:

$$\begin{aligned}
\mathcal{P}_{i_1}(i_2) &= \binom{\lambda t}{i_2}\left(1-\frac{1}{\phi}\right)^{\lambda t - i_2}\left(\frac{1}{\phi}\right)^{i_2} \\
&= \left(1-\frac{1}{\phi}\right)^{\lambda t}\binom{\lambda t}{i_2}\left(1-\frac{1}{\phi}\right)^{-i_2}\left(\frac{1}{\phi}\right)^{i_2} \\
&= \mathcal{P}_{i_1}(0)\binom{\lambda t}{i_2}\left(\frac{\phi-1}{\phi}\right)^{-i_2}\left(\frac{1}{\phi}\right)^{i_2} \\
&= \mathcal{P}_{i_1}(0)\binom{\lambda t}{i_2}\left(\frac{1}{\phi-1}\right)^{i_2}
\end{aligned}$$

∎

Since the probability $\mathcal{P}_{i_1}(i_2)$ is independent of the position i_1, we simplify the notation from now on, and we write simply $\mathcal{P}(i_2)$. We have:

$$\mathcal{P}(i_2) = \mathcal{P}(0)\binom{\lambda t}{i_2}\frac{1}{(\phi-1)^{i_2}},$$

where $\mathcal{P}(0) = \left(1 - \frac{1}{\phi}\right)^{\lambda t}$.

Corollary 8.4.2 *The probability $\mathcal{P}'_{i_1}(i_2)$ that the value $CBF[i_1]$ at position i_1 is greater than a given i_2 can be computed as follows:*

$$\begin{aligned}
\forall 0 \le i_1 \le \phi - 1, \forall 1 \le i_2 \le \lambda t, \mathcal{P}'_{i_1}(i_2) &= \mathcal{P}[CBF[i_1] \ge i_2] \\
&= 1 - \sum_{i_3=0}^{i_2-1}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t - i_3}\left(\frac{1}{\phi}\right)^{i_3} \\
&= \mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\frac{1}{(\phi-1)^{i_3}}.
\end{aligned}$$

Proof The corollary follows directly from proposition 8.4.1 since

$$\begin{aligned}
\mathcal{P}[CBF[i_1] \ge i_2] &= 1 - \sum_{i_3=0}^{i_2-1}\mathcal{P}\left(i_3\right) \\
&= 1 - \sum_{i_3=0}^{i_2-1}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t - i_3}\left(\frac{1}{\phi}\right)^{i_3}
\end{aligned}$$

For the second equality we simply observe that:

$$\begin{aligned} 1 &= \left(\left(1-\frac{1}{\phi}\right)+\frac{1}{\phi}\right)^{\lambda t} \\ &= \sum_{i_3=0}^{\lambda t}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t-i_3}\left(\frac{1}{\phi}\right)^{i_3} \\ &= \sum_{i_3=0}^{i_2-1}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t-i_3}\left(\frac{1}{\phi}\right)^{i_3}+\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t-i_3}\left(\frac{1}{\phi}\right)^{i_3} \end{aligned}$$

Hence,

$$\begin{aligned} \mathcal{P}'_{i_1}(i_2) &= 1-\sum_{i_3=0}^{i_2-1}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t-i_3}\left(\frac{1}{\phi}\right)^{i_3} \\ &= \sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{\lambda t-i_3}\left(\frac{1}{\phi}\right)^{i_3} \\ &= \left(1-\frac{1}{\phi}\right)^{\lambda t}\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\left(1-\frac{1}{\phi}\right)^{-i_3}\left(\frac{1}{\phi}\right)^{i_3} \\ &= \mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\frac{1}{(\phi-1)^{i_3}} \end{aligned}$$

∎

Again, since $\mathcal{P}'_{i_1}(i_2)$ does not depend on the position, we simplify the notation and write $\mathcal{P}'(i_2)$.

Note that $\mathcal{P}'(0)=1$ since there are no negative values in a counting Bloom filter. On the opposite side, $\mathcal{P}'(i_2)=0$ for $i_2>\lambda t$ since the weight of CBF is λt.

We now give upper bounds on the probabilities $\mathcal{P}'(i_2)$ in order to further bound the probability of producing a malicious counting Bloom filter.

Lemma 8.4.3 *There exists a constant $\mathcal{C}_1$ such that, for $1 \le i_2 \le \lambda t$:*

$$\mathcal{P}'(i_2) \le \mathcal{C}_1\frac{1}{i_2!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2}.$$

Proof We start the upper bounding by using the classical inequality on the binomial coefficients: $\binom{\lambda t}{i_3} \le \frac{(\lambda t)^{i_3}}{i_3!}$.

$$
\begin{aligned}
\mathcal{P}'(i_2) &= \mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_3}\left(\frac{1}{\phi-1}\right)^{i_3}\\
&\leq \mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\frac{(\lambda t)^{i_3}}{i_3!}\left(\frac{1}{\phi-1}\right)^{i_3}
\end{aligned}
$$

The sum is reminiscent of the classical power series of the exponential ($e^x = \sum_{i_1=0}^{\infty}\frac{x^{i_1}}{i_1!}$). We therefore proceed to a change of indexes in the sum and then add positive terms to use this power series in the upper bounding:

$$
\begin{aligned}
\mathcal{P}'(i_2) &\leq \mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\frac{(\lambda t)^{i_3}}{i_3!}\left(\frac{1}{\phi-1}\right)^{i_3}\\
&\leq \mathcal{P}(0)\sum_{i_3=0}^{\lambda t-i_2}\frac{(\lambda t)^{i_2+i_3}}{(i_2+i_3)!}\left(\frac{1}{\phi-1}\right)^{i_2+i_3}\\
&\leq \mathcal{P}(0)\sum_{i_3=0}^{\infty}\frac{1}{(i_2+i_3)!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2+i_3}
\end{aligned}
$$

To bring out the power series we need to remove the i_2 offset out of the sum. To this extent, we observe that $(i_2+i_3)! \geq i_3!i_2!$, thus:

$$
\begin{aligned}
\mathcal{P}'(i_2) &\leq \mathcal{P}(0)\sum_{i_3=0}^{\infty}\frac{1}{(i_2+i_3)!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2+i_3}\\
&\leq \mathcal{P}(0)\sum_{i_3=0}^{\infty}\frac{1}{i_2!i_3!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2+i_3}\\
&\leq \mathcal{P}(0)\frac{1}{i_2!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2}\sum_{i_3=0}^{\infty}\frac{1}{i_3!}\left(\frac{\lambda t}{\phi-1}\right)^{i_3}\\
&\leq \mathcal{P}(0)\frac{1}{i_2!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2}e^{\frac{\lambda t}{\phi-1}}\\
&\leq \mathcal{C}_1\frac{1}{i_2!}\left(\frac{\lambda t}{\phi-1}\right)^{i_2}
\end{aligned}
$$

where $\mathcal{C}_1 = \mathcal{P}(0)\,e^{\frac{\lambda t}{\phi-1}}$ depends on the general parameters of CBF but is constant with respect to i_2. ▮

We assume now that $\lambda t \leq \phi - 1$ (which is the case in the settings we adopt for counting Bloom filters (see section 8.4.3)). Hence, the constant $\mathcal{C}_1$ is very close to 1:

$$\begin{aligned} \mathcal{C}_1 &= \mathcal{P}(0) e^{\frac{\lambda t}{\phi-1}} \\ &= (1 - \frac{1}{\phi})^{\lambda t} e^{\frac{\lambda t}{\phi-1}} \\ \mathcal{C}_1 &\approx e^{-\frac{\lambda t}{\phi}} e^{\frac{\lambda t}{\phi-1}} \\ &\approx e^{\frac{\lambda t}{\phi(\phi-1)}} \\ \mathcal{C}_1 &\leq e^{\frac{1}{\phi}}. \end{aligned}$$

The fact that $\lambda t \leq \phi - 1$ also means that the upper bound on $\mathcal{P}'(i_2)$ is the product of a constant close to one, a factor $(\frac{\lambda t}{\phi - 1})^{i_2}$ which is decreasing exponentially and a factor decreasing even more than exponentially $(\frac{1}{i_2!})$.

We now prove a weaker but more practical result, that shows that the probability $\mathcal{P}'(i_2)$ decreases faster than a geometric series of ratio $\mathcal{P}(0)$. More precisely we prove the following Proposition.

Proposition 8.4.4 *Assume that* $\lambda t \leq \phi - 1$ *and that* $\phi \leq \frac{1}{1 - e^{\frac{1}{\lambda t} \ln \frac{2}{3}}} \approx \frac{\lambda t}{\ln \frac{3}{2}}$.

Then, for $1 \leq i_2 \leq \lambda t$,

$$\mathcal{P}'(i_2) \leq (\mathcal{P}'(1))^{i_2}.$$

Proof We prove the lemma by induction on $1 \leq i_2 \leq \lambda t$. We denote by $\mathcal{S}(i_2)$ the following statement: For $1 \leq i_3 \leq i_2$, $\mathcal{P}'(i_3) \leq (\mathcal{P}'(1))^{i_3}$.

Basis: $\mathcal{S}(1)$ amounts to $\mathcal{P}'(1) \leq (\mathcal{P}'(1))^1$ which is obviously true.

We also prove the second step of the induction separately.

$\mathcal{S}(2)$ amounts to $\mathcal{P}'(2) \leq (\mathcal{P}'(1))^2$.

We thus examine the difference $\mathcal{P}'(2) - (\mathcal{P}'(1))^2$.

$$
\begin{aligned}
\mathcal{P}'(2) - (\mathcal{P}'(1))^2 &= 1 - \mathcal{P}(0) - \mathcal{P}(1) - (1 - \mathcal{P}(0))^2 \\
&= \mathcal{P}(0) - \mathcal{P}(1) - \mathcal{P}(0)^2 \\
&= \left(1 - \frac{1}{\phi}\right)^{\lambda t} - \lambda t \left(1 - \frac{1}{\phi}\right)^{\lambda t - 1} \frac{1}{\phi} - \left(1 - \frac{1}{\phi}\right)^{2\lambda t} \\
&= \left(1 - \frac{1}{\phi}\right)^{\lambda t - 1} \left(1 - \frac{1}{\phi} - \frac{\lambda t}{\phi} - \left(1 - \frac{1}{\phi}\right)^{\lambda t + 1}\right) \\
&= \left(1 - \frac{1}{\phi}\right)^{\lambda t - 1} \left(1 - \frac{\lambda t + 1}{\phi} - \sum_{i_3=0}^{\lambda t+1} \binom{\lambda t + 1}{i_3} \left(\frac{-1}{\phi}\right)^{i_3}\right) \\
&= -\left(1 - \frac{1}{\phi}\right)^{\lambda t - 1} \sum_{i_3=2}^{\lambda t+1} \binom{\lambda t + 1}{i_3} \left(\frac{-1}{\phi}\right)^{i_3} \\
&= -\left(1 - \frac{1}{\phi}\right)^{\lambda t - 1} \sum_{i_3=2}^{\lambda t+1} u_{i_3}
\end{aligned}
$$

with $u_{i_3} = \binom{\lambda t+1}{i_3}\left(\frac{-1}{\phi}\right)^{i_3}$ for $2 \leq i_3 \leq \lambda t + 1$.

$\{u_{i_3}\}_{2 \leq i_3 \leq \lambda t+1}$ is an alternate series because, for $2 \leq i_3 \leq \lambda t$:

- $u_{i_3} u_{i_3+1} \leq 0$
- $-\frac{u_{i_3+1}}{u_{i_3}} \leq 1$:

$$
\begin{aligned}
-\frac{u_{i_3+1}}{u_{i_3}} &= \frac{\binom{\lambda t+1}{i_3+1}\left(\frac{-1}{\phi}\right)^{i_3} + 1}{\binom{\lambda t+1}{i_3}\left(\frac{-1}{\phi}\right)^{i_3}} \\
&= \frac{\lambda t + 1 - i_3}{\phi(i_3 + 1)}
\end{aligned}
$$

Since $\lambda t + 1 - i_3 \leq \lambda t + 1$ and $\lambda t + 1 \leq \phi$ by hypothesis it implies that:

$$
-\frac{u_{i_3+1}}{u_{i_3}} \leq \frac{1}{u_{i_3} + 1} \leq 1.
$$

According to the alternate series criterion (also called as Leibniz criterion) [Gou08], $\sum_{i_3=2}^{\lambda t+1} u_{i_3}$ is of the sign of the first term which is positive. Finally,

$$
\begin{aligned}
\mathcal{P}'(2) - (\mathcal{P}'(1))^2 &= -\underbrace{\left(1 - \frac{1}{\phi}\right)^{\lambda t - 1}}_{\geq 0} \underbrace{\sum_{i_3=2}^{\lambda t+1} u_{i_3}}_{\geq 0} \\
\mathcal{P}'(2) - (\mathcal{P}'(1))^2 &\leq 0
\end{aligned}
$$

This implies that $\mathcal{P}'(2) \leq (\mathcal{P}'(1))^2$ and proves $\mathcal{S}(2)$.

Inductive step: We consider $3 \leq i_2 \leq \lambda t$. We assume that $\mathcal{S}(i_2 - 1)$ holds then we prove that $\mathcal{S}(i_2)$ holds as well.

Assuming that $\mathcal{S}(i_2 - 1)$ holds means in particular that

$$\mathcal{P}'(i_2 - 1) \leq (\mathcal{P}'(1))^{i_2 - 1}.$$

Therefore if we prove that:

$$\mathcal{P}'(i_2) \leq \mathcal{P}'(i_2 - 1)\mathcal{P}'(1)$$

then it implies that $\mathcal{P}'(i_2) \leq (\mathcal{P}'(1))^{i_2}$.

We therefore examine the difference

$$\delta = \mathcal{P}'(i_2) - \mathcal{P}'(i_2 - 1)\mathcal{P}'(1) \tag{8.1}$$

$$\begin{aligned}
\delta &= \mathcal{P}'(i_2) - \mathcal{P}'(i_2 - 1)\mathcal{P}'(1) \\
&= \mathcal{P}'(i_2) - \mathcal{P}'(i_2 - 1)(1 - \mathcal{P}(0)) \\
&= (\mathcal{P}'(i_2 - 1) - \mathcal{P}(i_2 - 1)) - \mathcal{P}'(i_2 - 1) + \mathcal{P}'(i_2 - 1)\mathcal{P}(0) \\
&= -\mathcal{P}(i_2 - 1) + \mathcal{P}'(i_2 - 1)\mathcal{P}(0) \\
&= -\mathcal{P}(0)\binom{\lambda t}{i_2 - 1}\frac{1}{(\phi - 1)^{i_2 - 1}} + \mathcal{P}'(i_2 - 1)\mathcal{P}(0) \\
&= \mathcal{P}(0)\left(-\binom{\lambda t}{i_2 - 1}\frac{1}{(\phi - 1)^{i_2 - 1}} + \mathcal{P}(0)\sum_{i_3 = i_2 - 1}^{\lambda t}\binom{\lambda t}{i_3}\frac{1}{(\phi - 1)^{i_3}}\right) \qquad (8.2)
\end{aligned}$$

At this point it would be interesting to factorize $\binom{\lambda t}{i_2 - 1}\frac{1}{(\phi-1)^{i_2-1}}$. To this extent we need to write the elements under the sum differently and we observe that:

$$\binom{\lambda t}{i_2} = \frac{(\lambda t)!}{i_2!(\lambda t - i_2)!} = \frac{(\lambda t)!(\lambda t - i_2 + 1)}{(i_2 - 1)! i_2 (\lambda t - i_2 + 1)!} = \binom{\lambda t}{i_2 - 1}\frac{\lambda t - i_2 + 1}{i_2}.$$

We can repeat the process and obtain more generally that, for $i_2 \leq i_3 \leq \lambda t$:

$$\binom{\lambda t}{i_3} = \binom{\lambda t}{i_2 - 1}\prod_{i_4 = i_2}^{i_3}\frac{\lambda t + 1 - i_4}{i_4}.$$

We can now rewrite δ (8.2) as follows:

$$\begin{aligned}
\delta &= \mathcal{P}(0)\left(-\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}} + \mathcal{P}(0)\sum_{i_3=i_2-1}^{\lambda t}\binom{\lambda t}{i_3}\frac{1}{(\phi-1)^{i_3}}\right)\\
&= \mathcal{P}(0)\left(-\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}} + \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\right)\\
&\quad + \mathcal{P}(0)\mathcal{P}(0)\sum_{i_3=i_2}^{\lambda t}\binom{\lambda t}{i_2-1}\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4}\frac{1}{(\phi-1)^{i_2-1}}\frac{1}{(\phi-1)^{i_3-i_2+1}}\\
&= \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4}\frac{1}{(\phi-1)^{i_3-i_2+1}}\right)\right)\\
&= \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\underbrace{\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4}}_{i_3-i_2+1\text{ elements}}\frac{1}{(\lambda t)^{i_3-i_2+1}}\frac{(\lambda t)^{i_3-i_2+1}}{(\phi-1)^{i_3-i_2+1}}\right)\right)\\
&= \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4\lambda t}\left(\frac{\lambda t}{\phi-1}\right)^{i_3-i_2+1}\right)\right) \qquad (8.3)
\end{aligned}$$

We are only interested in the sign of δ and not in its actual value. The factor

$$\mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}} > 0$$

which is non negative does not impact the sign of δ and can thus be ignored.

Concerning the other factor we need to show that it is negative. To this extent we need to give an upper bound on the sum.

We observe that, for the terms under the product:

- $\lambda t + 1 - i_4 \leq \lambda t$,
- $i_4 \geq i_2$, hence $\frac{1}{i_4} \leq \frac{1}{i_2}$.

This leads us to the following upper bound on the product:

$$\begin{aligned}
\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4\lambda t} &\leq \underbrace{\prod_{i_4=i_2}^{i_3}\frac{\lambda t}{i_2\lambda t}}_{i_3-i_2+1\text{ terms}}\\
&\leq \frac{1}{{i_2}^{i_3-i_2+1}}.
\end{aligned}$$

By inserting this upper bound in δ (8.3):

$$\begin{aligned}
\delta &= \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\prod_{i_4=i_2}^{i_3}\frac{\lambda t+1-i_4}{i_4\lambda t}\left(\frac{\lambda t}{\phi-1}\right)^{i_3-i_2+1}\right)\right)\\
\delta &\leq \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\frac{1}{{i_2}^{i_3-i_2+1}}\left(\frac{\lambda t}{\phi-1}\right)^{i_3-i_2+1}\right)\right)\\
&\leq \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=i_2}^{\lambda t}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3-i_2+1}\right)\right)\\
&\leq \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\left(1+\sum_{i_3=1}^{\lambda t-i_2+1}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3}\right)\right)\\
&\leq \mathcal{P}(0)\binom{\lambda t}{i_2-1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\sum_{i_3=0}^{\lambda t-i_2+1}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3}\right) \qquad (8.4)
\end{aligned}$$

The sum on the right is a sum of a geometric series with ratio $\frac{\lambda t}{i_2(\phi-1)} < 1$ since $\lambda t \leq \phi-1$ by hypothesis. The sum can thus be computed and bounded as follows:

$$\begin{aligned}
\sum_{i_3=0}^{\lambda t-i_2+1}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3} &= \frac{1-\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{\lambda t-i_2}}{1-\frac{\lambda t}{i_2(\phi-1)}}\\
\sum_{i_3=0}^{\lambda t-i_2+1}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3} &\leq \frac{1}{1-\frac{\lambda t}{i_2(\phi-1)}}\\
&\leq \frac{i_2(\phi-1)}{i_2(\phi-1)-\lambda t} \qquad (8.5)
\end{aligned}$$

Furthermore, since $\lambda t \leq \phi - 1$:

$$\begin{aligned}
-\lambda t &\geq -\phi-1\\
i_2(\phi-1)-\lambda t &\geq i_2(\phi-1)-\phi-1\\
\frac{1}{i_2(\phi-1)-\lambda t} &\leq \frac{1}{(i_2-1)(\phi-1)}\\
\frac{i_2(\phi-1)}{i_2(\phi-1)-\lambda t} &\leq \frac{i_2}{i_2-1} \qquad (8.6)
\end{aligned}$$

And since we assumed that $i_2 \geq 3$,

$$\frac{i_2}{i_2-1} \leq \frac{3}{2} \qquad (8.7)$$

By combining 8.5, 8.6, and 8.7, and injecting the result in 8.4, we obtain:

$$\begin{aligned}
\delta &\leq \mathcal{P}(0)\binom{\lambda t}{i_2 - 1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\mathcal{P}(0)\sum_{i_3=0}^{\lambda t - i_2 + 1}\left(\frac{\lambda t}{i_2(\phi-1)}\right)^{i_3}\right) \\
\delta &\leq \mathcal{P}(0)\binom{\lambda t}{i_2 - 1}\frac{1}{(\phi-1)^{i_2-1}}\left(-1+\frac{3}{2}\mathcal{P}(0)\right) \qquad (8.8)
\end{aligned}$$

Claim: If $\phi \leq \frac{1}{1-e^{\frac{1}{\lambda t}\ln\frac{2}{3}}} \approx \frac{\lambda t}{\ln\frac{3}{2}}$, then $\mathcal{P}(0) \leq \frac{2}{3}$. The proof of the claim is given in the Lemma 8.4.5.

By using the claim in the last expression of δ (8.8), we obtain $\delta \leq 0$.

By definition of δ (8.1), this implies that $\mathcal{P}'(i_2) \leq \mathcal{P}'(i_2 - 1)\mathcal{P}'(1)$, which means that $\mathcal{S}(i_2)$ holds and thus by induction:

$$\forall 1 \leq i_2 \leq \lambda t, \mathcal{P}'(i_2) \leq (\mathcal{P}'(1))^{i_2}.$$

∎

Lemma 8.4.5 *If* $\phi \leq \frac{1}{1-e^{\frac{1}{\lambda t}\ln\frac{2}{3}}} \approx \frac{\lambda t}{\ln\frac{3}{2}}$, *then* $\mathcal{P}(0) \leq \frac{2}{3}$.

Proof

$$\begin{aligned}
& \mathcal{P}(0) && \leq \frac{2}{3} \\
\Leftrightarrow\quad & \left(1-\tfrac{1}{\phi}\right)^{\lambda t} && \leq \frac{2}{3} \\
\Leftrightarrow\quad & e^{\lambda t \ln\left(1-\frac{1}{\phi}\right)} && \leq \frac{2}{3} \\
\Leftrightarrow\quad & \lambda t \ln\left(1-\tfrac{1}{\phi}\right) && \leq \ln\left(\frac{2}{3}\right) \\
\Leftrightarrow\quad & \ln\left(1-\tfrac{1}{\phi}\right) && \leq \frac{1}{\lambda t}\ln\left(\frac{2}{3}\right) \qquad (8.9)
\end{aligned}$$

At this stage, we can obtain an exact result or an approximation. We start by the exact result:

$$\begin{aligned}
\Leftrightarrow\quad & 1-\tfrac{1}{\phi} && \leq e^{\frac{1}{\lambda t}\ln\left(\frac{2}{3}\right)} \\
\Leftrightarrow\quad & \tfrac{1}{\phi} && \geq 1 - e^{\frac{1}{\lambda t}\ln\left(\frac{2}{3}\right)} \\
\Leftrightarrow\quad & \phi && \leq \frac{1}{1-e^{\frac{1}{\lambda t}\ln\left(\frac{2}{3}\right)}}
\end{aligned}$$

Assuming that $\phi >> 1$, we can approximate $\ln\left(1-\frac{1}{\phi}\right)$ by $-\frac{1}{\phi}$. By injecting this approximation in 8.9 we derive:

$$-\frac{1}{\phi} \leq \frac{1}{\lambda t}\ln\left(\frac{2}{3}\right)$$

$$\phi \leq \frac{\lambda t}{\ln \frac{3}{2}}.$$

$\blacksquare$

In the sequel of the chapter we use only the approximated result as it is more practical to evaluate.

We proved the statement $\mathcal{S}(2)$ separately to obtain the $\ln \frac{3}{2}$ in the condition on ϕ, otherwise we would have had a $\ln 2$ which is not practical for our setting. This condition $\phi \leq \frac{1}{1-e^{\frac{1}{\lambda t}\ln\frac{2}{3}}} \approx \frac{\lambda t}{\ln \frac{3}{2}}$ which is required in our proof, but we conjecture that the Proposition 8.4.4 holds without this condition as well.

This concludes the analysis of the properties of CBF, and we use the results of this section to evaluate the probability of success of an attacker.

8.4.2.3 Probability of Success of an Attacker

At the end of section 8.4.2.1, we mentioned the three conditions that an adversary has to fill in order to launch a successful attack. The hardest condition for the adversary is to output an array simulating a counting Bloom filter which is smaller than the matching reference. Therefore we first compute the probability of success for this condition.

We assume that CBF_1 is a counting Bloom filter of size ϕ, which contains λ (unknown) elements. Each elements is inserted in CBF by using the `Insert` primitive. This primitive uses t hash functions denoted h_1,...,h_t. The output of these hash functions is assumed to be uniformly distributed over $[0, \phi - 1]$ (the set of indexes of CBF_1).

We also assume that $\lambda t + 1 \leq \phi \leq \frac{\lambda t}{\ln \frac{3}{2}}$, to be in the conditions of the propositions of the previous section.

Proposition 8.4.6 *Let CBF_2 be an array of size ϕ. The probability $\mathcal{P}[CBF_2 \prec CBF_1]$ that CBF_2 is smaller than CBF_1 is:*

$$\begin{aligned}\mathcal{P}[CBF_2 \prec CBF_1] &= \prod_{i_1=0}^{\phi-1} \mathcal{P}'(CBF_2[i_1]) \\ \mathcal{P}[CBF_2 \prec CBF_1] &\leq \prod_{i_1=0}^{\phi-1} \mathcal{P}'(1)^{CBF_2[i_1]}\end{aligned}$$

Proof By definition, $CBF_2 \prec CBF_1$ amounts to $\forall 0 \leq i_1 \leq \phi - 1,\ \ CBF_2[i_1] \leq CBF_1[i_1]$.

For each position $0 \leq j \leq \phi - 1$, the probability that $CBF_2[i_1] \leq CBF_1[i_1]$ is exactly the probability $\mathcal{P}'(CBF_2[i_1])$ that we defined earlier. Since the probabilities are independent at each position, the total probability is the product of each elementary probability. The inequality is a direct application of lemma 8.4.4, hence the proposition. $\blacksquare$

Proposition 8.4.7 *Let CBF_2 be an array of size ϕ and of weight w_{CBF_2}.*

The probability $\mathcal{P}[CBF_2 \prec CBF_1]$ that CBF_2 is smaller than CBF_1 is bounded by $\left(1 - (1 - \frac{1}{\phi}\right)^{\lambda t})^{w_{CBF_2}} \approx \left(1 - e^{-\frac{\lambda t}{\phi}}\right)^{w_{CBF_2}}$.

Proof According to Proposition 8.4.6:

$$\mathcal{P}[CBF_2 \prec CBF_1] \leq \prod_{i_1=0}^{\phi-1} \mathcal{P}'(1)^{CBF_2[i_1]}.$$

By definition, $\sum_{i_1=0}^{\phi-1} CBF_2[i_1] = w_{CBF_2}$, thus

$$\begin{aligned}\mathcal{P}[CBF_2 \prec CBF_1] &\leq \prod_{i_1=0}^{\phi-1} \mathcal{P}'(1)^{CBF_2[i_1]} \\ &\leq \mathcal{P}'(1)^{\sum_{i_1=0}^{\phi-1} CBF_2[i_1]} \\ &\leq \mathcal{P}'(1)^{w_{CBF_2}}.\end{aligned}$$

Moreover, $\mathcal{P}'(1) = 1 - \mathcal{P}(0)$, and $\mathcal{P}(0) = (1 - \frac{1}{\phi})^{\lambda t}$. Let us look at the Taylor series of order 2 of $(1 - \frac{1}{\phi})^{\lambda t}$ and of $e^{-\frac{\lambda t}{\phi}}$:

$$\left(1 - \frac{1}{\phi}\right)^{\lambda t} = 1 - \frac{\lambda t}{\phi} + \frac{\lambda t(\lambda t - 1)}{2\phi^2} + \circ\left(\frac{1}{\phi^2}\right)$$

$$e^{-\frac{\lambda t}{\phi}} = 1 - \frac{\lambda t}{\phi} + \frac{(\lambda t)^2}{2\phi^2} + \circ\left(\frac{1}{\phi^2}\right)$$

This development indicates that $e^{-\frac{\lambda t}{\phi}}$ is a good approximation of $\left(1 - \frac{1}{\phi}\right)^{\lambda t}$ (and is more practical to evaluate).

We therefore conclude that $\mathcal{P}[CBF_2 \prec CBF_1] \leq \left(1 - (1 - \frac{1}{\phi}\right)^{\lambda t})^{w_{CBF_2}} \approx \left(1 - e^{-\frac{\lambda t}{\phi}}\right)^{w_{CBF_2}}$.

▮

We now apply these results to the case of an attacker as defined in section 8.4.2.1. In the sequel of the chapter we use only the approximated expression as it is easier to evaluate.

We have a source node N_S which constructed the matching reference $CBF_S(M')$ as a counting Bloom filter of size ϕ. N_S inserted $|L_M|$ elements in $CBF_S(M')$ using t hash functions. We assume that N_S chose the parameters of $CBF_S(M')$ such that $|L_M|t + 1 \leq \phi \leq \frac{|L_M|t}{\ln \frac{3}{2}}$.

$CBF_S(M')$ is known by an intermediate node N_i and is used to compute a provably correct matching ratio for its neighbors. One of the neighbors N_k wants to cheat and claim a higher matching ratio than the legitimate one. N_k has to output an array of integers $CBF_{k_{claim}}(M')$ of size ϕ, verifying the three following properties:

1. $CBF_{k_{claim}}(M') \prec CBF_S(M')$,
2. $w_{CBF_{k_{claim}}(M')}$ is a multiple of t,

3. $w_{CBF_{k_{claim}}(M')} > w_{CBF_{k_{legit}}(M')}$.

N_k has a matching set $L_{M,k}$, which enables him to construct a legitimate counting Bloom filter $CBF_{k_{legit}}(M')$ leading to a legitimate matching ratio $p_{k_{legit}}(M') = \frac{|L_{M,k}|}{|L_M|}$.

All these information are legitimate for N_k and do not offer a challenge to N_k. The challenge for N_k is to claim a higher matching ratio; we thus remove the components known by N_k and consider

$$CBF_{S,k}(M') = CBF_S(M') - CBF_{k_{legit}}(M')$$

as the challenging reference (the subtraction of two counting Bloom filters corresponds to element by element subtraction: for $0 \leq i_1 \leq \phi - 1$, $CBF_{S,k}(M')[i_1] = CBF_S(M')[i_1] - CBF_{k_{legit}}(M')[i_1]$).

Indeed the challenge for N_k is really to produce an array $CBF_{k_{mal}}(M')$ of size ϕ which satisfies the following conditions:

- $CBF_{k_{mal}}(M')$ is smaller than $CBF_{S,k}(M')$: $CBF_{k_{mal}}(M') \prec CBF_{S,k}(M')$,
- the weight $w_{CBF_{k_{mal}}(M')}$ of $CBF_{k_{mal}}(M')$ is a non-zero multiple of t.

Then the total counting Bloom filter $CBF_{k_{claim}}(M') = CBF_{k_{legit}}(M') + CBF_{k_{mal}}(M')$ would lead to an increase of the matching ratio of N_k from $p_{k_{legit}}(M')$ to

$$p_{k_{claim}}(M') = p_{k_{legit}}(M') + p_{k_{mal}}(M') = \frac{|L_{M,k}|}{|L_M|} + \frac{w_{CBF_{k_{mal}}(M')}}{t|L_M|}.$$

Since N_k does not know $CBF_{S,k}(M')$, N_k can only make guesses and probabilistically model $CBF_{S,k}(M')$. What N_k knows about $CBF_{S,k}(M')$ is its weight:

$$\begin{aligned} w_{S,k}(M') = w_{CBF_{S,j}(M')} &= w_{CBF_S(M')} - w_{CBF_{k_{legit}}(M')} \\ &= (|L_M| - |L_{M,k}|)t \end{aligned}$$

and the general parameters which are the number t of hash function used and the size ϕ of the counting Bloom filter.

Theorem 8.4.8 *Let M' be a message with a header containing $|L_M|$ attributes.*

Assume the counting Bloom filter has a size ϕ and use t hash functions.

The probability of success of an adversary $\mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')]$ in generating an array $CBF_{k_{mal}}(M')$ which is accepted by N_i and results in an increase of the matching ratio by $p_{k_{mal}}(M')$ is upperly bounded by:

$$\begin{aligned} \mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')] &\leq \left(1 - e^{-\frac{(1-p_{k_{legit}}(M')|L_M|t}{\phi}}\right)^{w_{CBF_{k_{mal}}(M')}} \\ &\leq \left(1 - e^{-\frac{(1-p_{k_{legit}}(M')|L_M|t}{\phi}}\right)^{t} \\ &\leq \left(1 - e^{-\frac{|L_M|t}{\phi}}\right)^{t} \end{aligned}$$

Proof A direct application of proposition 8.4.7 leads to the fact that the probability that an array $CBF_{k_{mal}}(M')$ of size ϕ and of weight $w_{CBF_{j_{mal}}(M')}$ is smaller than $CBF_{S,k}(M')$ is bounded by:

$$\left(1 - e^{-\frac{w_{S,k}(M')}{\phi}}\right)^{w_{CBF_{k_{mal}}(M')}}.$$

$CBF_{S,k}(M') = CBF_S(M') - CBF_{k_{legit}}(M')$, therefore

$$w_{S,k}(M') = w_{CBF_S(M')}(1 - p_{k_{legit}}(M')) = |L_M|t(1 - p_{k_{legit}}(M')),$$

which leads to the first inequality.

In order for $CBF_{k_{mal}}(M')$ to be accepted, it should not only verify $CBF_{k_{mal}}(M') \prec CBF_{S,k}(M')$, but also $w_{CBF_{k_{mal}}(M')}$ should be a non-zero multiple of t. Hence it is only possible to increment the matching ratio by multiples of $\frac{1}{|L_M|}$, but intermediate values would be rejected.

Since the probability of success of the adversary decrements exponentially with the weight of the malicious array, the highest probability of success for the adversary corresponds to the smallest acceptable weight, which is t, which gives the second inequality.

Furthermore the probability of success depends on the legitimate matching ratio: the higher the legitimate ratio, the harder it is for the adversary to succeed. Therefore the probability of success of the adversary in the most favorable case, which is when no attributes are shared and the adversary tries to cheat by the smallest amount possible, is bounded by $(1 - e^{-\frac{|L_M|t}{\phi}})^t$, which is the third inequality. ■

Note that the first formula is meaningful only if $p_{k_{mal}}(M')$ is a multiple of $\frac{1}{|L_M|}$ to satisfy the second of the aforementioned conditions (otherwise the claimed counting Bloom filter would be rejected).

The formula of $\mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')]$ shows that the probability of success of an adversary decreases exponentially with the malicious ratio increase and, decreases also depending on the value of the legitimate matching ratio.

To illustrate these facts, let us take an example of a message with 10 attributes. We assume that there are two adversaries, the first one with matching ratio 0 and the second one with matching ratio 50%. In that case, it is easier for the first adversary to succeed in raising its matching ratio from 0 to 10% than for the second to raise it from 50% to 60%. It is also twice more difficult for the adversaries to raise the ratio from 0 to 20% (respectively from 50% to 70%) than from 0 to 10% (respectively from 50% to 60%).

The security of the assurance scheme hence depends on the general parameters of the counting Bloom filter. We therefore focus in the next section on how to set these parameters in an optimal way.

8.4.3 Tuning the global Parameters

The general parameters of counting Bloom filters are the size ϕ, the number of inserted elements $|L_M|$ and the number of hash functions t. The goal of this section is to explain how to choose these parameters to minimize the probability of success of an attacker.

First of all, we assume that the maximum number of attributes that can be inserted in the header of a message is bounded and known in advance, we designate it as n_{max}. Then for all messages M', the probability of success of the adversary is bounded by

$$\mathcal{P}_{adv} \leq \left(1 - e^{-\frac{n_{max}t}{\phi}}\right)^t.$$

If we fix ϕ, then the function $p_{max} : t \mapsto \left(1 - e^{-\frac{n_{max}t}{\phi}}\right)^t$ is $\mathcal{C}^\infty$ on $[1, +\infty[$.

Proposition 8.4.9 *$p_{max} : t \mapsto \left(1 - e^{-\frac{n_{max}t}{\phi}}\right)^t$ reaches its minimum in $t_0 = \frac{\phi}{n_{max}} ln(2)$ and $p_{max}(t_0) = 2^{-t_0}$.*

Proof p_{max} is $\mathcal{C}^\infty$ on $[1, +\infty[$, thus it is possible to compute the derivative of p_{max}:

$$p'_{max}(t) = \left(\ln\left(1 - e^{-\frac{n_{max}t}{\phi}}\right) + \frac{n_{max}t}{\phi} \frac{e^{-\frac{n_{max}t}{\phi}}}{1 - e^{-\frac{n_{max}t}{\phi}}}\right)\left(1 - e^{-\frac{n_{max}t}{\phi}}\right)^t.$$

We need to find the zero t_0 of p'_{max} in order to determine the minimum of p_{max}. p'_{max} is a product of two terms, the one on the right hand side is always positive and non-zero, therefore we focus on the left hand side one:

$$\begin{aligned} \ln\left(1 - e^{-\frac{n_{max}t_0}{\phi}}\right) + \frac{n_{max}t_0}{\phi} \frac{e^{-\frac{n_{max}t_0}{\phi}}}{1 - e^{-\frac{n_{max}t_0}{\phi}}} &= 0 \\ \left(1 - e^{-\frac{n_{max}t_0}{\phi}}\right)\ln\left(1 - e^{-\frac{n_{max}t_0}{\phi}}\right) + \frac{n_{max}t_0}{\phi} e^{-\frac{n_{max}t_0}{\phi}} &= 0 \end{aligned}$$

We proceed to a variable change: we define $y = 1 - e^{-\frac{n_{max}t_0}{\phi}}$. Then $\frac{n_{max}t_0}{\phi} = -\ln(1-y)$. Therefore:

$$\begin{array}{lcl} & p'_{max}(t_0) & = 0 \\ \Leftrightarrow & \left(1 - e^{-\frac{n_{max}t_0}{\phi}}\right)\ln\left(1 - e^{-\frac{n_{max}t_0}{\phi}}\right) + \frac{n_{max}t_0}{\phi} e^{-\frac{n_{max}t_0}{\phi}} & = 0 \\ \Leftrightarrow & y\ln(y) - \ln(1-y)e^{\ln(1-y)} & = 0 \\ \Leftrightarrow & y\ln(y) - (1-y)\ln(1-y) & = 0 \\ \Leftrightarrow & y\ln(y) & = (1-y)\ln(1-y) \end{array}$$

Therefore $y = 1 - y$, which leads to $y = \frac{1}{2}$ and finally $\frac{n_{max}t_0}{\phi} = -\ln(1 - \frac{1}{2}) = \ln(2)$. For a fixed ϕ, the value of t that minimizes p_{max} is therefore $t_0 = \frac{\phi}{n_{max}} \ln(2)$, and

$$p_{max}(t_0) = (1 - e^{-\frac{n_{max}}{\phi}\frac{\phi}{n_{max}}\ln(2)})^{t_0} = (1 - \frac{1}{2})^{t_0} = 2^{-t_0}.$$

∎

This result shows that there is a trade-off between security and performance: for a fixed n_{max} increasing t and ϕ exponentially increases the security and linearly the size of the bloom filter.

The strategy to set the parameters is thus the following:

1. Set the maximum number of elements that might be inserted in the counting Bloom filter n_{max},
2. Choose a security parameter t such that the probability $\mathcal{P}_{adv}$ of success of the attacker is bounded by 2^{-t},
3. Set the size ϕ of the counting Bloom filter as $\phi = \left\lceil \frac{n_{max}t}{\ln(2)} \right\rceil$.

We observe that this strategy fixes ϕ as required by the propositions: $n_{max}t + 1 \leq \phi \leq \frac{n_{max}t}{\ln(\frac{3}{2})}$.

This strategy prioritizes security over performance: it defines the desired security level ($\mathcal{P}_{adv} \leq 2^{-t}$) and then sets the minimal size ϕ to achieve this security level.

Note that t does not need to be very large. While typical security margins for encryption systems are of the order of 2^{-80}, one has to consider practically the needs in this scheme: an attacker N_k in this scheme does not decipher a secret message, nor does it reveal a private key, it only manages to lure N_i into believing that its matching ratio $p_k(M')$ is higher than what it actually is.

Furthermore, contrary to encryption systems where an attacker could have access to several plaintext/ciphertext pairs for a given key, the attacker cannot run several trials on the same counting Bloom Filter $CBF_S(M')$, because even if N_S wants to impose the same set of conditions twice, the resulting counting Bloom filter will change because of the random seed of the PEKS scheme, therefore running several instances of the protocol does not lead to a further advantage for the attacker.

Finally, one should observe that the probability computed is an upper bound and is obtained with very restrictive conditions:

- $L_M = n_{max}$, which means that N_S uses n_{max} attributes in the header of the message,
- N_k has a legitimate matching ratio of 0 ($p_{k_{legit}}(M') = 0$).

With these conditions, N_k has 2^{-t} probability of succeeding in making N_i believe that its matching ratio is $1/n_{max}$ instead of 0. In many cases, this would not be of any use to the attacker, because the attacker needs to claim the highest matching ratio among the neighbors of N_i in order to take advantage of its attack. The attacker does not even know the matching ratio of the other neighbors, so the only way for N_k to be sure to benefit from its attack is to claim a matching ratio of 1, and the probability of N_k succeeding falls down to $2^{-tn_{max}}$.

For all these reasons, it is sufficient to set a small value for t, for example $t = 10$ which already results in a probability of success for the easiest (and most probably useless) attack inferior to 1 over 1000. This discussion simply shows that the computation assurance scheme is dissuasive and thus enforces trust in the correctness of claimed matching ratios. We discuss these parameters more extensively in section 9.2.

8.5 Conclusion

In this chapter we presented a solution to enforce computation assurance of the matching ratio: we showed that, with our solution, an adversary claiming a matching ratio higher than its actual value has a success rate which is exponentially decreasing with the difference between the claimed and legitimate matching ratio. The solution is also efficient from a communication overhead perspective and is enhances privacy protection, as nodes cannot even discover the matching attribute names of their neighbors.

CBF	counting Bloom filter
ϕ	number of positions (buckets) in CBF
$CBF[i]$	value of CBF at the position i
t	number of hash functions used in CBF
$h_1,\ldots,h_t$	hash functions used in CBF
w_{CBF}	weight of CBF

$CBF_S(M')$	matching reference generated by N_S for message M'
$x_{M',j}$	elements inserted in $CBF_S(M')$
$\|L_M\|$	number of element of set L_M
$CBF_k(M')$	counting Bloom filter generated by N_k for message M'
$p_k(M')$	matching ratio of N_k for message M'

CBF_1	counting Bloom filter containing λ elements
$\mathcal{P}[X]$	probability of event X
$\mathcal{P}_{i_1}(i_2)$	probability that $CBF_1[i_1] = i_2$
$\mathcal{P}(i_2)$	probability that a bucket in CBF_1 has value i_2
$\mathcal{P}'_{i_1}(i_2)$	probability that $CBF_1[i_1] \geq i_2$
$\mathcal{P}'(i_2)$	probability that a bucket in CBF_1 has value greater than i_2

CBF_2	array of size ϕ
$CBF_{S,k}(M')$	challenging reference from the persepctive of N_k
$w_{CBF_{S,k}(M')}$	weight of $CBF_{S,k}(M')$
$CBF_{k_{legit}}(M')$	legitimate counting Bloom filter of N_k for message M'
$CBF_{k_{claim}}(M')$	claimed counting Bloom filter by N_k for message M'
$CBF_{k_{mal}}(M')$	malicious array of N_k for message M'
$w_{CBF_{k_{mal}}(M')}$	weight of $CBF_{k_{mal}}(M')$
$p_{k_{legit}}(M')$	legitimate matching ratio of N_k for M'
$p_{k_{claim}}(M')$	claimed matching ratio of N_k for M'
$p_{k_{mal}}(M')$	malicious increment of the matching ratio of N_k for M'
$\mathcal{P}_{adv}[p_{k_{legit}}(M') \rightarrow p_{k_{legit}}(M') + p_{k_{mal}}(M')]$	probability that N_k succeeds in increasing its matching ratio from $p_{k_{legit}}(M')$ to $p_{k_{legit}}(M') + p_{k_{mal}}(M')$

Table 8.2: Notations used in the description and the proofs of the efficient computation assurance scheme

Chapter 9

Global Evaluation

In this chapter, we evaluate the three solutions for payload confidentiality, user privacy and computation assurance globally.

We first remind the big picture and how the three schemes are combined. The global solution features two phases:

- the setup phase, during which nodes contact a trusted entity to retrieve the keying material as well as the global parameters of the system,
- the runtime phase, during which the communication occurs opportunistically without access to any trusted entity.

During the runtime phase, the communication flow of a message $M = \mathcal{H}(M)||\mathcal{P}(M)$ sent by the source N_S is the following:

- N_S encrypts the payload $\mathcal{P}(M)$ using the ENCRYPT_PAYLOAD primitive with the attributes of the destination as public keys, and encrypts the header $\mathcal{H}(M)$ using the ENCRYPT_HEADER primitive, obtaining an encrypted message $M' = \mathcal{H}(M')||\mathcal{P}(M')$. N_S also generates a counting Bloom filter $CBF_S(M')$ which is used as matching reference. Finally N_S sends the encrypted message M' along with the matching reference $CBF_S(M')$ to its neighbors.
- Whenever an intermediate node N_i receives M' and $CBF_S(M')$, N_i sends $\mathcal{H}(M')$ only to its neighbor N_k. N_k discovers the shared attributes between $\mathcal{H}(M')$ and $Prof(k)$ thanks to the MATCH_HEADER primitive and constructs an associated counting Bloom filter $CBF_k(M')$. N_k sends $CBF_k(M')$ to N_i which compares it with $CBF_S(M')$ and extracts the correct matching ratio $p_k(M')$. N_i then takes a forwarding decision based on $p_k(M')$; if N_i decides to forward the message to N_k then N_i sends M' and $CBF_S(M')$ to N_k.
- The destination N_D is easily identifiable because its matching ratio $p_D(M')$ is 1. When the destination receives M', it decrypts the payload $\mathcal{P}(M')$ thanks to the DECRYPT_PAYLOAD primitive and therefore recovers the original payload $\mathcal{P}(M)$ of the message.

We now evaluate the complete security framework resulting from the combination of the three schemes for data confidentiality, user's privacy and computation reliability.

9.1 Security Evaluation

9.1.1 Key Management

We describe the global key management scheme for the combination of the three solutions.

As mentioned in section 6.5.1, the payload confidentiality solution features a setup phase during which node each node N_i contacts a PKG to retrieve:

- system parameters $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, P_{pub}, H_1, H_2 \rangle$,
- m secrets $A_{priv_{i,j}}$ for $1 \leq j \leq m$.

The PKG keeps a secret key called $master - key$.

In the setup phase of the user privacy solution (section 7.4.1), each node N_i contacts a TTP to retrieve:

- system parameters $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, H_1, H_3 \rangle$ and the public key pk_{TTP} of the TTP,
- m secrets (trapdoors) $T_{i,j}$ for $1 \leq j \leq m$.

The TTP keeps a secret key sk_{TTP}.

By deploying both solutions together, it is possible to pool the efforts of the TTP and PKG. We consider only one trusted entity TTP which generates global system parameters as follows:

1. Run $\mathcal{G}$ on input sp, generate a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a cryptographic bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
2. Choose a generator P of $\mathbb{G}_1$.
3. Choose three cryptographic hash functions:
   - $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$,
   - $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^\nu$ for some $\nu \in \mathbb{Z}^+$,
   - $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^{\log q}$.
4. Pick a random $s \in \mathbb{Z}_q^*$. s is the $master - key$ and sk_{TTP} at the same time. Then sP plays the role of pk_{TTP} and P_{pub} simultaneously.

In total the system parameters are:

$$params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, \nu, P, pk_{TTP}, H_1, H_2, H_3 \rangle \, ,$$

and the TTP stores one master secret sk_{TTP}.

The main advantage in pooling the TTP and PKG is for the nodes N_i. Indeed the nodes need to retrieve m private keys $A_{priv_{i,j}}$ and m trapdoors $T_{i,j}$. By pooling the TTP and PKG we observe that:

$$A_{priv_{i,j}} = \texttt{MIB-Extract}(params, master-key, A_{i,j}) = sk_{TTP}H_1(A_{i,j}),$$

and

$$T_{i,j} = \texttt{SE-Trapdoor}(params, sk_{TTP}, A_{i,j}) = sk_{TTP}H_1(A_{i,j}).$$

Hence by using a unique entity to play the roles of both TTP and PKG we have $T_{i,j} = A_{priv_{i,j}}$, which means that the nodes can proceed with both the payload confidentiality and user privacy solutions with only m secrets.

9.1.2 Operational Security

In the previous sections, we evaluated the security of each building block separately. In particular we showed that:

- the data confidentiality proposal is an end-to-end service which is semantically secure against a chosen plaintext attack (IND-MID-CPA): only the destination can decrypt the encrypted message (section 6.7),
- the user's privacy solution is also semantically secure and enables nodes to discover only shared attributes thanks to the associated trapdoors (section 7.5),
- the computation reliability scheme guarantees that successfully cheating on the matching ratio is improbable (section 8.4.2).

From an operational point of view, the encryption and encoding functions for data confidentiality and user's privacy, ENCRYPT_ PAYLOAD and ENCRYPT_HEADER, use only public functions and public keys that are distributed during the setup phase to all nodes. Any source node N_S can therefore send messages, even before meeting the destinations during the runtime phase. This characteristic is particularly interesting in opportunistic networks, as there is no end-to-end connectivity between source and destination.

Furthermore, these primitives enable for a flexible definition of the destination based on the destination's context instead of the destination identity.

The security of our schemes for context-based forwarding is based on the trusted communities assumption presented in definition 3.2.1. This assumption states that nodes do not harm others belonging to the same community (based on shared attributes), and enabled us to define a privacy solution in the model 3.

The trusted communities assumption makes it also easier to manage colluding attackers. For instance, suppose that an encrypted message is sent to faculty in INRIA. A node N_1 with attributes $(Status, Student)||(workplace, INRIA)$ may collude with another node N_2 with attributes $(Status, Faculty)||(workplace, EURECOM)$ and obtain the key needed to decrypt the message. Similarly, they could decrypt messages sent to students in EURECOM. This attack implies that N_1 and N_2 merge their decryption capabilities and are able to impersonate each other. In particular, it means that N_2 now can discover the attribute $(workplace, INRIA)$ in any message and harm the INRIA community even though N_2 works at EURECOM. Thus, by colluding with N_2, N_1 would violate the trusted communities assumption (and similarly for N_2). Collusion attack is therefore ruled out by the trusted communities assumption.

Moreover, the ENCRYPT_PAYLOAD and ENCRYPT_HEADER primitives also avoid the dictionary attack, because they make use of internal randomization: their output is different at each execution, even if the input does not change. The computation assurance scheme maintains this property as it uses the same randomized values generated in ENCRYPT_HEADER.

Within the trusted community assumption, the proposed framework ensures privacy of the destination and confidentiality of the payload against eavesdroppers but also curious intermediate nodes, while enabling the computation of the probability used in forwarding decisions. The computation assurance component adds resilience to context-based forwarding as it efficiently prevents nodes from claiming malicious matching ratio: this component thus prevents traffic subversion and mitigates the impact of malicious nodes.

Furthermore, the scheme features packets unlinkability and is therefore resilient against traffic analysis from outsiders. Indeed, thanks to the randomness in ENCRYPT_HEADER and ENCRYPT_PAYLOAD, it is hard to link the headers of packets and therefore it is impossible to know if two packets have the same characteristics (in terms of destination or attributes) or not. The computation assurance scheme also prevents inferring such information from the matching ratio as the data contained in the counting Bloom filter does not leak information on the neighbors profile. Furthermore it is hard to detect and analyze communication flows since the network topology is frequently modified due to nodes mobility, therefore the route between a source and destination differs for consecutive packets.

Finally we point at another interesting property of our combined schemes which is a consequence of the privacy preserving mechanism: as opposed to Boneh's identity-based scheme where the identity of the destination is sent in clear and accessible by any malicious node, the encryption keys used to ensure payload confidentiality can only be discovered by nodes which share the corresponding attributes. This property offers an additional security property as an attacker would first need to find which attributes were used to encrypt the message before being able to launch an attack as in the Boneh's attacker model.

The combination of the three mechanisms thus enforces not only privacy and confidentiality but more generally secure forwarding. We now analyze the storage overhead and the performance of the combined schemes.

9.2 Performance Evaluation

9.2.1 Storage

Concerning storage, as mentioned in section 9.1.1, each node has to store m secrets in addition to the public parameters. Each of these secrets is in fact an element of $\mathbb{G}_1$ a group of prime order q. The storage overhead of the secrets is therefore qm bits, which is linear in the number m of attributes.

During the setup phase, each node needs to fetch its m secrets and the public parameters, but the key management overhead is small from the perspective of the TTP. The TTP needs indeed to set up the public parameters only and to store them, together with a key pair (pk_{TTP}/sk_{TTP}), but it does not need to store all the attribute values of each node (contrary to certification authorities e.g.): trapdoors and private keys are generated upon request. The TTP is therefore lightweight and the storage overhead for the TTP is minor.

9.2.2 Communication Overhead

Concerning communication overhead, we only consider the overhead during the runtime phase: we consider that the key management overhead is negligible given that it takes place offline and thus does not compete with the opportunistic communication.

The proposed security solution does not add significant communication overhead: the size of the message header is linear in the number of attributes that it includes, with or without the security solution, but it remains small in comparison with the size of the payload. The security solution only modifies the attribute values through the PEKS function which has a $2q$ bits output, therefore the size of the attribute values increases by a constant factor, while the size of the payload is not significantly modified during the encryption process.

The computation assurance solution yet adds an additional overhead as it requires the exchange of counting Bloom filters. Counting Bloom filters are arrays containing ϕ positions or buckets. The size β of each bucket has to be set large enough to avoid bucket overflows which would break the counting Bloom filter properties. It is also important to choose the smallest acceptable value in order to minimize the communication overhead since each counting Bloom filter has a total size of $\beta\phi$. In [FCAB00], Li et al. show that the probability the value v at any position being larger than $\xi \in \mathbb{Z}^+$ in a counting Bloom filter of length ϕ with n_{max} inserted elements and $t = \frac{\phi}{n_{max}} \ln(2)$ hash functions is asymptotically:

$$\mathcal{P}[max(v) > \xi] \leq \phi \left(e\frac{\ln(2)}{\xi} \right)^{\xi}.$$

Therefore

$$\mathcal{P}[max(v) > 16] \leq 1.37.10^{-15}\phi.$$

If $\phi << 10^{15}$ and if we choose $\beta = 4$ bits for the size of each bucket then the probability of an overflow in a bucket is negligible. And in the unlikely event of an overflow happening, N_S

can simply recompute a new counting Bloom filter $CBF_S(M')$ with the same attributes but different random numbers so this is really not an issue. An intermediate node N_j does not face this issue at all since its counting Bloom filter $CBF_k(M')$ is smaller than $CBF_S(M')$. Therefore the communication overhead incurred by the computation assurance solution is 4ϕ bits.

9.2.3 Computation Overhead

From a computation performance point of view, the groups that are used are usually implemented as group of rational points on elliptic curves. Elliptic curve operations used in all the primitives are of the same order of magnitude as classical asymmetric cryptography, but they are still more expensive than symmetric encryption.

Indeed, the most costly operation is the pairing computation: one pairing computation per encryption of payload or header, and one pairing per `SE-Test` evaluation or payload decryption. The user privacy solution thus incurs a computation cost linear with the number of attributes used in the header of the message, while the payload confidentiality solution incurs a constant cost (independent of the number of attributes). A pairing computation requires around 11 ms on a pentium III 1 GHz according to the benchmarks established by Lynn based on the PBC library [Lyn06]. In comparison, one 1024 bits RSA decryption takes around 13 ms.

This cost is acceptable for small texts (according to Moore's law the computation power of mobile devices should exceed that of a pentium III 1 GHz by the end of 2010), like the values of attributes but it is prohibitive when it comes to encrypting large data, like the payload. To circumvent this obstacle, the sender can define a symmetric data encryption key which can further be encrypted with the encryption mechanism proposed in section 6.5.2. We did not mention this option in the description of the scheme for the sake of clarity, but for practical deployment this option should be implemented.

Concerning the computation assurance solution, the cost of generating the counting Bloom filters amounts to $|L_M|t \leq n_{max}t$ hash computations which is negligible in comparison with pairing computations.

9.2.4 Typical Figures

To illustrate the performance of the global solution more concretely, we provide some typical figures.

First of all, the number m of attributes defining each profile is usually small in experiments, for example in PROPICMAN [NGP07] the simulation is run with six attributes only. The impact of this parameter is not decisive for communication as it only affects storage, so we can imagine a more complex setting with $m = 100$. Yet the number of maximum attributes that can be inserted in the header of a message should be small as it directly leads to an increase in the communication overhead. We therefore assume a maximum number of attributes in the headers of messages to be $n_{max} = 20$.

The level of security in groups over elliptic curves depends on a security parameter called the MOV degree [MVO91]: by carefully choosing the elliptic curve it is possible to adjust the trade-off between key size and computation time, while maintaining a given level of security. In the case of mobile devices, computation resources are more constrained than storage, therefore it is preferable to choose a curve with small MOV degree, e.g. 2. In such settings it is sufficient to have q of 512 bits length to have a security equivalent to 1024 bits RSA. The storage overhead of the secrets is therefore $qm \approx 50$ `Kbits`, which is negligible when compared with the capacity of flash memories for mobile phones.

The size of the payload depends greatly on the application (exchanging a short message or a film lead to quite different results), but we assume an average value of 1 Mbit (a picture or a small audio file).

The size of the header is linear with the number of attributes in the header. If we assume that each attribute value is stored in a 16 character string and that each character's length is 8 bits then the PROPICMAN solution requires 128 bits for each attribute, while our proposal featuring user's privacy requires $2q \approx 1$ `Kbit` (q bits for the random number and q bits for the PEKS value), which is 8 times more.

A first idea to reduce this overhead is to use only one random number for all the attributes in a message instead of one per attribute. This roughly reduces the overhead to a half. This might still be seen as a drawback but we deliberately recommend the use of an elliptic curve with small MOV degree to save computation resources. If the communication overhead is considered as more important, it is possible to use curves with a higher MOV degree of 6: in that case it is possible to consider groups of order three times smaller and the overhead would be reduced to 2.5 times (or even 1.3 times more with just one random number per message) more instead of 8 times more, but this comes at the cost of increased computation requirements. Anyway, by keeping our conservative approach of 512 bits prime number q, and even one random number per attribute, the total size of the header with n_{max} attributes is approximately 20 Kbits which is still negligible in comparison with the payload size.

Concerning the computation assurance solution, we already defined n_{max}, and we need to define ϕ and t.

t first is used as a security parameter, since the probability of success of an adversary can be bounded by 2^{-t}. As explained in section 8.4.3, it is not necessary to choose a very high value for t as it does not lead to revealing a secret but only to being able to cheat on the matching ratio. By choosing $t = 10$ for example the probability of success of an attacker would still be bounded by 10^{-3} in the most favorable case for the attacker and other probabilities of success are presented in table 9.1. This table shows that the probability of success for significant attacks is very low (for reference the typical security margin for symmetric encryption is $2^{-80} \approx 10^{-24}$). Of course it is possible to choose a higher value for t to make sure that even in the most favorable case the attacker would not succeed with probability more than 2^{-80} but t impacts first on the counting Bloom filter processing time (each element requires the computation of t hash values) and second and more importantly on the filter size. We therefore believe that choosing a smaller value for t as we did is a better trade-off.

The number of positions ϕ of the counting Bloom filter according to the strategy

Table 9.1: Probability $\mathcal{P}_{adv}(p_{j_{legit}}(M') \rightarrow p_{j_{legit}}(M') + p_{j_{mal}}(M'))$ of an adversary N_j with legitimate matching ratio $p_{j_{legit}}(M')$ to successfully claim a matching ratio of $p_{j_{legit}}(M') + p_{j_{mal}}(M')$ with a message M' containing n attributes in the header. The general parameters used for the counting Bloom filter are $n_{max} = 20$, $t = 10$, and $\phi = 289$.

n \ $\mathcal{P}_{adv}$	$0 \rightarrow \frac{1}{\lvert L_M \rvert}$	$0 \rightarrow \frac{2}{\lvert L_M \rvert}$	$0 \rightarrow \frac{1}{2}$	$0 \rightarrow 1$	$\frac{1}{2} \rightarrow \frac{1}{\lvert L_M \rvert} + \frac{1}{2}$	$\frac{1}{2} \rightarrow 1$	$1 - \frac{1}{\lvert L_M \rvert} \rightarrow 1$
6	5.10^{-8}	3.10^{-15}	1.10^{-22}	2.10^{-44}	9.10^{-11}	7.10^{-31}	2.10^{-15}
10	5.10^{-6}	2.10^{-11}	2.10^{-27}	4.10^{-54}	1.10^{-8}	1.10^{-40}	2.10^{-15}
20	1.10^{-3}	9.10^{-7}	7.10^{-31}	5.10^{-61}	5.10^{-6}	4.10^{-54}	2.10^{-15}

explained in section 8.4.3 should be $\phi = \left\lceil \frac{n_{max} t}{\ln(2)} \right\rceil = 289$ with $n_{max} = 20$ and $t = 10$. We first observe that $\phi << 10^{15}$, and thus if we choose to allocate 4 bits for each position in the counting Bloom filter, the total size of the counting Bloom filter is slightly more than 1 Kbit while the probability of a bucket overflow to happen would be less than 2.10^{-12}. The size of the counting Bloom filters is therefore really negligible in comparison with the size of the header: the computation assurance scheme is very efficient and does not add a significant overhead to the privacy preserving solution alone. Just for comparison purposes, applying the basic idea for computation reliability presented in section 8.2.2 would require an overhead of 10 Kbits, thus the use of counting Bloom filters really offers a decisive advantage from a performance perspective on top of the advantage from a privacy point of view.

On this matter, we mentioned in section 8.4.1 that the size of the set of possible preimages that lead to a counting Bloom filter is around $\frac{q}{\phi^t} \approx 2^{448}$. This proves that a brute-force attack to break the privacy-preserving properties of the computation assurance solution is out of reach of current computing power.

9.3 Extensions

In this section we discuss some possible extensions to the proposed schemes.

9.3.1 Revocation

As in many DTN protocols, key revocation is a difficult problem. From a management perspective, the TTP provides all keys during the setup phase but it does not play any role in the runtime phase. This offline TTP is therefore compatible with an opportunistic network.

The problem of key revocation is a new problem that arises in the particular configuration in which we use PEKS but it was not an issue in the original PEKS scheme of Boneh et al. [BCOP04]. In [BCOP04], the revocation of the capability of using the `SE-Trapdoor` function was directly linked with the revocation of the private key of the destination and was therefore a classical problem. In our design, the issuer of the trapdoors is the TTP, and the same trapdoor is given to all nodes with the same profile. Yet, profiles are dynamic and therefore it is important to be able to distribute new trapdoors to nodes which profile changed. In that case, trapdoors of other nodes (which profile did not change) should also be updated in order to guarantee a property like forward secrecy.

Concerning identity-based encryption, Boneh et al. [BF01] suggest to solve the problem of revocation by adding a timestamp to identities. Instead of simply using the identity ID of a node as public key, one would therefore use (ID, ϑ) where ϑ is a timestamp. The TTP gives the node with identity ID the private key corresponding to (ID, ϑ) at time ϑ. The private key is automatically revoked after a period of time, because the public key that is used becomes $(ID, \vartheta + 1)$.

In order to come up with a solution to key revocation in the context of opportunistic networks, we propose to divide the time in epochs θ^l, where l is a positive integer. For each epoch θ^l, the TTP generates a new private/public key pair sk^l_{TTP}/pk^l_{TTP} with $pk^l_{TTP} = sk^l_{TTP}P$, and nodes need to contact the TTP once during each epoch to get their updated keys. The use of epochs allows for a very loose synchronization between nodes and is therefore suitable for opportunistic networks.

The epoch's duration is chosen according to the network parameters such that all nodes can access the TTP once during an epoch. The duration of an epoch is also considered longer than the time required by any packet to reach any node in the network.

During epoch θ^l nodes need to enter setup phase with the TTP once to fetch the secrets corresponding to sk^l_{TTP}/pk^l_{TTP}, but they use the secrets of epoch θ^{l-1} to encrypt the messages because some nodes might not have fetched their secrets of epoch θ^l yet. Nodes also need to store the secrets of epoch θ^{l-2}: indeed during epoch θ^{l-1}, nodes use the secrets corresponding to θ^{l-2} to encrypt the messages. It is therefore possible that a message was sent at the end of epoch θ^{l-1} encrypted with the secrets of θ^{l-2} and is in the network at epoch θ^l. Since the duration of an epoch is longer than the time required by any packet to reach any node in the network, packets encrypted with older secrets than those of θ^{l-2} are automatically destroyed or dropped.

To summarize, nodes have a very loose synchronization since they only need to enter setup phase once in each epoch θ^l. The amount of secrets that they need to store is three times the amount of secrets required for one epoch; they indeed need to store the secrets corresponding to:

- θ^l once they fetch them,
- θ^{l-1} to encrypt the messages such that they can be decrypted by nodes that have not yet fetched the secrets of θ^l,
- θ^{l-2} to be able to decrypt the messages sent during epoch θ^{l-1} and that have not expired yet.

The use of epochs therefore enables "delay-tolerant" key revocation over three epochs, while being compatible with the principles of opportunistic communication.

9.3.2 Protection against malicious TTP

The TTP plays a crucial role in the proposed framework. The TTP is indeed the only entity which can compute trapdoors and private keys of nodes based on the attribute values. The TTP is a trusted entity and therefore is assumed to behave properly but it is also a single point of failure that has the capability to decrypt all messages by using its private key sk_{TTP}.

It is therefore important to distribute the capabilities of the TTP and to remove the single point of failure. This is a new issue with respect to the original PEKS architecture. Indeed, while in the original design of Boneh et al. [BCOP04], the destination is computing the trapdoors and therefore there is no problem of key escrow, in our scheme the TTP is computing all trapdoors for other nodes. To this extent, we propose to distribute the trust and the capabilities on several third parties by adding the contribution of each one as follows.

Assume there are ω parties denoted by TTP_k with $1 \leq k \leq \omega$. All these entities use the same global parameters but each one generates a different private/public key pair denoted sk_{TTP_k}/pk_{TTP_k} with $pk_{TTP_k} = sk_{TTP_k}P$.

Then, a source node N_S encrypts the header $\mathcal{H}(M)$ by using the sum of all the public keys of the TTPs as public key:

$$H'(M) = ||_{j \in L}(E_j, \texttt{SE-PEKS}(params, \sum_{k=1}^{\omega} pk_{TTP_k}, A_{M,j})).$$

During setup phase, nodes N_i need to fetch trapdoors $T_{k_{i,j}} = \texttt{SE-Trapdoor}(params, sk_{TTP_k}, A_{i,j})$ generated by each TTP_k. The total trapdoor associated with each attribute is

$$\sum_{k=1}^{\omega} T_{k_{i,j}}.$$

These trapdoors can then be used as second input of the `SE-Test` function to compute the matching ratio. This scheme is consistent thanks to the bilinearity of the map $\hat{e}$; the proof is very similar to the proof of consistency of the multiple identity-based encryption scheme presented in Theorem 6.4.1 and is therefore omitted.

Concerning the security of the scheme, if up to $\omega - 1$ TTPs are malicious and collude, they cannot reconstruct a complete trapdoor. A complete trapdoor is indeed:

$$\sum_{k=1}^{\omega} T_{k_{i,j}} = \sum_{k=1}^{\omega} sk_{TTP_k} H_1(A_{i,j}) = \left(\sum_{k=1}^{\omega} sk_{TTP_k}\right) H_1(A_{i,j}).$$

Imagine that the last $\omega - 1$ collude then the trapdoor is composed of:

$$\underbrace{\sum_{k=1}^{\omega} T_{k_{i,j}}}_{unknown} = \underbrace{sk_{TTP_1} H_1(A_{i,j})}_{unknown} + \underbrace{\left(\sum_{k=2}^{\omega} sk_{TTP_k}\right) H_1(A_{i,j})}_{known}.$$

Here *known* and *unknown* refer to what is known or unknown by the colluding TTPs. So even if $\omega - 1$ TTPs collude they do not have any clue about the total trapdoor. If the malicious TTPs collude with a node, then they can access the total trapdoor of course but they cannot expose the private key of the remaining honest TTP because:

$$\underbrace{\sum_{k=1}^{\omega} T_{k_{i,j}}}_{known} = \underbrace{sk_{TTP_1}}_{unknown} \underbrace{H_1(A_{i,j})}_{known} + \underbrace{\left(\sum_{k=2}^{\omega} sk_{TTP_k}\right) H_1(A_{i,j})}_{known} .$$

To expose sk_{TTP_1} the malicious entities need to be able to compute a discrete logarithm in $\mathbb{G}_1$, where the Diffie-Hellman problem is supposed to be hard.

Hence even if $\omega - 1$ TTPs collude with some nodes, they still cannot find the private key of the remaining TTP, and the security of the scheme is preserved. If all ω TTP collude though, it becomes as if the scheme consists of just one TTP, and the problems explained at the beginning of this section arise.

The same idea can be used to solve the problem of distributing the trust over multiple TTPs for the payload encryption: the sum of the keys of the various TTPs is used in the encryption and the decryption process. To be more precise, the encryption of the payload of message M uses the sum $\sum_{k=1}^{\omega} pk_{TTP_k}$ of the public keys of all TTPs as parameter:

$$\mathcal{PLD}(M') = \texttt{MIB-Encrypt}(params, \{A_{M,j}\}_{j \in L_M}, \mathcal{PLD}(M)),$$

where the public key in *params* is $P_{pub} = sum_{k=1}^{\omega} pk_{TTP_k}$.

The nodes need to retrieve private keys corresponding to their attributes $A_{k,priv_{M,j}}$ from TTP_k. Then, the decryption at the destination N_D of $\mathcal{PLD}(M')$ is performed with the sum of private keys of all TTPs for all attributes:

$$\mathcal{PLD}(M) = \texttt{MIB-Decrypt}(params, \mathcal{PLD}(M'), \{sum_{k=1}^{\omega} A_{k,priv_{M,j}}\}_{j \in L_M}),$$

where the public key in *params* is $P_{pub} = \sum_{k=1}^{\omega} pk_{TTP_k}$.

The proof of consistency is again based on the bilinearity of the pairing and is omitted.

As a conclusion, in order to alleviate the trust on a single entity, we propose to distribute the security capabilities (private key and trapdoor computation) among several third parties. In this new setting, nodes still use the functions defined in section 6.5 and 7.4 but apply the following simple modification:

- $pk_{TTP} = \sum_{k=1}^{\omega} pk_{TTP_k}$,
- $A_{priv_{i,j}} = \sum_{k=1}^{\omega} A_{k,priv_{i,j}}$,
- $T_{i,j} = \sum_{k=1}^{\omega} T_{k_{i,j}}$.

The difference is that nodes need to contact several TTPs to get their secrets (but then they only need to store the sum of all these secrets so this does not incur an additional cost in terms of storage) and that no single TTP can break user's privacy or confidentiality; only the collusion of all ω TTPs can result in such exposure.

9.3.3 Weighting the attributes

In PROPICMAN [NGP07] and HiBOp [BCJP07], authors propose to weight the attributes to express the relative importance of each attribute: for example, the workplace is sometimes more valuable to perform context-based forwarding than the nationality. In PROPICMAN and HiBOp, the weights are system parameters and they are defined along with the attribute names: the weights do not change over time, and are the same for all nodes.

Thanks to the properties of counting Bloom filters, it is possible to cope with the need for weights in our security mechanisms: a straightforward solution consists in incrementing the positions in the counting Bloom filter by the weight of the attribute instead of 1 for each hash function. This solution maintains the properties of the counting Bloom filter with respect to the partial order relation and does not affect the design of the solution.

The only parameter that needs to be reconsidered is n_{max} which needs to be incremented to take into account the possibility to weight the attributes. This in turn results in an increment in the size ϕ of the counting Bloom filter since $\phi = \frac{n_{max} t}{\ln(2)}$, but this is not really an issue if the weights are kept reasonably small.

The real issue is from a privacy perspective: the fact that each attribute results in an increment of more than one for each hash function result in a modification of the distribution of the values in the counting Bloom filter. The probabilistic analysis presented in section 8.4.2 does not hold anymore, and attackers would be able to infer some information on the attributes in the counting Bloom filter through a statistical analysis.

This issue is not due to our computation assurance mechanism though: even without this mechanism, the simple fact that attributes are weighted and not all on equal foot enables nodes to infer information on the attributes of their neighbors by analyzing at the matching ratio. We illustrate this fact with a simple example. Imagine that there are only three attributes of weights 2, 2, and 3 respectively. Then if a node claims a matching ratio of $\frac{3}{7}$ it indirectly reveals that the shared attribute is the third one.

Weighting the attributes therefore presents a risk of privacy exposure through statistical analysis. Our mechanisms can cope with weights but they do not solve the privacy exposure risk. Solving this issue is an interesting future research problem.

9.4 Conclusion

In this part, we focused on the analysis of security issues in context-based forwarding mechanisms. We studied the payload confidentiality, user privacy and computation assurance requirements in such protocols and defined the security primitives required to perform secure content-based forwarding within trusted communities. These primitives require the use of carefully chosen public functions to ensure both privacy and forwarding operations.

We then presented an original solution to solve the issues of confidentiality and privacy which is derived from Identity-Based Encryption and Public Encryption with Keyword Search. The use of identity-based encryption in a multiple attribute setting enforces end-to-end payload confidentiality with no end-to-end key management, while the specific use of PEKS allows intermediate nodes to securely discover partial matches between their profile and the message context while preserving user privacy in the trusted communities assumption. The encryption functions are secure against dictionary attacks and traffic analysis, thanks to the use of an internal random number in the input. The solution relies on an offline TTP.

Preserving privacy through computation on encrypted data does not guarantee the correctness of the computed data though. We therefore defined an additional mechanism that enforces trust in the authenticity of the claimed matching ratio. The design of this scheme in turn takes into account new privacy requirements. This computation assurance mechanism is based on preimages of one-way function for the assurance part, and on counting Bloom filters for the privacy and performance aspects.

Our schemes suit opportunistic networks well because they incur reasonably low storage and computation overhead and they rely on an offline TTP which is not required for the correct execution of the protocol during the communication.

Part III

Privacy-Preserving Content-Based Routing in Mobile Opportunistic Networks

Chapter 10

Privacy in Content-Based Communication

10.1 Introduction

As opposed to context-based communication, messages in content-based communication are not sent to a predefined (not even implicit) destination: messages are simply sent to all interested nodes, and are forwarded based on their content and the interests of users. These interests are independent of the characteristics of the nodes and might change frequently therefore a solution relying on private keys or trapdoors for each node and for each interest is out of question.

A classical content-based communication example is the publish/subscribe paradigm which allows for flexible and dynamic communication among a large number of participants. As opposed to classical messaging systems, in publish/subscribe, communicating parties are loosely coupled in that the source of the information does not need to know potential recipients of the information and the recipients do not need to know where the information originates from. In a content-based publish/subscribe (CBPS) system the forwarding of data segments between the sources and the recipients does not take into account the addresses of communicating parties but is performed based on the relationship between the content of each message and the interest of recipients. The recipients, who inform the publish-subscribe system about the messages they are interested in through subscription messages, are thus called subscribers. Publish-subscribe applications range from large scale content distribution applications such as stock-quote distribution to dynamic messaging between loosely-coupled parties in on-line social networks. The properties of content-based publish/subscribe are interesting for opportunistic communication but they require some adaptations: contrary to classical CBPS, content-based communication in opportunistic networks cannot rely on an infrastructure. Furthermore, opportunistic networks are peer-to-peer by essence and therefore nodes should be able to advertise for their interests and forward other nodes messages at the same time.

From a security perspective, the flexibility of content-based communication comes with a high cost in increased exposure in terms of data security and privacy. Apart from classical data security concerns such as confidentiality and integrity of messages, source authentication, access control and authorization of subscribers, publish-subscribe also raises new challenges inherent to the collapsed forwarding scheme that is the underpinning of content-based communication. In classical layered communication systems, the application layer information can be protected with various security mechanisms like encryption and message authentication without affecting the underlying data forwarding mechanisms implemented in the network layer. In case of content-based communication, protection of the content with similar security mechanisms would conflict with the forwarding functions since the latter rely on the very content that is being transmitted for their basic operations. Content-based communication therefore calls for mechanisms enabling computation on encrypted data. To be more precise, content-based communication requires new solutions that allow intermediate nodes to perform routing operations based on data protected with encryption mechanisms. The first requirement is for a secure forwarding mechanism that would achieve the look-up in forwarding tables using encrypted content as the search key. Furthermore, an important privacy requirement in content-based publish-subscribe is the confidentiality of the messages through which subscribers inform the network about their interests. Whilst encryption of these messages appears to be a suitable solution for subscriber privacy, such encryption operation raises an additional challenge for the forwarding mechanism. Hence not only the search key for the look-up mechanism but also the forwarding table itself would be based on encrypted data. Some existing security primitives such as keyword search with encrypted data or private information retrieval seem to partially meet the new requirements raised by secure and privacy preserving data forwarding in content-based communication but none of the existing security mechanisms actually addresses both the problem of secure look-up and the secure building of forwarding tables in a comprehensive manner.

In this chapter, we suggest a set of security mechanisms that allow for privacy-preserving forwarding of encrypted content based on encrypted subscriber interest messages. The main advantages of this solution are that it achieves privacy of the subscribers with respect to their interests in a potentially hostile model whereby nodes do not trust one another. The solution relies on a scheme called multi-layer encryption that allows intermediate nodes to manage forwarding tables and to perform content forwarding using encrypted content and based on encrypted subscriber messages without accessing the cleartext version of those data. Our solution further avoids key sharing among end-users and targets a content-based communication model where nodes can be subscribers and forward messages at the same time.

10.2 Reference Model

In this section we first describe the classical content-based Publish/Subscribe model. We then show the limitations of this model and propose an extended model adapted to opportunistic networking.

10.2.1 Content-Based Publish/Subscribe (CBPS)

Content-based Publish/Subscribe is the most prominent example of the content-based communication paradigm: despite the lack of wide-area deployment of CBPS applications nowadays (which is analyzed by Raiciu et al. in [RRH06]), many solutions including commercial products (e.g. [OPS10, Web, PSH10, DDS]) are readily available to benefit from this technology, and the applications that could take advantage of CBPS cover a broad range of scenarios (stock quotes [WQA+02], RSS feeds [LRS05], security alerts [CCC+05], and even location based services [CCR03]). CBPS assumes that messages are forwarded from publishers to subscribers through a dedicated infrastructure based on the content of the messages. To be more precise, the classical CBPS model as described in many papers like [CRW01, SL07] consists of:

- **end-users** divided in:
  - **publishers** which publish information in the form of *event notifications*,
  - **subscribers** which express their interests in a certain content in the form of *subscription filters*,
- the CBPS infrastructure composed of **brokers** (intermediate nodes) whose task is to disseminate notifications sent by publishers to the interested subscribers. To this extent, brokers need to construct routing tables based on the received subscription filters, and look-up the event notifications in these routing tables.

In this model, the CBPS infrastructure can be viewed, from the perspective of each publisher, as a tree which root node is the publisher itself and which leaf nodes are the subscribers (whether interested in the content published by the publisher or not). This model is thus usually analyzed by considering the case of a network with only one publisher for the sake of simplicity. This model features a complete decoupling between the publisher and the subscriber, and the routing tasks are solely performed by the infrastructure of brokers (using the approach presented by Banavar et al. in [BCM+99] or SIENA [CRW01] for example); the logical communication tree is therefore constructed by the brokers, while publisher and subscribers do not require knowledge of this topology, they just communicate with the broker closest to them.

10.2.2 Content-based Communication in Mobile Opportunistic Networks

The classical CBPS model presented in the previous section is interesting to provide an efficient content-based message dissemination service over an infrastructure. Most of the research on CBPS so far has focused on fixed networks, but the inherent characteristics of the CBPS communication paradigm are also attractive to opportunistic networks. Indeed, communication in CBPS:

- presents a strong decoupling between publishers and receivers and therefore does not a-priori require end-to-end connectivity,

- is asynchronous, as the publisher does not require an acknowledgment from the subscriber(s) after an event notification: the infrastructure is taking care of reliably delivering events to the interested subscribers,
- is inherently disseminational, because it allows a publisher to notify an event to many subscribers,
- is scalable and dynamic since it relies on local information only.

In spite of this combination of characteristics, adapting CBPS to MobiOpps, is not a simple task. The main problem is that MobiOpps cannot assume an infrastructure of brokers, as MobiOpps are ad hoc and peer-to-peer in essence: all nodes have to cooperate and take part in the forwarding process. This raises the requirement for a peer-to-peer content-based communication model, where all nodes might assume the roles of publishers, subscribers and brokers. In the literature, few approaches have been proposed relying on different models such as acyclic directed graphs ([DGRS03]), multicast communication over MANETs ([ZS00]) or optimized trees in wireless networks ([HGM03]). We present an overview of these approaches in section 10.8. Since the communication model is not the main purpose of our work, we consider for the sake of simplicity the model of optimized trees.

In this model, we assume that each node constructs a spanning tree for data dissemination: this spanning tree is rooted at the node (acting as publisher) and reaches all other nodes. The spanning tree used for communication depends on the node publishing content, thus for the sake of simplicity we consider the case of a unique publisher (the case of multiple publisher corresponds to the superposition of the associated spanning trees). To be more precise, we assume that the network is composed of n nodes N_i for $1 \leq i \leq n$.

The communication graph is a tree rooted at a node N_p ($p \in [1, n]$), which acts as a publisher in this instantiation of content-based communication. The remaining nodes are organized in a logical tree based on the physical graph: nodes linked in the tree are in communication range of one another, but not all nodes in communication range share a logical link. We denote by $Par(i) \in [1, n]$ the index of the parent of N_i (i.e. $N_{Par(i)}$ is the parent of N_i) for all nodes except the root, and by $Chd(i) \subset [1, n]$ the set of indexes of children of N_i (i.e. $\forall j \in Chd(i),\ Par(j) = i$) for all nodes except the leaves.

We can also recursively define::

- $Par^l(i) \in [1, n]$, the index of the l-th level (or l-th hop) parent of node i (for $l \geq 1$) as $Par^1(i) = Par(i)$ and $Par^l(i) = Par(Par^{l-1}(i))$ if it exists (otherwise $Par^l(i) = \oslash$),
- $Chd^l(i) \subset [1, n]$ the set of indexes of l-th level (or l-th hop) children of node i (for $l \geq 1$) as $Chd^1(i) = Chd(i)$ and $Chd^l(i) = \bigcup_{j \in Chd(i)} Chd^{l-1}(j)$ if it exists (otherwise $Chd^l(i) = \oslash$).

With both these definitions, we can define $\mathcal{N}^l(i)$ the l-hop neighborhood of node N_i as the indexes of all nodes at distance less than l hops from node N_i:

$$\mathcal{N}^l(i) = \bigcup_{1 \leq k \leq l} Par^k(i) \cup Chd^k(i).$$

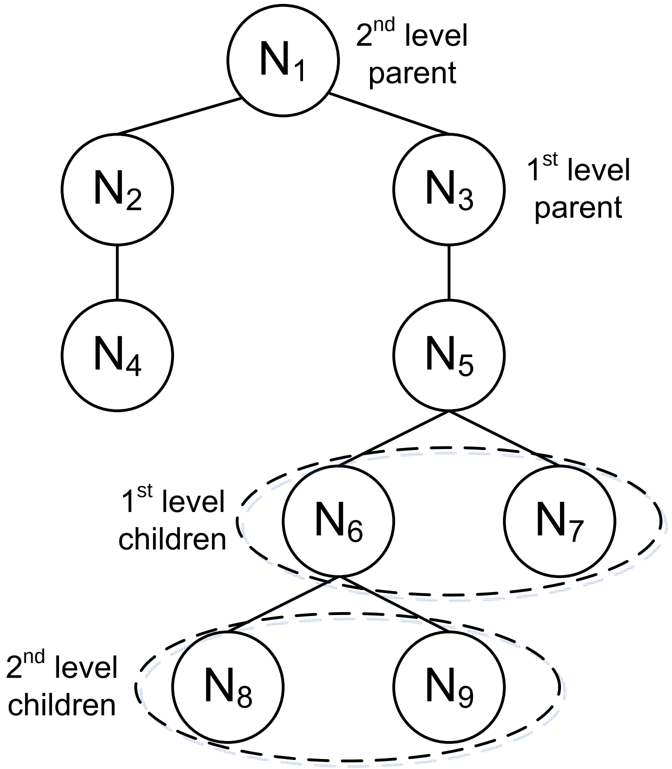


Figure 10.1: Example of a content-based tree structure. The Publisher is N_1. If we consider the node N_5, it's one level parent is node N_3 and its second level parent is N_1. It's first level children are N_6 and N_7, and its second level children are N_8 and N_9. We thus have $\mathcal{N}^1(5) = \{3, 6, 7\}$ and $\mathcal{N}^2(5) = \{1, 8, 9\}$.

These notations are illustrated with a simple example on Figure 10.1.

Note that, as required, this model does not differentiate between brokers and subscribers: all nodes communicate with their neighbors, build routing tables, and forward subscription filters and event notifications; some nodes are interested in the events notified while some are not.

The publisher here has a specific role as it is the root of the tree, but it can also act as subscriber and forwarder (broker) in the trees rooted at other nodes.

10.2.3 Messages in Content-Based Communication

In content based communication, there are two main types of messages:

- **subscription filters**, sf, sent by a node to advertise its interest in some content,
- **event notifications**, en, sent by a node to publish content.

Information contained in each event should fit within an event schema, and the subscription filters are predicates against this schema. Ideally content-based communication should support complex subscription filters that encompass any logical expression on any set of keywords. Yet most classical CBPS models only consider equality filters with only one keyword and events are composed of two parts: one routable attribute ra (corresponding to a keyword) and a payload pld. The equality matching is indeed the mostly used filtering function in the literature since it can be used as a basis to support range queries as introduced in [RR06]. Nodes acting as brokers use this matching operation between filters and the routable attribute of event notifications to route published content. If we take as an example the commonly used stock quote dissemination problem, a subscription filter could be $(price = 120)$ which would match an event like

$$(\underbrace{price = 120}_{\texttt{routable attribute}}, \underbrace{[symbol = "STM", price = 120, volume = 1000]}_{\texttt{payload}}).$$

In [CRW01], authors show that content-based routing and in-network matching are vital for the performance and scalability of content-based systems. To this extent, if two subscriptions match the same content, then only one of them should be propagated in the network and only one entry should be added in the routing table. We thus need to define a notion of equivalence between filters in order to aggregate them:

Definition 10.2.1 *Two filters f_1 and f_2 are equivalent if they match the same events.*

To sum up, our reference model consists of peer nodes organized in a communication tree rooted at the publisher. Nodes can advertise their interests through subscription filters but only the root publishes content through event notifications. Subscription filters and event notification need to follow a predefined format such that nodes can match subscription filters with event notifications. Thus, nodes N_i also have to build routing tables RT_i based on the subscription filters that they receive, to forward these subscription filters to their parents $N_{Par(i)}$ (unless an equivalent filter has already been forwarded), and to forward event notifications to interested nodes by looking-up the event notifications in the routing tables.

We now focus on the main subject tackled in this chapter, namely privacy issues in content-based communication.

10.3 Privacy Issues in Content-Based Communication

In this section, we focus on the problem of user privacy. We first present the confidentiality requirements that are necessary to preserve user privacy, and then the implication of confidentiality requirements on routing. We then describe the threat model that we consider.

10.3.1 Privacy, Confidentiality, and Routing

As explained in section 3.4, privacy is a critical criterion for acceptance of MobiOpps. Privacy from a node acting as a subscriber point of view refers to the fact that subscribers do not want any other nodes to spy on their interests and be able to profile them. There are many aspects of privacy protection, one essential requirement is to guarantee data confidentiality.

Confidentiality in CBPS networks has first been analyzed in [WCEW02] where the authors identify three confidentiality issues, defined as follows:

- **Information confidentiality**: Can the infrastructure perform content-based routing, without the publishers trusting the infrastructure with the content?
- **Subscription confidentiality**: Can subscribers obtain dynamic, content-based data without revealing their subscription filters to the publishers or to the infrastructure?
- **Publication confidentiality**: Can publishers control which subscribers may receive particular publications?

These issues can directly be translated in content-based communication, except that the roles are shared by all nodes; in particular the infrastructure is replaced by all the nodes. Subscription confidentiality is obviously a must to preserve subscribers' privacy but it is not sufficient: we also need to take information confidentiality into consideration, otherwise adversaries could infer the subscription filter by analyzing the information which matches it. From a publisher's perspective privacy may not be as crucial. Publishers publish some content which is meant to be received by some nodes, hence they often do not require a full-fledged privacy but they require publication confidentiality. Publication confidentiality is an access control rather than privacy issue: publishers want to be able to authorize certain subscribers to be able to access the content they publish while preventing unauthorized ones from learning valuable information about it. Since publication confidentiality is not necessary to ensure privacy, we do not consider it in the sequel of this chapter, especially that orthogonal solutions can be developed to ensure it.

To ensure privacy we hence have to fulfill two main confidentiality requirements, namely:

- **Information confidentiality:** this confidentiality requirement may look paradoxical as content-based routing is, by definition, based on evaluations of the content of notifications against subscription filters. The challenge is to be able to perform evaluations on event notifications against subscriptions while data is encrypted and without leaking information on the corresponding content.
- **Subscription confidentiality:** this is the dual problem of information confidentiality. Nodes sending a subscription filter do not want to reveal their interests to other nodes but they still want to receive the content they are interested in and only this

one. The challenge in this case is to match a content with an encrypted subscription without disclosing the subscription filter. This is a problem of secure function evaluation, where a broker has to evaluate a hidden function (the filter which was encrypted by the subscriber).

In summary, information and subscriber confidentiality in content-based communication call for new mechanisms to achieve privacy-preserving routing of encrypted data with the capability of matching encrypted event notifications against encrypted subscription filters in order to ensure user privacy. As explained in section 3.4 the level of information that can be revealed without threatening the privacy of users depends on the trust level. In classical CBPS, the trust among nodes can depend on the roles that nodes assume: for example nodes might be willing to trust the infrastructure more than end-users. In content-based communication, however, all nodes assume all roles and therefore we consider the trust level to be uniform among nodes. In practice, nodes might have stronger trust relationship based on criteria external to the content-based communication model (for example based on community membership), but we consider the most challenging case where nodes do not have a priori trust with any other nodes, and we thus target the privacy model 4 (or full privacy) described in section 3.4. In this model, information and subscription confidentiality must be guaranteed against all other nodes. In order to ensure information and subscription confidentiality, some encryption mechanisms will be used and thus the use of such mechanisms raise entirely new problems for content-based forwarding:

- **Building routing tables:** Nodes have to build routing tables using routing information -subscription filters- which is disseminated in the network to subsequently allow for the routing of content in a possibly optimized fashion. The challenge in our case is that nodes have to build their routing tables with encrypted filters (to satisfy the subscription confidentiality constraint) and to aggregate theses encrypted routing information. Aggregation of routing tables' entries is not strictly a security concern but is nonetheless a strong requirement from the point of view of performance.
- **Look-up:** Once routing tables are built, nodes can forward data -event notifications- from publishers to interested subscribers in an optimized way. The challenge for nodes acting as brokers in the dissemination process is to be able to perform the look-up of encrypted data (to fulfill the information confidentiality requirement) in routing tables where entries include encrypted subscription filters.

User privacy thus calls for a solution that achieves privacy-preserving routing of encrypted data based on encrypted routing information. We define more precisely the attacker model in the next section.

10.3.2 Threat Model and Security Assumptions

As in many works concerning privacy (e.g. [SL07]), we assume a honest-but-curious model for all nodes: we assume that the nodes are computationally bounded and do not deviate from the designed protocol, but they may be interested in learning more than needed to correctly run the protocol to expose other user privacy. A curious publisher may indeed be interested in knowing which nodes are interested in the content it publishes. More generally, any node may try to sneak on others to determine what their interest are or at least if they have some common interests. In particular, forwarding nodes may eavesdrop on the messages routed through them to discover the content of an event notification or a subscription filter.

However, all the nodes are honest and do not deviate from the designed protocol, meaning for instance that nodes correctly route the information they receive as indicated by the protocol, they do not drop packets or forward packets in a wrong way. Denial of service attacks are not taken into consideration. Furthermore, nodes reveal neither their secret nor received data to other nodes. We also take into account malicious but passive nodes outside of the network, which can overhear communications and try to break end-users' privacy.

In summary, in this chapter, we are only interested in guaranteeing that nodes cannot discover the interests of other nodes, while ensuring correct networking operation over encrypted data. One naturally turns to searchable encryption and keyword search [SWP00, BCOP04] that are cryptographic techniques most likely to meet the requirements of secure routing in content-based communication. Unfortunately none of the existing searchable encryption and keyword schemes address both the secure forwarding and the table building requirement of content-based communication. We thus tailor in this chapter a dedicated solution to meet the specific requirements of privacy-preserving content-based routing.

10.4 Privacy-Preserving Routing with Multiple Layer Encryption

10.4.1 Multiple layer commutative encryption (MLCE)

The basic idea behind our solution is to use multiple layer commutative encryption (MLCE) in order to meet the privacy requirements raised by content-based communication. MLCE allows intermediate nodes in charge of routing protected messages to perform secure transformations without having access to the data that is being transferred. This feature of MLCE lends itself very well to solving the problem of routing encrypted data as raised by content-based communication.

In multiple layer encryption, data is encrypted several times with different keys. In the case where the encryption layers all use the same cryptosystem, and if this cryptosystem is commutative, then the layers can be added and removed in any order. An encryption mechanism $\mathcal{E}$ is commutative if, for any data d, any keys k_1, k_2 we have :

$$\mathcal{E}_{k_2}(\mathcal{E}_{k_1}(d)) = \mathcal{E}_{k_1}(\mathcal{E}_{k_2}(d)).$$

We propose to use multiple layer commutative encryption in order to ensure secure routing in content-based communication where the publisher publishes encrypted events and the nodes acting as subscribers send their encrypted subscription filter to other untrusted nodes. The idea is for the subscriber to encrypt its subscription filter with $lr \geq 2$ layers corresponding to the lr next hops, and for the publisher to do the same with its event notifications. Intermediate nodes remove one encryption layer and add a new one without destroying the other layers so that the data is always protected by at least $lr - 1$ layers of encryption. Thus nodes forwarding messages do not have access to data in cleartext. Still, this mechanism allows secure look-up as well as efficient and secure routing table building thanks to the commutativity of the layers. The number of layers lr is a security parameter that has a performance impact, and we discuss the choice of the parameter lr in section 10.7.3.

To further introduce the solution, let us consider a minimalist example. In this example, we consider three nodes in line, namely a subscriber denoted by N_s, then a broker denoted by N_b and finally a publisher denoted by N_p. We denote by $k_{i,j}$ a key shared between node N_i and N_j. N_s encrypts its data x_s with $\mathcal{E}_{k_{s,p}}(\mathcal{E}_{k_{s,b}}(x_s))$ and so does N_p with its data x_p: $\mathcal{E}_{k_{s,p}}(\mathcal{E}_{k_{b,p}}(x_p))$. The broker now can remove the layers corresponding to $k_{s,b}$ and $k_{b,p}$ respectively to obtain $\mathcal{E}_{k_{s,p}}(x_s)$ and $\mathcal{E}_{k_{s,p}}(x_p)$. Hence, it cannot access the data directly but it is able to perform a matching operation for the secure look-up since x_s and x_p are encrypted under the same keys.

Therefore, given a commutative cryptosystem we are able to do secure routing and hence protect the privacy of publishers and subscribers. Yet, commutative cryptosystems are scarce, and although many security solutions assume the existence of a commutative cipher, few of them deal with a concrete commutative cryptosystem. We developed a scheme based on the Pohlig-Hellman cryptosystem, that we carefully adapted to our case in order to provide a complete and concrete solution. Privacy-preserving routing with MLCE is achieved through four security primitives that are detailed in the next section.

10.4.2 Security Primitives

To further refine the privacy-preserving routing using MLCE we identify the generic operations required for secure event dissemination as follows:

- **ENCRYPT_FILTER:** used by nodes to generate encrypted subscription filters. On input a subscription filter and some keying material it outputs an encrypted version of the subscription filter.
- **ENCRYPT_NOTIFICATION:** used by the publisher to encrypt its notifications. On input an event notification and some keying material it outputs an encrypted version of the subscription filter.
- **SECURE_LOOK_UP:** allows a node to decide whether an encrypted notification matches one of the encrypted subscriptions of its routing table. This primitive should only return the boolean result of the matching operation.
- **SECURE_TABLE_BUILDING:** allows a node to build a routing table and to compare two encrypted subscriptions. As the previous primitive, this one should

return the boolean result of the matching operation, but it should not leak any additional information about the subscriptions.

The last primitive enables aggregation of equivalent subscription filters; aggregation is optional from a pure privacy point of view (it even induces additional difficulties) but it is vital from a performance point of view to comply with some content-based routing optimizations. If two subscriptions match the same content there is indeed no need to forward both of them to the node's parent. The node only needs to store both of them with the corresponding information on the child node in its routing table and it further forwards only one message to its parent.

All nodes process messages in a generic way to maintain the multiple layers and manage the security primitives at the same time. This processing is summarized in Table 10.1. In this table, the node is denoted by N_i, one of its lr-hop child by N_c (with $c \in Chd^{lr}(i)$), the encryption algorithm with a key k is $\mathcal{E}_k$ and the corresponding decryption algorithm is $\mathcal{D}_k$. $k_{i,c}$ denotes a key shared by N_i and N_c, while $k_{i,Par^{lr}(i)}$ is a key shared by N_i and $N_{Par^{lr}(i)}$. On the left N_i receives an encrypted subscription filter f from N_c and on the right an encrypted event notification en which matches the interest of N_c. We now formally describe our solution in the next section.

Filters propagation	**Event dissemination**
Remove an encryption layer: $\mathcal{D}_{k_{i,c}}(f)$	Remove an encryption layer: $\mathcal{D}_{k_{i,Par^{lr}(i)}}(en)$
Update the routing table RT_i: $SECURE\_TABLE\_BUILDING(RT_i, \mathcal{D}_{k_{i,c}}(f))$	Secure look-up: $SECURE\_LOOK\_UP(RT_i, \mathcal{D}_{k_{i,Par^{lr}(i)}}(en))$
Add an encryption layer: $\mathcal{E}_{k_{i,Par^{lr}(i)}}(\mathcal{D}_{k_{i,c}}(f))$	Add an encryption layer: $\mathcal{E}_{k_{i,c}}(\mathcal{D}_{k_{i,Par^{lr}(i)}}(en))$
Forward the message to its parent $N_{Par(i)}$	Forward the message to one of N_i's children

Table 10.1: Message processing at N_i

10.5 Proposed solution

We propose a solution that meets the requirements of information and subscription confidentiality based on multiple layers of Pohlig-Hellman encryptions whereby nodes can privately subscribe to events without the need to share a unique and common key with the publisher. This solution allows nodes to act as subscribers and brokers simultaneously by subscribing to events and sending their own subscription filters while performing the routing operation. Figure 10.2 presents an overview of the scheme on a simple example.

10.5.1 Choice of the Commutative Cryptosystem

In this section we motivate the choice of the Pohlig-Hellman cryptosystem to implement our solution.

Commutative cryptosystem are scarce since commutativity is often considered as a negative property from a cryptographer's perspective. Yet, in some scenarios (e.g. our solution) commutativity is mandatory for the correctness of the scheme. We therefore investigated known cryptosystems to find the most suitable one for our scheme.

We first investigated symmetric cryptosystems, as they are much more efficient than asymmetric ones from a performance perspective. Yet, to the best of our knowledge, the only commutative symmetric cryptosystem is the one-time pad, which consists in a simple XOR operation between the message and a key. The security of this scheme yet relies on a frequent update of the keys: as the name one-time pad suggests, the encryption key needs to be updated after each encryption. Hence the same key cannot be used to encrypt a subscription filter and a routable attribute, therefore this cryptosystem is not adapted to our solution.

Concerning asymmetric schemes, many schemes are inherently commutative because they are based on modular exponentiations. Yet a careful investigation shows that few fit the requirements of our solution. RSA ([RSA78]) for instance, is based on a modular exponentiation and is therefore commutative with respect to the encryption keys under a given modulus. The problem is that the modulus has to be the same for all nodes. Yet to generate a private/public key pair, nodes do not only require knowledge of the modulus, but also knowledge of the two primes composing it. However any node which knows the two primes composing the modulus can compute the private key corresponding to any public key under the same modulus. The only viable solution would therefore be to transfer the key distribution to a Trusted Third Party. This Trusted Third Party, would generate a modulus, and give each node a public/private key pair corresponding to this modulus. By doing so, nodes would know the modulus but not the primes composing it, and therefore they would not be able to expose the private keys of other nodes. This approach is not satisfying because it calls for a central authority to distribute the keys, thus defeating the self-organizing property of content-based communication, but it still can be acceptable if we consider the Trusted Third Party offline. However, even in that case, RSA as well as all public key cryptosystem make use of a randomization process such that encrypting twice the same message results in two different ciphertexts. If we remove this randomization step, then the scheme is prone to dictionary attack, as a malicious node could encrypt all possible words using any public key. Therefore neither RSA, nor any public key cryptosystem are suitable to our solution.

Our solution thus requires a commutative cryptosystem with a secret (shared) key. The Pohlig-Hellman cryptosystem lends itself well to our scheme as it has all the desired properties. It uses a modulus q, which a prime number and which is public. Then any node can generate a key pair, one key for encryption and one for decryption but both of these keys need to remain secret and are shared only among two nodes. Hence it is an asymmetric cryptosystem with secret keys only. We describe the Pohlig-Hellman cryptosystem more precisely in the next section.

10.5.2 The Pohlig-Hellman Cryptosystem

The Pohlig-Hellman cryptosystem [PH78] is defined as a tuple $(q, \mathcal{K}, \mathcal{E}, \mathcal{D})$ as follows:

- q is a large prime known by all nodes (it is a system parameter)
- $\mathcal{K}$ outputs a pair of keys (k_i, d_i) such that $k_i d_i \equiv 1 \mod (q-1)$;
- $\mathcal{E}(q, k_i, x)$ returns $x^{k_i} \mod q$;
- $\mathcal{D}(q, d_i, y)$ returns $y^{d_i} \mod q$

Since $k_i d_i \equiv 1 \mod (q-1)$, we have $x^{k_i d_i} \equiv x \mod q$ (Fermat's little theorem).

The encryption operation is based on an exponentiation and is therefore inherently commutative. Indeed, for any message x and pair of keys k_i, k_j:

$$\begin{aligned} \mathcal{E}\left(q, k_i, \mathcal{E}(q, k_j, x)\right) &= \left(x^{k_i} \mod q\right)^{k_j} \mod q \\ &= x^{k_j k_i} \mod q \\ &= \mathcal{E}\left(q, k_j, \mathcal{E}(q, k_i, x)\right). \end{aligned}$$

Similarly, the same property is verified by the decryption operation.

The addition and subtraction of a layer in our solution respectively corresponds to a Pohlig-Hellman encryption and decryption operation. Thanks to the commutative property of the Pohlig-Hellman cryptosystem, any node is able to add and suppress encryption layers if it stores the corresponding keys. Indeed, a direct consequence of commutativity is that, for any message x and pair of keys k_i, k_j:

$$\begin{aligned} \mathcal{D}\left(q, d_j, \mathcal{E}\left(q, k_i, \mathcal{E}(q, k_j, x)\right)\right) &= \mathcal{D}\left(q, d_j, \mathcal{E}\left(q, k_j, \mathcal{E}(q, k_i, x)\right)\right) \\ &= \mathcal{E}(q, k_i, x). \end{aligned}$$

Since the security of this cryptosystem relies on the hardness of the Discrete Logarithm Problem, the same key k_i can be used to encrypt several different messages (as opposed to one-time pad for example). Moreover, this cryptosystem is asymmetric in the sense that the encryption key differs from the decryption key. However, as opposed to classical asymmetric cryptosystems such as RSA [RSA78], if a node knows one of the keys, it can automatically deduce the remaining key. Therefore there is no "public key"; all keys are secret and they are only revealed to authorized nodes. The Pohlig-Hellman cryptosystem presents thus peculiar properties as it is an asymmetric system with secret (or shared) key.

In the sequel of this chapter, the key pair shared between node N_i and N_j is denoted indifferently by $(k_{i,j}, d_{i,j})$ or $(k_{j,i}, d_{j,i})$.

The Pohlig-Hellman cryptosystem being described, we formally define the four security primitives in the next sections.

10.5.3 Propagation of Subscription Filters and Building of Routing Tables

Concerning the propagation of subscription filters, subscribers first need to encrypt their filters with the primitive ENCRYPT_FILTER to preserve their privacy. These encrypted subscription filters are then processed by other nodes: after removing an encryption layer these nodes build their routing table using the SECURE_TABLE_BUILDING, then they add an encryption layer to maintain the MLCE properties and propagate the subscription filter further in the network.

10.5.3.1 ENCRYPT_FILTER

ENCRYPT_FILTER is used by a subscriber N_i to securely send its subscription filter to its parent. Following the MLCE scheme N_i needs to encrypt its subscription filter f with lr layers of encryption corresponding to the lr next hops. Therefore this primitive requires $lr + 1$ inputs: the subscription filter f and the keys corresponding to the lr next parents of N_i, namely $k_{i,Par(i)}$,...,$k_{i,Par^{lr}(i)}$.

Then:

$$\begin{aligned} ENCRYPT\_FILTER(f, k_{i,Par(i)}, ..., k_{i,Par^{lr}(i)}) &= \underbrace{\mathcal{E}(q, k_{i,Par^{lr}(i)}, \mathcal{E}(...\mathcal{E}(q, k_{i,Par(i)}, f)))}_{lr\ \texttt{layers}} \\ &= f^{k_{i,Par^{lr}(i)} \ldots k_{i,Par(i)}} \mod q \end{aligned}$$

Given this result, N_i sends this encrypted filter to its parent $N_{Par(i)}$and it also indicates that this filter is its own in a second part of the message. Therefore, N_i sends the message $[f^{k_{i,Par^{lr}(i)} \ldots k_{i,Par(i)}} \mod q; N_i]$ to its parent node $N_{Par(i)}$. The second part of the message is important for node $N_{Par(i)}$ to know which decryption key to use to decrypt the message.

10.5.3.2 SECURE_TABLE_BUILDING

This primitive aims at building optimized routing tables. Hence it updates the routing table either by adding a new row or by updating an existing row if it receives a matching filter.

When a node N_j receives a message from one of his children, the first step is to remove the encryption layer corresponding to the key shared by N_j and N_i indicated in the second part of the message. N_i is either the node which generated the message, or in the generic case (after the message has been propagated at least lr times) an lr-th hop children of N_j. These cases are similar in that in both cases one of the encryption layers uses the key $k_{i,j}$.

In the generic case, the message received by N_j is of the form

$$[f^{k_{Par^{lr-1}(i),Par^{2lr-1}(i)} \ldots k_{i,Par^{lr}(i)}} \mod q; N_i],$$

with $j = Par^{lr}(i)$. The message received can thus be rewritten as:

$$[f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)} \ldots k_{i,j}} \mod q; N_i].$$

Table 10.2: Processing of subscription filters by N_j. In this table, we assume that N_j receives a message $[sf, N_i]$, where sf is an encrypted subscription filter and N_i a node.

1. Remove one encryption layer from sf by using the key $k_{i,j}$:

$$sf \leftarrow \mathcal{D}(q, k_{i,j}, sf).$$

2. Look up sf in RT_j:

   - If sf already exists in RT_j:
     (a) Add N_i to the corresponding line in RT_j,
     (b) Break.
   - If sf does not appear in RT_j and if $Par^{lr}(j) \neq \oslash$:
     (a) Add a new entry in RT_j as $sf \rightarrow N_i$,
     (b) Add an encryption layer to sf by using the key $k_{j,Par^{lr}(j)}$ shared with the lr-th level parent of N_j:

     $$sf \leftarrow \mathcal{E}(q, k_{j,Par^{lr}(j)}, sf),$$

     (c) If $j = Par^{lr}(i)$, set $l = Par^{lr-1}(i)$, otherwise set $l = i$,
     (d) Send $[sf, N_l]$ to $N_{Par(j)}$.

By using the decryption operation of the Pohlig-Hellman cryptosystem, N_j thus retrieves the following filter encrypted with $lr - 1$ layers:

$$\mathcal{D}(q, k_{i,j}, \underbrace{f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)} \cdots k_{i,j}}}_{lr \texttt{ layers}}) = \underbrace{f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)} \cdots k_{Par(i),Par(j)}}}_{lr-1 \texttt{ layers}} \mod q.$$

N_j then operates on this filter encrypted $lr - 1$ times. N_j needs indeed to update its routing table and for this purpose, it needs to check whether the encrypted filter is equivalent to some entries in the routing table. For this purpose, N_j inputs its routing table RT_j and the filter encrypted with $lr - 1$ layers in the $SECURE\_TABLE\_BUILDING$ primitive with two possible outcomes:

- either N_j finds out that $f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)} \cdots k_{Par(i),Par(j)}}$ is equal to one of the entries of RT_j; in this case it updates the routing table RT_j by adding N_i in the corresponding row to be able to forward event notifications accordingly and does nothing else,
- or it finds out that $f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)} \cdots k_{Par(i),Par(j)}}$ is a new subscription that has no equivalent in the routing table RT_j; N_j then updates RT_j with a new row indi-

cating that notifications corresponding to $f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)}\dots k_{Par(i),Par(j)}}$ should be forwarded to N_i.

In the latter case, N_j also needs to forward the message further to its parent $N_{Par(j)}$. Therefore, to maintain the security of the MLCE scheme, N_j first adds another encryption layer with the key $k_{j,Par^{lr}(j)}$ shared with its lr-th hop parent by using the encryption operation of the Pohlig-Hellman cryptosystem:

$$\mathcal{E}(q, k_{j,Par^{lr}(j)}, \underbrace{f^{k_{Par^{lr-1}(i),Par^{lr-1}(j)}\dots k_{Par(i),Par(j)}}}_{lr-1\ \texttt{layers}}) = \underbrace{f^{k_{j,Par^{lr}(j)}\dots k_{Par(i),Par(j)}}}_{lr\ \texttt{layers}} \mod q.$$

Then N_j sends the message $[f^{k_{j,Par^{lr}(j)}\dots k_{Par(i),Par(j)}}, N_{Par(i)}]$ to $N_{Par(j)}$. The second part of the message enables $N_{Par(j)}$ to know the right decryption key to remove one layer of encryption (namely $d_{Par(i),Par(j)}$). The algorithm used by intermediate nodes to process encrypted filters is summarized in Table 10.2.

10.5.4 Content Distribution and Secure Look-up

Symmetrically, a publisher N_p first uses the $ENCRYPT\_NOTIFICATION$ to encrypt the event notification with the corresponding keys and forwards the packet to its children. Then, after removing one encryption layer, intermediate nodes, run the $SECURE\_LOOKUP$ primitive and they accordingly add another encryption layer and forward the message to interested nodes.

10.5.4.1 ENCRYPT_NOTIFICATION

ENCRYPT_NOTIFICATION is used by a publisher N_p to encrypt an event notification. The event notification en is composed of a routable attribute ra and a payload pld. Similarly to ENCRYPT_FILTER, ENCRYPT_NOTIFICATION encrypts the event notifications with lr layers of encryption. The difference is that the notification is encrypted for all lr-th hop children of N_p, while nodes acting as subscribers had a unique lr-th hop parent. Thus for each children $i \in Chd^{lr}(p)$, N_p uses the ENCRYPT_NOTIFICATION primitive, which takes the event notification (ra, pld) and the lr keys $k_{p,i}...k_{p,Par^{lr-1}(i)}$ shared with its child nodes N_i to $N_{Par^{lr-1}(i)}$. $ENCRYPT\_NOTIFICATION$ then returns:

$$ENCRYPT\_NOTIFICATION(ra, pld, k_{p,i}, ...k_{p,Par^{lr-1}(i)}) = [en_1, en_2, en_3],$$

where:

$$en_1 = ra^{k_{p,i}\dots k_{p,Par^{lr-1}(i)}} \mod q,\ en_2 = pld^{k_{p,i}\dots k_{p,Par^{lr-1}(i)}} \mod q,\ en_3 = N_p.$$

10.5.4.2 SECURE_LOOK_UP

When an intermediate node N_j receives an encrypted event notification, N_j first suppresses an encryption layer in the first two elements of the encrypted notification. The generic

Table 10.3: Processing of event notifications by N_j. In this table, we assume that N_j receives a message $[ra, pld, N_i]$, where ra is an encrypted routable attribute, pld an encrypted content and N_i a node.

1. Remove one encryption layer from ra by using the key $d_{i,j}$:

$$ra \leftarrow \mathcal{D}(q, d_{i,j}, ra).$$

2. Look up ra in RT_j:

   - If ra does not appear in RT_j, break.
   - If ra appears in RT_j:
     (a) If $i = Par^{lr}(j)$, set $l = Par^{lr-1}(j)$, otherwise set $l = i$,
     (b) Remove on encryption layer from pld by using the key $d_{i,j}$:

$$pld \leftarrow \mathcal{D}(q, d_{i,j}, pld).$$

     (c) For each node N_a appearing in the row corresponding to ra in RT_j:
       i. Add an encryption layer to ra and pld by using the key $k_{j,a}$:

$$ra \leftarrow \mathcal{E}(q, k_{j,a}, ra),$$

$$pld \leftarrow \mathcal{E}(q, k_{j,a}, pld),$$

       ii. Send $[ra, pld, N_l]$ to N_b, the child of N_j which is a parent of N_a ($b = Chd(j) \cap \mathcal{N}^{lr}(a)$).

form of a notification received by N_j is indeed:

$$[ra^{k_{l,Par^{lr}(l)} \ldots k_{j,Par^{lr}(j)}} \mod q; pld^{k_{l,Par^{lr}(l)} \ldots k_{j,Par^{lr}(j)}} \mod q],$$

where $l \in Chd^{lr-1}(j)$ and therefore $j = Par^{lr-1}(l)$.

N_j owns the key $d_{j,Par^{lr}(j)}$, and can thus use the decryption operation of the Pohlig-Hellman cryptosystem $\mathcal{D}(q, d_{j,Par^{lr}(j)}, .)$ on the two elements of the encrypted notification to obtain a routable attribute and a payload encrypted with $lr - 1$ layers of encryption:

$$\mathcal{D}(q, d_{j,Par^{lr}(j)}, \underbrace{ra^{k_{l,Par^{lr}(l)} \ldots k_{j,Par^{lr}(j)}}}_{lr \texttt{ layers}}) = \underbrace{ra^{k_{l,Par^{lr}(l)} \ldots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}}_{lr-1 \texttt{ layers}} \mod q$$

$$\mathcal{D}(q, d_{j,Par^{lr}(j)}, \underbrace{pld^{k_{l,Par^{lr}(l)} \ldots k_{j,Par^{lr}(j)}}}_{lr \texttt{ layers}}) = \underbrace{pld^{k_{l,Par^{lr}(l)} \ldots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}}_{lr-1 \texttt{ layers}} \mod q$$

Then, given the routable attribute encrypted $lr-1$ times and the routing table RT_j, $SECURE\_LOOKUP(ra^{k_{l,Par^{lr}(l)}\cdots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}, RT_j)$ returns the list of children nodes where the corresponding message will be forwarded. The look-up in this case simply consists in an equality check between $ra^{k_{l,Par^{lr}(l)}\cdots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}$ and each of the rows of RT_j.

For each corresponding destination, N_a, N_j generates a message by encrypting the routable attribute and the symmetric key with an additional layer of encryption (to maintain the MLCE properties) with the key $k_{a,j}$, thus obtaining:

$$\mathcal{E}(q, k_{a,j}, \underbrace{ra^{k_{l,Par^{lr}(l)}\cdots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}}_{lr-1 \texttt{ layers}}) = \underbrace{ra^{k_{a,j},k_{l,Par^{lr}(l)}\cdots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}}_{lr \texttt{ layers}} \mod q$$

$$\mathcal{E}(q, k_{a,j}, \underbrace{pld^{k_{l,Par^{lr}(l)}\cdots k_{j,Par^{lr}(j)}}}_{lr \texttt{ layers}}) = \underbrace{pld^{k_{a,j}k_{l,Par^{lr}(l)}\cdots k_{Par^{lr-2}(l),Par^{2lr-2}(l)}}}_{lr \texttt{ layers}} \mod q$$

and N_j sends the routable attribute and the payload encrypted lr times to $N_{Par^{lr-2}(l)}$.

Note that in the generic case $Par(Par^{lr-2}(l) = j$ and $Par(a) = l$. The processing of encrypted event notifications is described in Table 10.3.

We illustrate this protocol with an example in the next section for a better understanding.

10.6 An Example

In order to illustrate our solution we define a simple network with one publisher (N_1), and other nodes N_2 to N_{14}. The corresponding tree topology is presented in figure 10.3. We consider the stock-quote market example and we assume that nodes N_4, N_9, N_{10}, N_{12}, and N_{13} subscribe to a common subscription filter ($price = 120$) and that N_1 publishes an event ($price = 120, [symbol = "STM", price = 120, volume = 1000]$). In this example f is ($price = 120$), ra is also ($price = 120$), and the payload pld is $[symbol = "STM", price = 120, volume = 1000]$. We also assume that N_{14} subscribes to a different filter f' which is ($price = 100$).

We set the number of layers $lr = 2$.

In this case, each node shares key pairs with its two hop neighbors namely its parent, grand-parent, children, and grand-children. For example, N_6 shares nine pairs of keys $(k_{6,5}, d_{6,5})$, $(k_{6,3}, d_{6,3})$, $(k_{6,8}, d_{6,8})$, $(k_{6,9}, d_{6,9})$, $(k_{6,10}, d_{6,10})$, $(k_{6,11}, d_{6,11})$, $(k_{6,12}, d_{6,12})$, $(k_{6,13}, d_{6,13})$, $(k_{6,14}, d_{6,14})$, respectively with N_5, N_3, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, and N_{14}.

10.6.1 Propagation of Subscription Filters

Nodes acting as subscribers first encrypt their filters twice with the keys corresponding to their parents and grand-parents and send those encrypted filters to their parents. For example N_{10} sends to N_8 the following:

$$[f^{k_{10,8}k_{10,6}} \mod q; N_{10}].$$

Similarly N_{12} sends to N_9:

$$[f^{k_{12,9}k_{12,6}} \mod q; N_{12}].$$

N_9, N_{13}, and N_{14} send similar messages as well.

When subscription filters are propagated, intermediate nodes remove one encryption layer and build their routing table. For example node N_8 builds a routing table with a unique row that indicates that $f^{k_{10,6}} \mod q$ corresponds to N_{10}.

We take now the example of node N_6 which receives encrypted subscription filters from N_{10} through N_8, from N_{12}, N_{13} and N_{14} through N_9, and from N_9. In this solution, aggregation is not performed directly but after two hops, hence N_6 is able to aggregate the subscriptions of N_{12}, N_{13} and N_9 only. N_{12}, N_{13}, and N_9 subscribe to the same filter f but the encrypted form of the filter is different at each node. *In fine* the routing table RT_6 is represented in Table 10.4.

$f^{k_{9,5}}$	$\rightarrow$	N_9, N_{12}, N_{13}
$f^{k_{8,5}}$	$\rightarrow$	N_{10}
$f'^{k_{9,5}}$	$\rightarrow$	N_{14}

Table 10.4: Routing table RT_6 of N_6

At node N_5 the subscriptions of N_8 and N_9 are aggregated. The routing table RT_5 is presented in Table 10.5. Note that the routing tables are local, they only take into account two hop distances, hence there is no trace at N_5 of the subscriptions of N_{10}, N_{12}, N_{13} or N_{14}, but only of their parents.

$f^{k_{6,3}}$	$\rightarrow$	N_9, N_8
$f'^{k_{6,3}}$	$\rightarrow$	N_9

Table 10.5: Routing table RT_5 of N_5

The routing table RT_3 of N_3 is very similar to RT_5 as can be seen in Table 10.6.

$f^{k_{5,1}}$	$\rightarrow$	N_6
$f'^{k_{5,1}}$	$\rightarrow$	N_6

Table 10.6: Routing table RT_3 of N_3

The process of removing an encryption layer, updating the routing table, adding an encryption layer and forwarding the filter goes on until all nodes receive it. We illustrate the complete propagation of a filter in Table 10.7.

10.6.2 Dissemination of Event Notifications

N_1 wants to notify an event $pld = [symbol = ''STM'', price = 120, volume = 1000]$ with routable attribute $ra = (price = 120)$. For each grand-child it creates a message encrypted twice, once with the key corresponding to this grand-child and once with the parent of this

N_{12}	f
$N_{12} \rightarrow N_9$	$[f^{k_{12,9}k_{12,6}} \mod q; N_{12}]$
N_9	$f^{k_{12,6}} \mod q$
$N_9 \rightarrow N_6$	$[f^{k_{9,5}k_{12,6}} \mod q; N_{12}]$
N_6	$f^{k_{9,5}} \mod q$
$N_6 \rightarrow N_5$	$[f^{k_{9,5}k_{6,3}} \mod q; N_9]$
N_5	$f^{k_{6,3}} \mod q$
$N_5 \rightarrow N_3$	$[f^{k_{5,1}k_{6,3}} \mod q; N_6]$
N_3	$f^{k_{5,1}} \mod q$

Table 10.7: Propagation of a filter f from N_{12} to N_3

child (which is a child of N_1), and then it sends the message to its children. For instance N_1 creates two messages:

- $[ra^{k_{1,2}k_{1,4}} \mod q, pld^{k_{1,2}k_{1,4}} \mod q, N_1]$ which is sent to N_2,
- $[ra^{k_{1,3}k_{1,5}} \mod q, pld^{k_{1,3}k_{1,5}} \mod q, N_1]$ which is sent to N_3.

The last part of the message indicates which key should be used to decrypt the message.

The children of N_1 then remove one encryption layer from the routable attribute and perform secure look-up in their routing table. For example N_3 performs:

$$\mathcal{D}(q, d_{1,3}, ra^{k_{1,3}k_{1,5}}) = ra^{k_{1,5}} \mod q.$$

N_3 then looks up this information in its routing table RT_3 (see Table 10.6). Since $ra = f$, it deduces that the message has to be forwarded to N_6 only (and not N_7). Therefore, N_3 adds an encryption layer with the key $k_{3,6}$ and sends the following message to N_5:

$$[ra^{k_{3,6}k_{1,5}} \mod q, pld^{k_{3,6}k_{1,5}} \mod q, N_1].$$

The process of removing an encryption looking up the routable attribute in the routing table and sending one message per interested node is carried on until the message reaches all interested nodes. Table 10.8 illustrates the propagation of an event notification with routable attribute $ra = f$ on one path from publisher N_1 to node N_{13}. *In fine*, a node which is a subscriber, performs two decryptions to access the symmetric encryption key. For example, N_{13} performs:

$$\mathcal{D}(q, d_{6,13}, \mathcal{D}(q, d_{9,13}, pld^{k_{6,13}k_{9,13}})) = pld \mod q.$$

10.7 Analysis

In this section, we evaluate the security and the performance of the scheme.

Step	Event notification
N_1	$[ra, pld]$
$N_1 \rightarrow N_3$	$[ra^{k_{1,3}k_{1,5}} \mod q, pld^{k_{1,3}k_{1,5}} \mod q, N_1]$
N_3	$[ra^{k_{1,5}} \mod q, pld^{k_{1,5}} \mod q]$
$N_3 \rightarrow N_5$	$[ra^{k_{3,6}k_{1,5}} \mod q, pld^{k_{3,6}k_{1,5}} \mod q, N_1]$
N_5	$[ra^{k_{3,6}} \mod q, pld^{k_{3,6}} \mod q]$
$N_5 \rightarrow N_6$	$[ra^{k_{3,6}k_{5,9}} \mod q, pld^{k_{3,6}k_{5,9}} \mod q, N_3]$
N_6	$[ra^{k_{5,9}} \mod q, pld^{k_{5,9}} \mod q]$
$N_6 \rightarrow N_9$	$[ra^{k_{6,13}k_{5,9}} \mod q, pld^{k_{6,13}k_{5,9}} \mod q, N_5]$
N_9	$[ra^{k_{6,13}} \mod q, pld^{k_{6,13}} \mod q]$
$N_9 \rightarrow N_{13}$	$[ra^{k_{6,13}k_{9,13}} \mod q, pld^{k_{6,13}k_{9,13}} \mod q, N_6]$
N_{13}	$[ra, pld]$

Table 10.8: Evolution of a message published by N_1 on its path to N_{13}

10.7.1 Security Evaluation

We first show that the proposed encryption mechanism with multiple encryption layers ensures confidentiality against external attackers that do not participate to any networking or security operation and further show that it is reaching its privacy goal.

In a work evaluating the security of cryptosystems in the multi-user setting [BBM00], Bellare et al. have essentially shown that if a cryptosystem is secure in the sense of indistinguishability, then the cryptosystem in the multi-user setting, where related messages are encrypted using different keys, is also secure. When a message is encrypted with two independent keys it is at least as secure as any individual encryption. Thus, the scheme is at least as secure as a one layer encryption and external attackers cannot link encrypted messages to the corresponding cleartext.

Furthermore, thanks to the use of multiple encryption layers, the confidentiality of messages relies on the use of keys belonging to different users. Messages are namely forwarded and continuously modified by the addition and removal of encryption layers but they remain unaccessible to intermediate nodes forwarding the message or eavesdroppers at all times. Even if two nodes are subscribing with the same filter they are not able to tell so because each one encrypts it with different keys.

Our protocol hence preserves privacy thanks to secure and efficient routing, moreover it provides the following features:

1. The security of the Pohlig-Hellman cryptosystem is based on the discrete logarithm problem in a finite field of prime order which is hard when the exponent is unknown. Hence we can use the same key several times which simplifies key management.

2. The secure aggregation operation is very simple as well since it is a simple equality test between two filters. The fact that the aggregation takes place after lr hops is a drawback from a performance perspective but it is an advantage from a privacy

perspective as it enables nodes which are neighbors to subscribe to the same filter without discovering that they subscribed to the same filter.

3. Since there is no need for a shared secret, any node can be a subscriber. Nodes for instance can be subscribers and forwarders at the same time and the privacy of all subscribers is still preserved which is a very interesting feature especially in a peer-to-peer environment.

10.7.2 Performance

From a performance perspective, the scheme requires lr Pohlig-Hellman encryptions from subscribers and publishers when sending messages, and lr Pohlig-Hellman decryptions for subscribers receiving a content. The overhead on end-users is thus linear in the parameter lr, and the influence of this parameter is discussed in more details in section 10.7.3.

Concerning intermediate nodes processing messages and forwarding them requires one decryption and one encryption regardless of lr. We did not implement the Pohlig-Hellman cryptosystem, but its processing time can be compared to that of an RSA decryption or signature verification, as both deal with a modular exponentiation. In [TG04], Tillich and Großschädl showed that, on average, an RSA signature implemented on a J2ME phone with a key size of 1937 bits took less than 3 seconds on an Ericsson P900. This might look as an important overhead but the implementation they used was based on a Java edition without optimizations, and the key size choice of 1937 bits is higher than the classical 1024 bits considered as secure until now. Furthermore, their benchmark was performed in 2004: the Ericsson P900 features a PNX4000 156 MHz processor, while nowadays smartphones are equipped with processors exceeding 1 Ghz with hardware accelerators (e.g. the HTC HD2 which features a 1GHz Snapdragon processor). Therefore performing one encryption and one decryption with the Pohlig-Hellman cryptosystem and a key size of 1024 bits would take less than 0.1 seconds on nowadays smartphones.

Hence, even though asymmetric cryptosystems are more expensive than their symmetric counterparts, they can reasonably be used on nowadays mobile devices.

10.7.3 Trade-off Between Performance and Security

As explained earlier, our protocol relies on the use of lr layers of encryption in order to preserve users' privacy. These lr layers of encryption are sufficient to protect against a collusion between $lr - 1$ consecutive nodes, yet if lr nodes in a row decide to collude, they can remove all encryption layers and hence threaten privacy. Our scheme thus allows for a protection against collusion attack by increasing the number of encryption layers as described in [ÖM07]. Therefore, the privacy of the scheme and its resistance to collusion attacks depends on the choice of the number of encryption layers denoted by lr.

The larger values for lr implies a larger number of nodes to collude to break it. However, with large lr, key storage per node becomes a burden and the key distribution overhead can have an impact on the performance of the protocol. Furthermore, aggregation occurs only after lr hops so the larger the parameter lr the less efficient the aggregation mechanism.

Finally choosing a larger lr implies more encryption and decryption operations for the subscribers and the publishers. The choice of lr is hence a trade-off that depends on the scenario and the topology of the network.

If it is possible to assume that nodes do not collude then choosing $lr = 2$ provides optimal performance while protecting user privacy. Such a scenario is possible in a controlled environment with nodes belonging to two different groups (with a conflict of interest) which are interleaved: any path is guaranteed to alternate between a node belonging to the first group and a node belonging to the second one. In opportunistic networks this could be implemented by using trusted communities: if nodes belonging to different trusted communities do not collude, then choosing a path that alternates between various trusted communities protects against collusion attacks and preserves privacy.

If such an assumption does not hold, the presented scheme does not preserve the privacy of all nodes but still protects the network globally. Indeed, if there is no control on the intermediate nodes, an attacker can always bring lr malicious nodes and put them in front of a given node to expose its privacy. So the scheme does not guarantee the privacy of all nodes, but it protects globally the network by increasing the attack cost for the attacker. In challenged and non-controlled environments, having better than nothing security is still a notable achievement.

10.8 Related Work

Publish/subscribe is a messaging paradigm that allows the creation of flexible and scalable distributed systems. SIENA ([CRW01]) is an example of a popular content-based publish subscribe system, but many others have been developed ([Bir93, BCM^{+}99, DGRS03]). Most of the efforts in this area concern pure networking issues, like performance or scalability.

There have been very few attempts at enabling **publish/subscribe systems in mobile networks**. In [HGM01] and [HGM04], Huang and Garcia Molina present a first approach enabling mobility for subscribers and publishers but maintaining a fixed network of brokers. They present also a decentralized approach with a possible extension to MANETs but does not provide a complete solution. Skjelsvik et al. also analyze in [SGP04] the routing issues akin to the design of publish/subscribe in MANETs. In [HGM03], authors describe a complete solution for building optimized publish-subscribe trees in wireless networks in a distributed way. Another approach is proposed by Chen and Schwan in [CS05] which consists in reconfiguring a content distribution overlay based on modification in the physical topology and on brokers' load. Their solution yet requires each broker to be provided with a global view of the network and not only local information. In [DGRS03] where authors propose a solution for peer-to-peer publish/subscribe based on logical directed acyclic graphs instead of relying on a tree structure. In [CP05], Costa and Picco propose a protocol relying on a undirected connected graph where routing is semi-probabilistic: routing is deterministic in the neighborhood of subscribers and probabilistic outside. Other approaches not based on tree structures also include Content-Based Multicast [ZS00] and STEAM [MC02]: both solutions propagate messages only locally (within a maximum distance from the subscriber) and do not rely on any routing structure, hence those approaches do not support content distribution to the whole network. Finally we remind the three approaches CBRHDM [BBQ$^+$05], CBCDM [HG08], and ACBRDTM [CMMP06] presented in section 2.2.3, which explicitly deal with disruptions and high mobility, and are therefore the closest to MobiOpps. However these solutions only perform content-based forwarding and do not propose routing mechanisms as in publish/subscribe.

To the best of our knowledge, **security issues in publish/subscribe systems** have been analyzed only in the fixed network setting. Wang et al. [WCEW02] analyze the security issues and requirements that arise in CBPS systems. They mainly identify classical security problems (like authentication, integrity or confidentiality) and adapt them to the CBPS case. Yet they do not provide concrete or specific solutions to these new problems.

In [OP01], Opyrchal and Prakash focus on the confidentiality issue only on the last leg from end-point brokers to subscribers in a way that is more efficient than group security in terms of key management. Yet, their scheme assumes that brokers are completely trustworthy.

Recently two interesting works concerning confidentiality in CBPS have been published. First, in [RR06], authors focus on notification and subscription confidentiality only. They define the confidentiality issues in a formal model and propose then few solutions depending on the subscription and notification format. Yet they assume that publishers and subscribers share a secret which reduces the decoupling of CBPS, and which means that the solution cannot be adapted to a peer-to-peer setting. Furthermore, in their attacker model, only the brokers are honest-but-curious, the publishers and subscribers are assumed to be trustworthy. This assumption is very strong because the group of publishers and subscribers may be very large. Such a scheme does not protect subscribers' privacy against other curious subscribers for example, let alone against malicious subscribers.

Second, in [SL07], authors propose a specific key management scheme and then a probabilistic multi-path event routing to prevent frequency inferring attacks. In their threat model all nodes (publishers, subscribers and brokers) are assumed to be honest-but-curious. The main weakness of the scheme is the requirement for an online Key Distribution Center (KDC) which is a centralized authority that is trusted not to be curious and decipher all the communication messages. The requirement for this online authority implies that the scheme does not fit an opportunistic network scenario. Concerning content-based event routing, this scheme considers that events have some routable attributes which are tokenized in order to become pseudorandom chains and prevent dictionary attack. Like in [RR06] they adapt the protocol of Song et al. [SWP00] but they do not motivate the use of this particular solution rather than easier and lighter ones. The keys used for tokenizing the routable attributes are derived from the information provided by the KDC which depends on the role of each node, therefore this solution is also not adapted to a peer-to-peer scenario. Furthermore their way of ensuring privacy is through multiple path routing, whereas we protect privacy by cryptographic means.

Finally, in [OPA07], Opyrchal et al. deal with privacy in CBPS, but the focus of the paper is mainly on privacy policy management and not on the design of a cryptographical protocol to achieve it.

Multiple encryption was previously proposed in [GRS96] where authors propose onion routing, to limit a network's vulnerability to traffic analysis. It provides anonymous communication for HTTP through proxies using the RSA commutative encryption scheme. Independently, Pannetrat and Molva use multiple layer encryption in [PM02] for the distribution of confidential data from 1 source to a group of n nodes. This particular algorithm ensures multicast confidentiality and it also prevents the compromise of the whole group whenever a subset of nodes are compromised. In [ÖM07], authors proposed a similar approach for data collection in wireless sensor networks where in this case, n nodes are sending some data to 1 source, the sink. In addition to confidentiality, authors also take the advantage of the inherent homomorphic property in the underlying encryption technique in order to ensure aggregation over encrypted data. Our scheme combines both of these approaches to ensure secure routing and hence subscriber privacy in the $n-to-m$ model akin to CBPS.

Private matching: the underpinning of the secure look-up and secure table building primitives is a matching operation using encrypted data. Private matching has been introduced for equality matches [AES03, LTH04] and extended to more general settings [FNP04, CH08]. Yet a careful study of the problem shows that there is a subtle but important difference between private matching and the requirements of our scheme. Private matching is indeed a two-party protocol between a client and a server where the client learns at the end the information that he shares with the server, whereas in our case the matching operation has to be performed by a third party which has no control over the data.

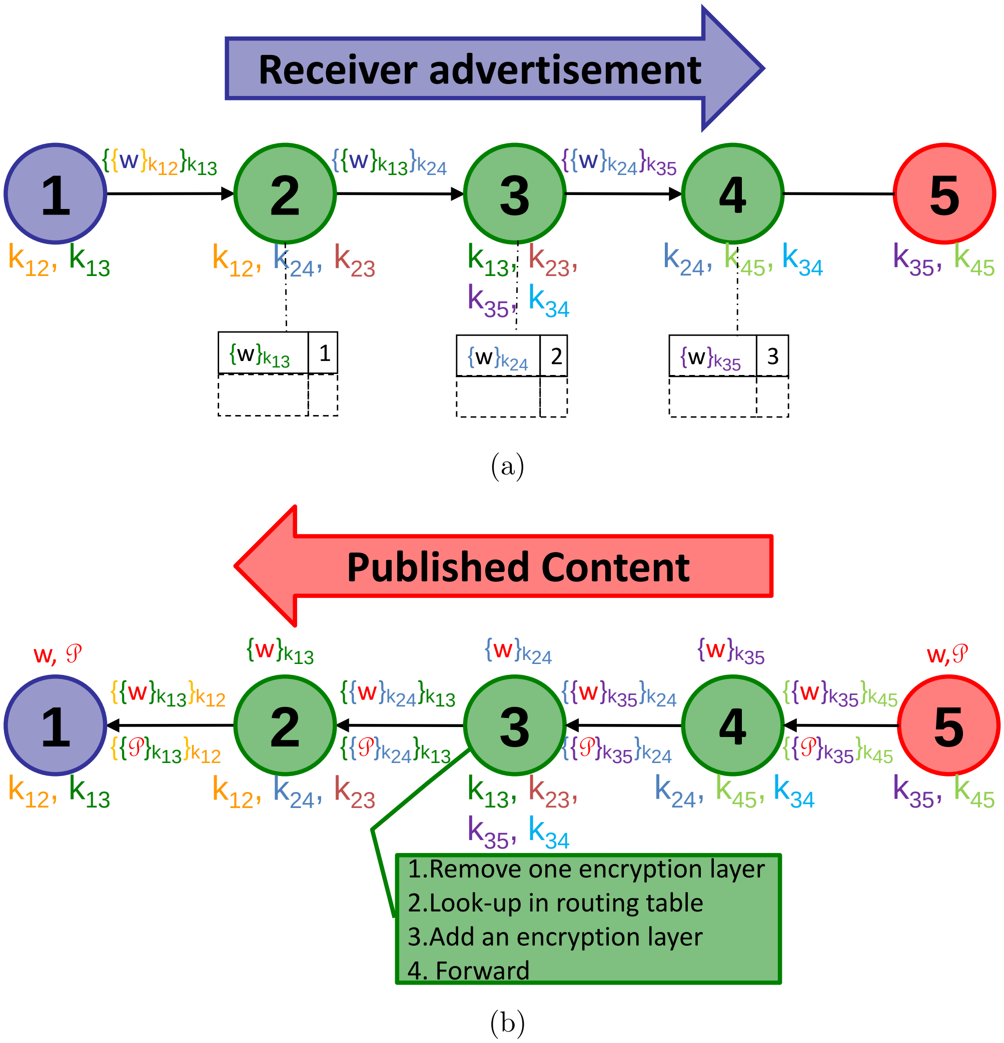
10.9 Conclusion

In this chapter, we analyzed privacy issues in content-based communication. We first analyzed the differences between the classical content-based publish/subscribe paradigm and content-based communication in opportunistic networks. Then, in order to solve the privacy issues with cryptographic tools, we analyzed the link between privacy and confidentiality and identified two confidentiality requirements, namely subscription and information confidentiality. This led us to the more general problem of routing encrypted events using encrypted subscription filters. This problem of secure routing requires two main primitives, namely **building of encrypted routing tables** with aggregation of encrypted filters and **secure look-up** of encrypted events with encrypted routing tables to disseminate the events efficiently. These two primitives have to be designed together with the other classical primitives in order to solve the privacy-preserving routing which had no existing solution.

We then presented a solution to this problem based on multiple layer commutative encryption. MLCE allows brokers to perform secure transformations without having access to the data that is being transferred. Nodes can indeed remove or add an encryption layer without destroying the others and hence perform aggregation, routing tables building or look-up on private data protected by the other layers. Privacy is thus guaranteed among all nodes, including subscribers and eavesdropping outsiders.

Our solution uses the Pohlig-Hellman cryptosystem, and is the first scheme which enables privacy-preserving routing with no shared secret between publishers and subscribers. A key feature of this protocol is that it allows nodes to be subscribers and forwarders (brokers) at the same time while preserving privacy of all nodes as required in the opportunistic network scenario. This protocol can also be tailored to withstand collusion attacks at a certain performance cost.

Key management is a crucial issue for the correctness of this solution: each node N_i has to share Pohlig-Hellman key pairs with each member of its lr-hop neighborhood $\mathcal{N}^{lr}(i)$. Key management needs to be local only but it should be topology-dependent. In this chapter we assumed that nodes had the required keys, but distributing them requires either a central key distribution by an authority which is not appealing in opportunistic networks and defeats the locality property, or the design of a dedicated self-organized local key management solution. The latter is the focus of the next chapter.



(a)

(b)

Figure 10.2: Multiple Layer Commutative Encryption overview: simple example with five nodes and $lr = 2$. Each node shares keys with its one and two hops neighbors (shown below the nodes). (a)Receiver advertisement propagation: a receiver advertisement w is encrypted twice according to the keys shared with the next two hops. Intermediate nodes remove one encryption layer, build their routing tables with data encrypted with one layer, encrypt it again and forward it to the next hop. (b) Published content dissemination: the published content is also encrypted twice with the keys corresponding to the next two hops. The payload $\mathcal{P}$ and the routable attribute w corresponding to the content are encrypted separetly. Intermediate nodes can remove one encryption layer, look-up the result in their forwarding table, then they add an encryption layer and forward the packet to the next hop.

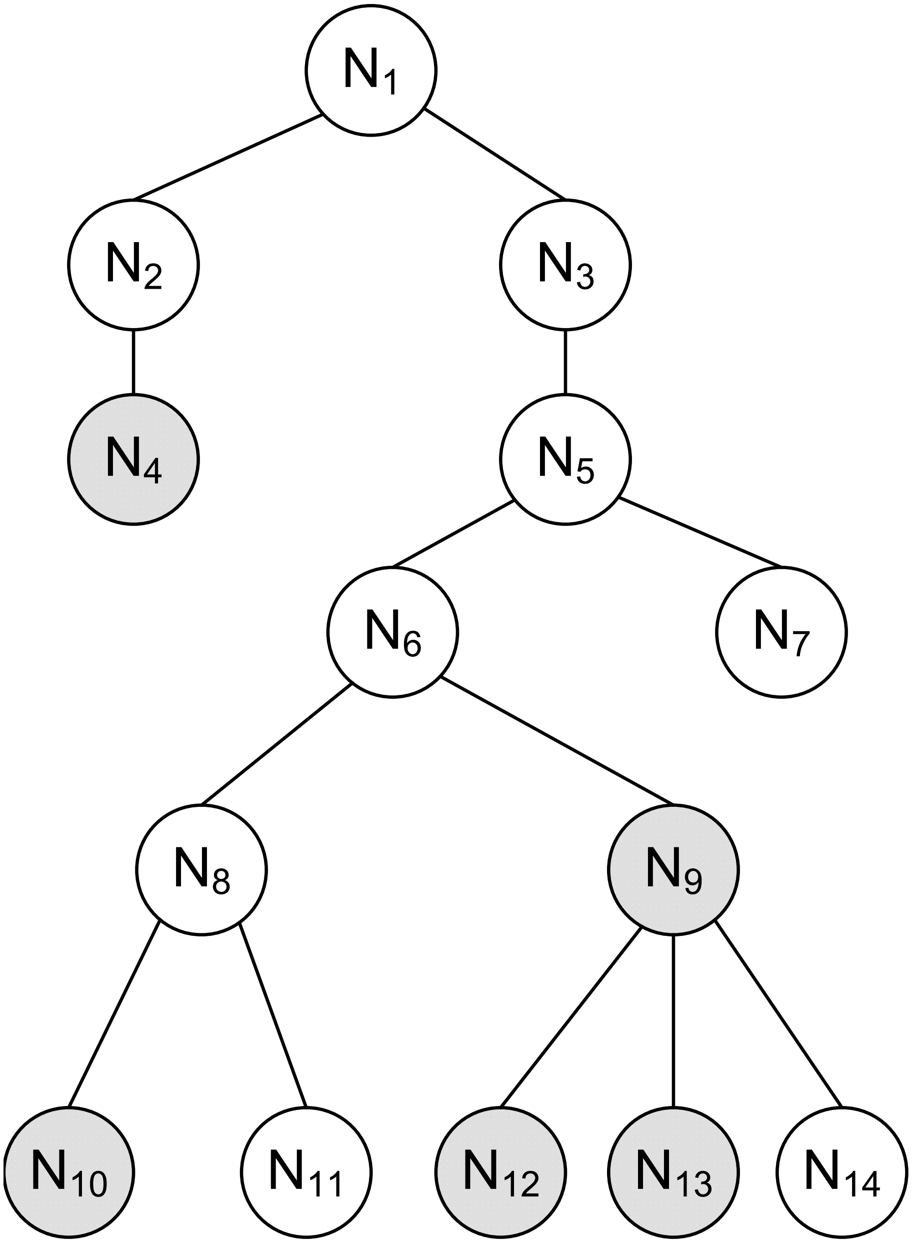


Figure 10.3: Network used as illustration

Chapter 11

Bootstrapping Security Associations in Opportunistic Networks

11.1 Introduction

In the previous chapter, we presented a solution to meet the privacy requirements of content-based forwarding in opportunistic networks. This solution uses of multiple layer commutative encryption (MLCE) and allows to perform secure operations on encrypted content as proposed in [SÖM09a, SÖM09c]. When using MLCE, a node encrypts the data with lr layers of encryption corresponding to the lr next hops. Such a solution therefore calls for an innovative key management scheme that should ensure local and self-organized security associations between a node and its neighborhood: each node should share a key with all its neighbors that are less than lr hops away. The key management should thus depend heavily on the neighborhood topology which is fundamental for the multiple layer encryption scheme to work properly. Because of the lack of infrastructure, this also means that the neighborhood topology itself should be securely discovered.

The main goal of this chapter is therefore to analyze the challenges raised by key management in order to come-up with a dedicated key management solution. This solution should feature local, self-organized and topology-dependent bootstrapping of security associations along with a secure neighborhood discovery. In order to optimize the performance of the scheme, and to cope with the dependency between topology and security, it is indeed more efficient to perform both neighborhood discovery and security associations with all r-hops neighbors together rather than in two separate steps. We achieve this goal by using an authenticated version of Diffie-Hellman key agreement together with encapsulated signatures that protect the integrity of key management messages at each hop. Moreover, since the security of MLCE is directly linked to the number of consecutive colluding nodes, it is important to guarantee that each node can claim only one identity and only one position in the neighborhood. Creation of bogus identities through Sybil attacks would then be a crucial threat against which our scheme is protected thanks an off-line Identity Manager as presented in [SÖM10a].

In this chapter, we first analyze the new security challenges regarding key management in the context of opportunistic networks and extract important requirements for key management in this context. We then present a self-organized and local mechanism that bootstraps security associations with the discovery of the neighborhood topology thanks to the use of certificates and signature chains. The proposed scheme relies on two phases: a first phase where nodes are connected to an Identity Manager that provides them with unique pseudonyms, and a second phase where the opportunistic communication takes place and where there is no need for the Identity Manager.

11.2 Problem Statement

In this section, we define the security requirements of a key management protocol in opportunistic networks and present the threat model tht we consider.

11.2.1 Key Management Requirements

11.2.1.1 Requirements akin to Opportunistic Networks

Key management in opportunistic networks is a challenging task. The lack of end-to-end connectivity underpinning opportunistic networks has indeed strong implications on the problem of key management. For instance, nodes cannot agree on end-to-end keys nor rely on an online: key agreement can only be local. Furthermore, online centralized authority or security server cannot be used if end-to-end connectivity cannot be assumed. This implies in particular that public key encryption is not suitable to opportunistic communication as it requires an online Public Key Infrastructure that generates and manages the public key certificates.

Key management for identity-based cryptography is more adapted to opportunistic networks as it only requires an offline Public Key Generator. Therefore identity-based cryptography is generally a good candidate for opportunistic networks because they do not require certificates (and they are used by Asokan et al. in [AKK^{+}07] in this context). However, identity-based cryptographic tools are not suitable for content-based forwarding, whereby messages are forwarded depending on their content and the interests advertised by nodes, therefore the (set of) destination is unknown at the source.

A suitable key management solution for content-based communication in opportunistic networks should thus be local and self-organized and should not depend on the identities of the nodes.

11.2.1.2 Specific Requirements of the MLCE Solution

The security of MLCE strongly depends on the location of the nodes in the topology. Indeed, nodes need to establish security associations in the form of pairwise keys with all nodes that are at most lr hops away. Given the layered structure of MLCE, the assurance of privacy strongly depends on the position of nodes in terms of hop-distance: the key agreement scheme should therefore be bootstrapped on the topology of the neighborhood. Neighborhood topology needs also to be discovered because of the lack of infrastructure.

In the previous chapter, we assumed indeed that the tree topology overlay used for content-based communication was generated in a local and decentralized way. This implies that each node is aware of its lr-hop neighborhood topology. Securely discovering the neighborhood topology is yet a non-trivial task which in turn requires security services because nodes should guarantee their claimed hop-distance to their neighbors and should not claim fake distances which would have an impact on the security of MLCE. Classical solutions to guarantee the hop-distance for more than one-hop ([HPJ05a, HJP02a]) use cryptographic mechanisms and assume that nodes already own verifiable keying materials (e.g. identity certificates).

Hence there is a cyclic dependency between secure neighborhood discovery and key management in MLCE similar to the dependency cycle between secure routing and security services in MANETs identified by Bobba et al. in [BEGA03]. In order to take into account this dependency between network topology and security, and in order to avoid running two separate protocols, one for neighborhood discovery and one for local key management, security associations should thus be locally bootstrapped along with a lightweight neighborhood discovery solution.

11.2.2 Threats

11.2.2.1 Generic Attacks

In order to bootstrap security associations and discover the neighborhood topology each node should launch a dedicated communication protocol. Thus, as with the design of any communication protocol, the key management protocol should consider the regular attacks which can be classified as follows:

- **Passive attacks**: malicious nodes only eavesdrop on communication; they do not take part in the forwarding process and therefore can only discover the content of the packets if those are not protected. Therefore protocol messages should be encrypted in order to prevent such attacks.
- **Active attacks**: malicious nodes can either modify packets or launch replay or man-in-the-middle attacks. In the particular case of key management in MLCE, the goal of active attackers would be to discover a key by establishing security associations with a legitimate node without complying with the local topology. The impact of pollution or other kind of attacks where nodes only aim at disrupting the protocol without gaining any advantage, are not analyzed in this thesis.

11.2.2.2 Sybil Attack

In addition to classical attacks, the key management protocol should take into account the attacks specific to MLCE. The security of MLCE is indeed based on the parameter lr and if lr consecutive nodes collude they can break the MLCE scheme.

Thus, if nodes can launch Sybil attacks [Dou02] by simulating many different identities claiming different hop distances they can weaken the security of the MLCE scheme. Indeed, in this case one single node (the malicious node) simulating lr identities and claiming different positions for each identity would receive one key per layer and would therefore easily decrypt the content of packets although it does not have the right to. Hence, a node should only have a unique unspoofable identity (pseudonym) and a global mechanism of identity management has to be defined.

To summarize, content-based opportunistic networking requires a local and self-organized key management mechanism. Nodes should establish key pairs with all nodes which are at most lr hops away. Moreover, nodes should also be able to determine the position of each node in order to achieve the security goals of MLCE, and therefore security associations should be bootstrapped along with neighborhood discovery. Finally, as with any regular protocol, the new key management protocol should be protected from regular network attacks.

11.3 Proposed Solution

In order to meet the requirements detailed in the previous section , we propose a solution for bootstrapping security associations which features two phases. Indeed, nodes require anchors to be uniquely identified in the network, and each node should have only one valid anchor to prevent Sybil attacks. Therefore, we propose first a setup phase, during which nodes are connected to an Identity Manager (IM) that generates and distributes these anchors in the form of certificates. The keying material received during this phase can be considered as long-term keying material that allows the computation of short-term keys resulting from the establishment of security associations in a secure way.

During the regular network operations, nodes do not need to communicate with the Identity Manager anymore and the long term keys are not used by the application. We hereafter describe these two phases in detail.

11.3.1 Setup Phase

During the setup phase, nodes contact an IM, which is a lightweight security server that generates pseudonyms and certificates on-the-fly but does not manage certificates as in classical public key infrastructures. For the sake of clarity, we assume the existence of a single Identity Manager (IM), but the infrastructure could be more sophisticated with a distributed architecture for example. The IM generates a public/private key pair pk_{IM}/sk_{IM}, and pk_{IM} is known by all nodes. The role of the IM is twofolds:

1. **Enforcing privacy:** The IM first provides nodes with pseudonyms in order to enforce privacy. In opportunistic networks real identities are indeed meaningless. Hence, using actual identities only incurs a privacy threat with no additional advantage over pseudonyms.

2. **Prevention of Sybil attacks:** The IM links the pseudonym to a real identity and a public/private key pair and certifies it. Indeed, even though identities are meaningless, nodes should be restrained to a unique pseudonym otherwise they could have several identities, which would lead to Sybil attacks. If a node could impersonate other nodes or simply produce several identities for himself, it could pretend to be at several positions at the same time, and therefore break the multi-layer scheme.

To fulfill these tasks, each node N_i first generates a public/private key pair pk_i/sk_i and then sends pk_i to the IM. The IM first verifies that N_i owns the associated private key with a challenge-response exchange, and then requests the node for some information I_i to uniquely identify N_i. The requested set of information remains the same for all nodes at anytime (e.g. full name, date and place of birth) and is thoroughly verified by the IM (with the help of official documents like ID card or passport for example). The IM uses this set of information I_i together with a master key K (known only by the IM) in a message authentication code (MAC) function to generate a pseudonym for the node:

$$\mathcal{P}_i = MAC(K, I_i).$$

We assume that the MAC function used is hiding, which means that the MAC does not reveal any information about the authenticated message. In other words, $\mathcal{P}_i$ does not leak information with respect to I_i.

The IM then provides N_i with a certificate $\mathcal{C}_i$ which links the public key of N_i with its pseudonym, by signing these information:

$$\mathcal{C}_i = \{\mathcal{P}_i, pk_i, signature_{sk_{IM}}(\mathcal{P}_i, pk_i)\}.$$

The information exchange protocol between the IM and a node N_i is presented in figure 11.1. Note that a node can obtain several certificates with different public keys, but all the certificates include the same pseudonym and can therefore not be used for Sybil attacks.

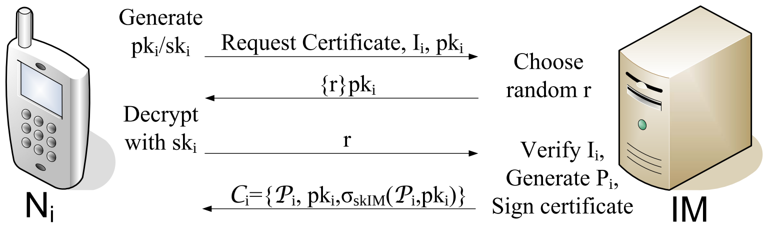

Generate pk_i/sk_i
Request Certificate, I_i, pk_i
Choose random r
$\{r\}pk_i$
Decrypt with sk_i
r
Verify I_i, Generate P_i, Sign certificate
$\mathcal{C}_i=\{\mathcal{P}_i, pk_i, \sigma_{skIM}(\mathcal{P}_i, pk_i)\}$
N_i
IM

Figure 11.1: Summary of the information exchange protocol with the IM.

This certification process, ensures that each node has only one pseudonym, and the corresponding certificate can be used to prove that this pseudonym was generated by the IM and is not random. Therefore, the use of this certificate effectively prevents Sybil attacks.

When the node N_i has retrieved its certificate $\mathcal{C}_i$, the setup phase ends and N_i can enter the runtime phase. During the runtime phase, communication is supposed to be delay-tolerant, therefore the IM is unreachable and secure communication should be possible without accessing the IM.

11.3.2 Bootstrapping Local Security Associations

We now assume that all nodes have already performed the setup phase and owns a pseudonym certificate as mentioned in the previous section.

During the second phase nodes need to establish ephemeral security associations by sharing keys with all their neighbors which are at distance less than lr hops. As mentioned previously, this key agreement depends on the local topology and therefore requires a secure neighborhood discovery. In order to optimize the number of message exchanges and to cope with the dependency between security and topology, we propose a local key agreement protocol along with neighborhood discovery: one protocol run provides the initiator with both a correct view of its neighborhood topology at lr hops distance and shared secrets with all lr-hops or less neighbors in a batch. On the one hand, the neighborhood discovery mechanism is inspired by secure routing protocols (like [HPJ05b]) with the noticeable difference that our solution is based on a hop count limit instead of targeting a destination: it therefore relies on signature chains to guarantee the integrity of the discovered topology. Contrary to secure routing in MANET, the goal of our protocol is not to perform end-to-end secure routing which is irrelevant in opportunistic networks, but simply to discover the local topology of the network. On the other hand, the key agreement scheme is derived from an authenticated version of Diffie-Hellman key agreement protocol, also called the station to station protocol [DVOW92]. We therefore assume that all nodes know a group G with generator g suitable for a Diffie-Hellman protocol. Furthermore, all exponentiations are taken modulo the cardinal of the group $|G|$ and we do not mention this modular extraction in the sequel of the chapter for the sake of clarity.

The protocol features four main steps. First a node initiates a Security Association Request for lr hops, this request is then forwarded to neighbors until the lr-th hop receives it. Then, a Security Association Reply is sent to the initiator through the reverse path of the request and finally the initiator can compute the shared keys. These four steps are detailed hereafter and an example of the execution of the protocol over one path is given in table 11.1.

11.3.2.1 Initiation of Security Association Request

When a node N_s wants to establish security associations with its neighbors, at distance less than lr hops, it needs to initiate a Security Association Request. It first chooses a random $r_s \in \mathbb{Z}^+_{|G|}$ and computes its Diffie-Hellman share g^{r_s} in order to establish short term keys with each of the neighbors. In order to prevent impersonation, N_s should also send its certificate received from IM during the previous phase. Finally, since the Security Association Request should not be forwarded after the lr-th hop, an additional iterator

Table 11.1: Example of Security Association bootstrapping. The initiator N_1 discovers its 3-hop neighborhood and establishes security associations with three nodes. The underlined font indicates changed message fields, relative to the previous message of the same type.

	N_1 **initiates Security Association Request**
N_1	randomly chooses $r_1 \in \mathbb{Z}^+_{\lvert G \rvert}$
	$\sigma_1 = signature_{sk_1}(SARq, 3, \{\mathcal{C}_1\}, \{g^{r_1}\}, \{\})$
$N_1 \rightarrow *$	$< SARq, 3, \{\mathcal{C}_1\}, \{g^{r_1}\}, \{\sigma_1\} >$
	Processing of Security Association Request by intermediate nodes
N_2	verifies σ_1 and randomly chooses $r_2 \in \mathbb{Z}^+_{\lvert G \rvert}$ and ρ_2
	$\sigma_2 = signature_{\underline{sk_2}}(SARq, \underline{2}, \{\mathcal{C}_1, \underline{\mathcal{C}_2}\}, \{g^{r_1}, \underline{g^{r_2}}\}, \{\underline{\sigma_1}\}, \underline{\rho_2})$
$N_2 \rightarrow *$	$< SARq, \underline{2}, \{\mathcal{C}_1, \underline{\mathcal{C}_2}\}, \{g^{r_1}, \underline{g^{r_2}}\}, \{\sigma_1, \underline{\sigma_2}\} >$
N_3	verifies σ_1 and randomly chooses $r_3 \in \mathbb{Z}^+_{\lvert G \rvert}$ and ρ_3
	$\sigma_3 = signature_{\underline{sk_3}}(SARq, \underline{1}, \{\mathcal{C}_1, \mathcal{C}_2, \underline{\mathcal{C}_3}\}, \{g^{r_1}, g^{r_2}, \underline{g^{r_3}}\}, \{\sigma_1, \underline{\sigma_2}\}, \underline{\rho_3})$
$N_3 \rightarrow *$	$< SARq, \underline{1}, \{\mathcal{C}_1, \mathcal{C}_2, \underline{\mathcal{C}_3}\}, \{g^{r_1}, g^{r_2}, \underline{g^{r_3}}\}, \{\sigma_1, \sigma_2, \underline{\sigma_3}\} >$
N_4	verifies σ_1 and randomly chooses $r_4 \in \mathbb{Z}^+_{\lvert G \rvert}$ and ρ_4
	$\sigma_4 = signature_{\underline{sk_4}}(SARq, \underline{0}, \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \underline{\mathcal{C}_4}\}, \{g^{r_1}, g^{r_2}, g^{r_3}, \underline{g^{r_4}}\}, \{\sigma_1, \sigma_2, \underline{\sigma_3}\}, \underline{\rho_4})$
	Security Association Reply($remaining\_hop\_count = 0$)
$N_4 \rightarrow N_3$	$< SARp, \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}, \{g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}\}, \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}, \{\rho_4\} >$
$N_3 \rightarrow N_2$	$< SARp, \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}, \{g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}\}, \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}, \{\rho_4, \underline{\rho_3}\} >$
$N_2 \rightarrow N_1$	$< SARp, \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}, \{g^{r_1}, g^{r_2}, g^{r_3}, g^{r_4}\}, \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}, \{\rho_4, \rho_3, \underline{\rho_2}\} >$
	Key computation
N_1	Verify the validity of the reply Shared keys : $g^{r_1 r_2}$ with N_2, $g^{r_1 r_3}$ with N_3 and $g^{r_1 r_4}$ with N_4 One established 3-hop path : N_2, N_3, N_4

should be included in the message and should be decremented at each hop. N_s signs all these information to prove their authenticity and broadcast the following message:

$$< SARq, lr, \mathcal{C}_s, g^{r_s}, \sigma_s > .$$

$SARq$ is just an identifier standing for Security Association Request and σ_s is a signature of the whole message with the private key sk_s, to be more precise

$$\sigma_s = signature_{sk_s}(SARq, lr, \mathcal{C}_s, g^{r_s}).$$

11.3.2.2 Processing of Security Association Requests

Upon receiving a Security Association Request, an intermediate node N_i first verifies the authenticity of the initial message by using the public key of N_s. N_i builds on the received message by adding its certificate and by decrementing the *remaining_hop_count* iterator. Then, as N_s, N_i generates a Diffie-Hellman share and includes it in the message. Finally, N_i signs the modified message: this produces a sequence of encapsulated signatures which validates the integrity of the message at each step. Thus, the general form of a Security Association Request contains three lists gradually filled in by intermediate nodes:

$$< SARq, remaining\_hop\_count, Certificate\_list, \\ DH\_share\_list, signature\_list > .$$

To be more precise, N_i first checks the authenticity of the initial request message by verifying the signature of the initiator. To do so, it reconstructs the initial request message which is:

$$< SARq, r, first(Certificate\_list), first(DH\_share\_list), \\ first(signature\_list) >$$

where first(.) designates the first element in a list. lr is computed as the addition of *remaining_hop_count* and the number of elements in the lists minus one. Then, the initial signature $first(signature\_list)$ is checked thanks to the public key of the initiator which can be found in $first(Certificate\_list)$.

If the signature is valid, the intermediate node N_i processes the request as follows:

- *remaining_hop_count* is decreased by one,
- N_i appends its own certificate $\mathcal{C}_i$ to $Certificate\_list$ in order to give a proof of its pseudonym $\mathcal{P}_i$ and to provide its public key pk_i,
- N_i needs to provide a Diffie-Hellman share for the key agreement, hence N_i draws a random number r_i and then appends g^{r_i} to DH_share_list,
- N_i needs to prove the integrity and authenticity of the modified request therefore it computes a signature σ_i of the modified message plus a random number ρ_i:

$$\sigma_i = signature_{sk_i}(ND, remaining\_hop\_count, \\ Certificate\_list, DH\_share\_list, \rho_i)$$

and appends σ_i to *signature_list.*

ρ_i is a random number that is revealed in the Security Association Reply as described in the next section. Indeed, in order to verify the authenticity of the path, the reply message should follow the same path in the reverse direction. Therefore, in addition to their

Diffie-Hellman shares, each node also generates a random number ρ_i kept secret, before signing the message. This random number guarantees that the reply returns through N_i: if the reply do not pass through N_i then σ_i cannot be verified and therefore the message is considered as not valid. We assume that the signature scheme does not leak any information about the signed message, therefore it is impossible to deduce the value ρ_i only from the signature σ_i.

After this processing, the message is broadcasted, except if the message reached the lr-th hop.

11.3.2.3 Security Association Reply

The reply has to follow the reverse path from which the discovery request has been forwarded, therefore the iterator is no longer needed. The reply mainly consists of the list of certificates, signatures and Diffie-Hellman shares at the last hop of the request. Furthermore, intermediate nodes N_i that receive back the reply, need to reveal the random number ρ_i they used in the request to allow the verification of their signature. Therefore the general format of the reply is:

$$\begin{aligned}< SARp, Certificate\_list, DH\_share\_list, signature\_list,&\\ random\_number\_list > .&\end{aligned}$$

$SARp$ is an identifier for the reply and $random\_number\_list$ corresponds to the list of random numbers used during the signatures of request messages.

The processing of reply messages by intermediate nodes is simple. Upon receiving a reply message, an intermediate node N_i first checks that it was on the request path, by looking for its own certificate $\mathcal{C}_i$ in $Certificate\_list$ and then appends the random number ρ_i it chose to $random\_number\_list$. Then N_i forwards the message to the next hop as listed in the $Certificate\_list$.

11.3.2.4 Key Computation

When the reply finally gets back to the initiator of the neighborhood discovery N_s, N_s thoroughly verifies its validity by checking that:

1. the number of elements in $Certificate\_list$, DH_share_list, $signature\_list$ is equal to $lr + 1$ while the number of elements of $random\_number\_list$ is equal to lr,

2. all the certificates in $Certificate\_list$ are related to different users (the pseudonyms should all be different) and valid (the signature of the IM on each certificate should be valid),

3. all the signatures in $signature\_list$ are valid. To do so, the initiator reconstructs the message at each hop and verifies the validity of the signature at each step by taking into account the corresponding random number listed in $random\_number\_list$.

If all these verifications succeed, N_s and the neighbors listed in the message compute their shared keys. The key shared with N_i is computed as $(g^{r_i})^{r_s}$ by the initiator and as $(g^{r_s})^{r_i}$ by N_i. N_s also knows of one lr-hop path in its neighborhood.

Note that, for one Security Association Request, the initiator should receive many replies, one per possible lr-hop path. Thanks to this mechanism, the initiator can fully construct its lr-hop neighborhood topology and establish security associations with all the nodes in this neighborhood.

11.3.3 Summary

The complete mechanism enables to bootstrap security associations along with neighborhood discovery in opportunistic networks: each reply results in the initiator knowing one lr-hop path and sharing keys with all the nodes on this path. With all the replies, the initiator can thus securely construct the topology of its lr-hop neighborhood. The proposed mechanism is local and self-organized and therefore complies with the delay-tolerant nature of opportunistic networks.

The mechanism relies on two phases: a setup phase where nodes have access to the IM and the runtime phase where the opportunistic communication actually takes place.

11.4 Evaluation

In this section we evaluate the security and performance of the proposed scheme.

11.4.1 Evaluation of the Setup Phase

This setup phase, whereby nodes communicate with the Identity Manager in order to get pseudonym certificates, protects the proposed mechanism against Sybil attacks. Indeed, since the pseudonym of a node is strongly linked with its real identity, nodes can only have one pseudonym, and malicious nodes cannot simulate multiple identities. Hence malicious nodes cannot share several keys corresponding to different distances with respect to a given node and thus cannot access any private message they are not authorized to.

The proposed IM has a completely different role than classical Certification authorities. The role of the IM is not to certify identities, it just certifies that a given node has one and only one pseudonym. The IM is lightweight by design because it does not need to keep track of the certificates it delivered. Each time a node asks for a certificate, the IM generates the associated pseudonym on-the-fly by requesting the same information, and the resulting pseudonym is always the same for a given node. During networking operations, the Identity Manager is not required anymore and the proposed scheme enables local and self-organized security associations.

11.4.2 Analysis of the security association mechanism

One of the main goal of the scheme is to bootstrap security associations between nodes which are less than lr hops away. In this section we analyze the security aspects of this feature.

11.4.2.1 Protection against Passive Attackers

Since the establishment of security associations is simply based on the Diffie-Hellman exchange protocol, eavesdropping is inherently prevented thanks to the hardness of the Discrete Logarithm and the Computational Diffie-Hellman Problems [Sti95]. Indeed, since given g^{r_1}, it is difficult to retrieve r_1 and given g, g^{r_1}, g^{r_2} it is difficult to compute $g^{r_1 r_2}$, key shares can be sent in clear and an adversary node cannot discover the key resulting from the association. Therefore, the security of the scheme against passive attackers results directly from the security of the Diffie-Hellman protocol.

11.4.2.2 Protection against Man-In-the-Middle Attack

Since the message exchange is not performed by only two nodes, the security guarantee offered by the Diffie-Hellman protocol is not sufficient, especially in the presence of active attackers. The first type of attacks that can be launched by an active attacker is man-in-the-middle attacks. Such attacks are effectively prevented by the use of an authenticated version of the Diffie-Hellman exchange protocol that adds signatures computed over key shares. Indeed, no node can forge a network discovery request initiated by node N_s because it requires the private key of N_s.

11.4.2.3 Incidence of Replay Attacks

An authentic request by N_S can still be replayed by a malicious node. However, a malicious node which replays a neighborhood discovery request cannot discover a shared key with other nodes because it does not know the random number r_s. Furthermore, since nodes still answer several identical requests by processing them the same way (and by using the same Diffie-Hellman share), this does not create false security associations, therefore this attack is not critical from a security perspective.

11.4.2.4 On the Modification of the STS Protocol

As explained in section 11.3.2, our protocol for establishing security associations is a modified version of the STS protocol [DVOW92]. The modifications with respect to the original STS protocol are twofolds:

- in the STS protocol, messages are signed and then encrypted with the shared key, whereas in our protocol we remove this encryption process,

- our protocol is composed of two message exchanges, whereas the original STS protocol requires a third message exchange whereby the originator signs both its share and the share of the other party.

Concerning the second point, we adopted this design for performance reason. It is possible to stick to the STS original version and add a third message sent by the originator back to the lr next hops in which the shares of the other nodes are signed by the originator. Yet the only additional security offered by this step is a protection against replay attack, and the incidence of this attack is minor as discussed in section 11.4.2.3.

Concerning the first point, Diffie et al. used the encryption with the shared key $g^{r_s r_i}$ as a proof of knowledge of $g^{r_s r_i}$. In a more recent work [Kra03], Krawczyk et al. showed that using encryption as a proof of knowledge is insecure and can lead to a misbinding attack: an attacker could lead N_s and N_i to share a key $g^{r_s r_i}$ but N_s would think that the key $g^{r_s r_i}$ is shared with the attacker instead of N_i. The attacker still does not get knowledge of the key $g^{r_s r_i}$, but this could still be problematic in critical scenarios such as banking (money could be credited the attacker instead of N_i). Krawczyk proposed an alternative scheme called SIGMA (for SIGn and MAc) [Kra03] to remove this flaw. In fact, all these issues are pertaining to the authentication part of the STS protocol. The goal of STS or SIGMA is to authenticate the entities N_s and N_i, and at the same time to share a key between these entities, thus binding the key with an identity.

In our protocol, this strong authentication mechanism is not required. The goal for N_s is to share keys with the lr next hops, but the identity of these nodes has no importance for N_s. The use of the authenticated version of Diffie-Hellman is only justified by the fact that N_s needs to make sure that the lr exchanged keys correspond to lr different nodes, the actual identity of those nodes making no difference (and pseudonyms are used to prevent linking a node to its real identity anyway). In case of a misbinding attack (as defined in [Kra03]) on our protocol, the attacker still needs to add its certificate, and prove that it is a different node from the others in the chain. This is the only relevant information for N_s as, there is no trust relationship implied by our protocol beyond the fact that the lr nodes are different, contrary to the general case targeted by STS and SIGMA.

The modifications that we brought to the STS protocol, while insecure for binding authenticated entities with a shared key, offer the right security for our protocol and provide a performance increase.

11.4.3 Evaluation of the Neighborhood Discovery Mechanism

We now analyze the security of the second main goal of the proposed scheme, which is to securely discover the neighborhood topology.

The mechanism of encapsulated signatures prevents most basic active attacks, and makes tampering of Security Association messages difficult:

- the mechanism of encapsulated signatures in security association requests protects the integrity of messages at each step. Therefore an intermediate node cannot forge the message of a previous node, in particular it cannot change the value of an iterator, nor can it modify the value of the Diffie-Hellman share. An intermediate node can only undo some steps to remove some nodes from the path and extend the neighborhood discovery hops in a grayhole attempt [HPJ05a]; i.e. by selectively dropping some messages or by removing some elements in the lists of the security association request message. But in this case the deleted nodes will not accept to forward the reply because their certificates are not in the certificate list anymore. To be successful this attack thus requires a way to circumvent the deleted nodes and in this case it is a wormhole and not a grayhole attack anymore.

- the mechanism also ensures that the path of the reply is the reverse of the request thanks to the use of the random numbers ρ_i. Indeed the signatures in the request messages cannot be verified if the ρ_i are not revealed and nodes only reveal them in reply messages if they were involved in the request path. An alternate solution would be to sign all the reply messages, but this would be more costly.

Wormhole attacks [HPJ03] that completely circumvent the deleted nodes and avoid message discarding can be successful and the source node would end up with a fake neighborhood topology in that it would contain nodes which are more than lr-hops away. The impact of this attack is however the same as the collusion attack in MLCE: if lr consecutive nodes collude they can break the scheme and access encrypted messages. Hence, it is possible to mitigate this attack by increasing the security parameter lr, which is chosen according to the expected maximum number of consecutive malicious nodes. Furthermore, we assume that nodes can securely determine their one-hop neighbors by using distance bounding techniques ([CH06, SPR+09]), which further mitigates the wormhole threat.

11.4.4 Performance Evaluation

The scheme requires asymmetric cryptography and signature computations to guarantee the local neighborhood topology. Nevertheless, the design of the mechanism takes into account the need to minimize the number of signatures and increase its performance. The use of the random numbers ρ_i avoids signing both requests and replies, and enables the signature of requests only, thus decreasing both the computation and communication overhead: intermediate nodes have to verify and to compute only one signature each, while the initiator has to verify lr signatures only. Signature verification is much more efficient than signature generation. The message length is roughly the size of the three main lists $Certificate\_list$,DH_share_list, $signature\_list$ which contain at most $r+1$ elements each, and in each of these elements the most important component is the public key. The message length is therefore linear in the number of hops lr.

It is possible to settle a trade-off between computation time, message length and security level by choosing between RSA signatures and elliptic curve signatures (ECDSA [ECD05]). In [TG04], Tillich and Großschädl compare the execution time of RSA and ECDSA signatures on various mobile phones. As explained in section 10.7.2, the devices they use are largely outperformed by nowadays smartphones, but the comparison they make is still useful. In particular it shows that, for equivalent security levels, ECDSA is more efficient than RSA with respect to signature generation, but the opposite holds for signature verification. Furthermore, the signature is shorter with ECDSA than with RSA. By choosing ECDSA signatures, the communication overhead is reduced and the computation load mainly affects the initiator (because the signature verification is more costly than its generation), while RSA distributes the computation overhead on all nodes involved in the protocol and implies a higher communication overhead. Therefore, ECDSA is more adapted to our protocol as it implies a message size and a fairer distribution of computation overhead.

It is worth noticing that the proposed protocol is not used for routing, but to bootstrap security associations from scratch. The proposed scheme can therefore be used as an anchor for further efficient key management based on these security associations. Using asymmetric cryptography to bootstrap security associations is a widely accepted concept, hence performance is not a critical issue for the proposed mechanism.

11.5 Related work

11.5.1 Key Management in Ad-Hoc Networks

The area of key management in opportunistic networking is quite new and the existing work in this area are rare: in [Far07], Farrell mentions some requirements of key management in DTN but no solution is proposed, and in [AKK$^+$07] Asokan et al. evaluate ID-based cryptography in the context of DTN, but this solution is not suitable for content-based forwarding as mentioned in section 11.2.1. In the broader area of peer-to-peer key management in mobile ad hoc networks (MANETs) many solutions have been proposed ([MDM07]). These solutions can be classified in two main categories:

- fully self-organized key management, which have been first proposed by Capkun et al. in [CBH03], and further studied in [CCH06, CHB06, MPR09]. These solutions require no authority, and are based on self-certificates (PGP-like) which are then used to sign other trusted nodes' certificates to form chains of trust. Key management therefore requires high-mobility to efficiently establish the chains of trust. Unfortunately, trust establishment is a time consuming operation. Furthermore, such fully organized schemes are inherently vulnerable against Sybil attacks, which is a major issue for MLCE (see section 11.2.1). Therefore fully self-organized key management cannot fit to our problem.

- authority-based solutions, rely on an external authority to bootstrap trust relations from certificates signed by the authority. In addition, most of them make use of an online authority with the accent on distributing this online authority either partially ([KKA03, WWF$^+$07, XI04, YK02, ZH99]) or fully ([KZL$^+$01, LZK$^+$02, JNP05]). All these approaches are based on threshold cryptography and require each certificate to be signed more than once online and therefore they are not suited to our problem either.

An important difference between all these solutions and our proposal is that key management in MANETs aims at establishing end-to-end keys whereas this is irrelevant in opportunistic networks. It is therefore hard to compare these solutions with ours, but we can tentatively say that our solution is in between the two mentioned categories: it makes use of an offline authority to prevent Sybil attacks, but online key agreement is self-organized and does not require an additional online authority, therefore it meets the DTN requirements.

11.5.2 Secure Neighborhood Discovery

Secure neighborhood discovery amounts to secure routing with a fixed number of hops instead of a given destination. Most existing secure routing solutions for MANETs (Ariadne [HPJ05b], SEAD [HJP02b], SRP [PH03])implicitly assume the existence of pre-established trust relationship among nodes wishing to communicate with each other (like prior shared keys or an authentic TESLA [PCTS02] key chain). Establishing such trust relationship requires a secure distribution scheme, which requires either an online central authority or a secure routing which is the goal of these schemes.

Hence, there is a cyclic dependency between secure routing and security services which was first analyzed by Bobba et al. in [BEGA03]. The authors propose to break the cycle dependency by using a secure binding mechanism between an IP address and an uncertified public-private key pair, which results in a statistically unique and unspoofable IP address. Their solution cannot prevent Sybil attacks yet and therefore it is not suited to our problem.

In contrast to these solutions, our solution breaks the dependency cycle and prevents Sybil attacks, by doing at the same time key agreement and neighborhood discovery securely thanks to certificates with unique pseudonyms provided by an offline Identity Manager. Our approach is therefore close to ARAN ([SDL$^+$02]) with the noticeable difference that ARAN certificates are used to certify an IP address which is dynamic and therefore this implicitly requires that the Certification Authority be online. Furthermore, ARAN requires signatures on route requests and replies which represents a non-negligible added cost, and ARAN do not use hop-count and can therefore not be used for neighborhood discovery.

11.6 Conclusion

The analysis of the characteristics of opportunistic networks and content-based forwarding, lead us to the conclusion that key management in such networks should be self-organized and local. This locality also involves a correct view of the neighborhood topology. We therefore designed a complete solution that enables bootstrapping of security associations along with secure neighborhood discovery.

This solution based on pseudonym certificates and encapsulated signatures enables key agreement between a node (the initiator) and all its neighbors which are at distance less than lr-hops without pre-established trust relationship or infrastructure. The solution also enables the discovery of the neighborhood's topology and withstands tampering by malicious nodes. We also proposed the use of an Identity Manager which provides each node with a unique certified pseudonym during a setup phase. This lightweight IM therefore effectively prevents Sybil attacks. Furthermore the IM is offline and is not required during networking operations; therefore the key management scheme is self-organized.

The proposed scheme can therefore be used as an anchor to content based forwarding in opportunistic networks based on multiple layer commutative encryption, which results in end-to-end confidentiality and privacy-preserving content-based forwarding solely based on a local and self-organized key management.

Conclusion

In this thesis we focused on security issues in opportunistic networks.

We first presented an overview of existing forwarding protocols dedicated to opportunistic networks. All these protocols follow the store, carry and forward principle which offers the dynamicity and flexibility required to enable opportunistic communications. The strategies differ by the number of replicas of a message that are forwarded in the network (the cost) and the type of information used to take forwarding decisions. We thus classified the protocols and identified three main categories:

- oblivious forwarding protocols require the definition by the source of a precise destination, and which take forwarding decisions based only on the destination.
- context-based forwarding protocols require an implicit definition of the destination through its context. Forwarding decisions are taken based on a comparison of the context of a message and the context of nodes.
- content-based forwarding protocols, where messages are forwarded from a sender to all interested nodes: the message does not specify a destination and the whole content of the message can be used to take forwarding decisions based on the interests of potential recipients.

We then analyzed the following security issues with the constraints of opportunistic networks:

- Cooperation enforcement is essential to opportunistic communication, as forwarding of messages is performed by all nodes and not by a dedicated infrastructure of routers. We proposed an original solution based on the hot-potato principle [ÖSM07b]. It is worth noting that cooperation enforcement is required for other operations besides forwarding.
- Trust establishment is a difficult task in opportunistic networks because of the lack of infrastructure. We proposed an interesting approach to establish trust based on trusted communities.
- Integrity and authentication are classical security requirements that can still be met by classical security mechanisms. Nonetheless classical security solutions have

shortcomings when messages need to be fragmented or modified during the forwarding process like in network coding, but this issue is not specific to opportunistic communication. We addressed the problem of integrity and authentication in network coding by designing a signature scheme for linear combinations of packets in [ÖSM07a, ÖSM07c].

- Confidentiality and privacy, which are the core subjects of this thesis: we proposed four privacy models depending on the trust assumptions and defined their impact on networking operations.

As a practical example, we designed a security framework for the Haggle node architecture and implemented some security primitives. In particular we implemented attribute certificates and showed their use in a scenario where privacy of users is enforced in the second model. As a future work, it would also be interesting to implement more complex solutions in more demanding privacy models, like the solutions proposed in the two other parts of this thesis.

Concerning context-based forwarding mechanisms, we investigated security issues and their relation to trust assumptions thoroughly. We then proposed innovative solutions for confidentiality and privacy based on the trusted communities assumption:

- Based on an extended version of identity-based cryptography where identities are replaced by a set of attributes, we proposed a solution for payload confidentiality which allows any node to encrypt a message to a destination with a given set of attributes without knowing which nodes satisfy these conditions. Only the destination can decrypt the message and access the payload. The security of this scheme was proved in the `IND-MID-CPA` model that we introduced in the same chapter: only nodes which own all the private keys corresponding to the encrypting attributes can decrypt the message. The trusted communities assumption in turn ensures that only the destination has those keys, and therefore the solution provides end-to-end confidentiality with no end-to-end key agreement.

- The privacy-preserving scheme enables a node to encrypt the header of a message with a public function, such that intermediate nodes can discover only the matching attributes between the message and their own context. The intermediate nodes do not learn information about non-matching attributes, because the underlying cryptographic scheme, called PEKS, is semantically secure. Our specific instantiation of PEKS thus allows intermediate nodes to securely discover partial matches between their profile and the message context while preserving user privacy in the trusted communities assumption.

Both schemes (payload confidentiality and user privacy) rely on the existence of an offline Trusted Third Party which distributes keying material during a setup phase. The TTP is however not required for the correct execution of forwarding primitives during opportunistic communication, and is therefore compatible with opportunistic network assumptions. The combination of these two solutions thus enables secure context-based forwarding in the third privacy model, under the assumption that nodes are honest-but-curious. However, if we consider malicious nodes as well, then a different security issue arises: nodes can lie about their matching ratio to subvert traffic.

To deal with this issue we proposed a computation assurance mechanism that prevents a node N_k from claiming an erroneous matching ratio. The idea of the scheme is to request that N_k proves that it shares an attribute by providing a preimage corresponding to the digest of a pseudo-random number under a cryptographic hash function. Proving matching attributes separately incurs a privacy threat on N_k though: we thus enhance the solution with the introduction of counting Bloom filters. The enhanced solution allows N_k to prove its matching ratio globally, by enabling the verifier node N_i to compare the counting Bloom filter of N_k with a matching reference. Besides solving the secure forwarding problem, this solution also is an original contribution as a basic security primitive for privately computing the cardinality of set intersection using counting Bloom filters. We analyze the security of this mechanism through a probabilistic approach, by modeling the characteristics of random counting Bloom filters and then by evaluating the probability of success of an attacker. We show in particular that the probability of success of the attacker is exponentially decreasing with the amount of error that the attacker introduces.

The combination of the three aforementioned solutions constitutes a complete security framework for context-based routing in opportunistic networks. We mentioned three extensions in the thesis. The first one deals with the case of malicious TTPs in a practical way by distributing the capabilities of the TTP over several trusted entities: only the collusion of all trusted entities can produce a malicious result. The second tackles the issue of revocation and we proposed a solution based on epochs and keying materials automatically expiring after a certain period of time. Finally we addressed the issue of weighting attributes which can be dealt with thanks to the properties of counting Bloom filters.

Concerning content-based routing, our goal was to provide a solution in the most demanding privacy model. In order to solve privacy issues in content-based communication, we analyzed the link between privacy and confidentiality and identified two confidentiality requirements, namely publisher and information confidentiality. This led us to the more general problem of routing encrypted content using encrypted subscription filters. Secure routing requires two main primitives, namely building routing tables with aggregation of encrypted filters and secure look-up of encrypted content with encrypted routing tables to disseminate published content efficiently. These two primitives have to be designed together with the other classical primitives in order to solve the privacy-preserving routing which had no existing solution. We presented a solution to this problem based on multiple layers of Pohlig-Hellman encryptions. Our MLCE scheme allows brokers to perform secure transformations without having access to the data that is being transferred. Intermediate nodes can indeed remove or add an encryption layer without destroying the others and hence perform aggregation, routing tables building or look-up on private data protected by the other layers. This is the first scheme which enables privacy-preserving routing with no shared secret between end-users, thanks to the commutativity of MLCE. Another key feature of this protocol (which is also a key difference with respect to publish/subscribe) is that it allows intermediate nodes to be subscribers at the same time while preserving privacy of all nodes which is appealing for opportunistic networks.

The security of the protocol and its resilience to collusion attacks depends on the number lr of layers, on security associations with the neighbors that are lr hops away, and on a correct view of the lr-hop neighborhood. The MLCE scheme thus requires a self-organized and local key management solution adapted to opportunistic networks. We therefore designed a complete solution that enables bootstrapping of security associations along with secure neighborhood discovery. This solution based on the use of pseudonym certificates and encapsulated signatures enables key agreement between a node (the initiator) and all its neighbors which are at distance less than lr-hops without pre-established trust relationship or infrastructure. The solution also enables the discovery of the neighborhood's topology and withstands tampering by malicious nodes. An Identity Manager (IM) provides each node with a unique certified pseudonym during a setup phase in order to prevent Sybil attacks. As for the TTP in context-based security mechanisms, the IM is offline and is not required during networking operations.

Combining MLCE and the key management scheme results in end-to-end confidentiality and privacy-preserving content-based forwarding solely based on a local and self-organized key management.

Future Work

We first present some interesting subjects of future work directly related to our proposed solutions and then turn to more general subjects.

Concerning our context-based solution, we proposed an extension to take into account weighting of attributes, and we showed that it might lead to an inference attack, whereby nodes would learn information on the profile of their neighbors simply from the value matching ratio. In fact, this issue is crawling from the design of the context-based forwarding scheme itself. Indeed, the source plays a role which is very different from other nodes, as it knows the attributes that are encrypted in the message even though it does not have the corresponding attributes in its profile. To solve this issue we propose two ideas:

- The first idea is to enable the source to send messages only to its own community. This would however considerably restrict the communication capabilities of nodes.

- The second proposal is to modify the forwarding process as proposed in [SOM10b]: nodes would broadcast messages to their neighbors, and neighbors would compute the matching ratio locally and not disclose to any other node. The matching ratio would then be used as metric to rank received messages and decide either to drop them or carry and forward them. This would result in a hybrid protocol, which would be epidemic in essence and use context as a replication metric.

These two approaches are first ideas to get rid of the specific role of the source in context-based forwarding, but future work might focus on finding other solutions.

Moreover, our context-based solution assumes that attributes are static or that at least they do not change frequently. However it is possible and more interesting to perform context-based forwarding with dynamic attributes. Proposing security solutions encompassing dynamic attributes necessarily implies the possibility of dynamic revocation. We proposed a solution to revocation based on epochs, which can be viewed as a pragmatic fix, but investigating a solution for dynamic revocation without relying on expiration after a certain time is a challenging task.

Concerning our content-based solution, the locality of the solution is appealing for dynamic environments such as opportunistic networks, however the proposed solution presents few limitations since the knowledge of several (at least two) next hops is constraining in really dynamic environments. Indeed, knowledge of the next few hops at each node implies that there is a sliding window of at least two hops on the path between subscribing nodes and publishing ones, which would be equivalent to requiring an end-to-end path. The proposed solution hence fits restricted MobiOpps scenarios where the topology is predictable.

As a result of this analysis we realize that any form of routing, in the sense of knowing more than the next hop, is unpractical in most opportunistic network because of dynamicity and disruptions: building routing table is inefficient if the routing tables need to be updated too frequently. Opportunistic communication implies frequent topology changes, hence routing should be discarded to the benefit of hop-by-hop forwarding. Doing solely content-based forwarding implies an epidemic spread of the content, each node proposing to the other its content. In this case, privacy requires techniques such as private matching or oblivious transfer.

Another alternative to achieve efficient content dissemination consists in using hybrid strategies: we already hinted at the benefits of context-based and epidemic forwarding strategies. Another hybrid approach is to combine content-based with context-based forwarding: the assumption behind such a forwarding strategy is that nodes which share context are likely to share interests as well. Proposals following this idea have been presented by Costa et al. in [CMMP08] and Baldoni et al. in [BBC$^+$05], and we believe that this idea presents an interesting potential for research, both from the point of view of networking and security.

Finally, most of our work in this thesis was related to computation on encrypted data in different scenarios: matching encrypted context, looking-up encrypted data in encrypted routing tables. In this area, Gentry announced very recently a breakthrough result: a fully homomorphic encryption [Gen09]. Such a scheme in fact enables the evaluation of any polynomial based on encrypted data, and could be used to solve all problems related to computation on encrypted data. However, research in this area is far from being over. First of all, the scheme proposed by Gentry is a nice theoretical result but is very costly, and, as acknowledged by Gentry, *"Making the full scheme practical remains an open problem"*. Depending on the exact scenario which is considered, it is hence interesting to design efficient solutions that exactly meet the requirements of the scenario, and this was our purpose in this thesis in the design of privacy-preserving schemes. This same quest for efficiency lead us to design a multiple identity-based encryption scheme: existing policy-based encryption [BMC06] or attribute-based encryption [BSW07] could solve the problem of payload confidentiality but at a higher cost.

Computation on encrypted data is a useful tool for electronic applications where privacy or confidentiality is critical such as health or biometry related applications. Another promising area of application is the exploding paradigm of cloud computing: generally, the trust level in the cloud is low, hence encrypting data and requesting the cloud to perform operations on encrypted data is an interesting approach to obtain a service while keeping control over the data.

Bibliography

[ABE+04] Bengt Ahlgren, Marcus Brunner, Lars Eggert, Robert Hancock, and Stefan Schmid. Invariants: a new design methodology for network architectures. In *FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 65–70. ACM, 2004.

[AES03] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant. Information sharing across private databases. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 86–97. ACM, 2003.

[AH00] Eytan Adar and Bernardo A. Huberman. Free Riding on Gnutella. *First Monday*, 5(10), 2000. http://www.firstmonday.org/.

[AKG+07] N. Asokan, Kari Kostiainen, Philip Ginzboorg, Jörg Ott, and Cheng Luo. Applicability of identity-based cryptography for disruption-tolerant networking. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 52–56. ACM, 2007.

[AKK+07] N. Asokan, Kari Kostiainen, Kari Kostiainen, Philip Ginzboorg, Philip Ginzboorg, Jörg Ott, Jörg Ott, Cheng Luo, and Cheng Luo. Towards Securing Disruption-Tolerant Networking. Technical Report NRC-TR-2007-007, Nokia Research Center, march 2007. http://research.nokia.com/files/NRC-TR-2007-007.pdf.

[ARH97] Alfarez Abdul-Rahman and Stephen Hailes. A Distributed Trust Model. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms*, pages 48–60. ACM, 1997.

[ASYP04] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black hole attack in mobile Ad Hoc networks. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 96–97. ACM, 2004.

[BBC+05] R. Baldoni, R. Beraldi, G. Cugola, M. Migliavacca, and L. Querzoni. Structure-less content-based routing in mobile ad hoc networks. In *ICPS '05: Proceedings of the IEEE International Conference on Pervasive Services*, pages 37–46. IEEE Computer Society, 2005.

[BBL05] Brendan Burns, Oliver Brock, and Brian Neil Levine. Mv routing and capacity building in disruption tolerant networks. In *INFOCOM 2005: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 398–408, March 2005.

[BBM00] M. Bellare, A. Boldyreva, and Silvio Micali. Public-key encryption in a multiuser setting: Security proofs and improvements. In *Advances in Cryptology - EUROCRYPT 2000*, pages 259–274. Springer Verlag, 2000.

[BBQ$^+$05] Roberto Baldoni, Roberto Beraldi, Leonardo Querzoni, Gianpaolo Cugola, and Matteo Migliavacca. Content-Based Routing in Highly Dynamic Mobile Ad Hoc Networks. *International Journal of Pervasive Computing and Communications*, 1(4):277–288, 2005.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BCC$^+$05] Laurent Bussard, Joris Claessens, Stefano Crosta, Yves Roudier, and Alf Zugenmaier. Can we take this off-line? Credentials for Web services supported nomadic applications. In *SAR'05, 4th Conference on Security and Network Architectures, June 6-10, 2005, Batz-sur-Mer, France*, 06 2005.

[BCJP07] Chiara Boldrini, Marco Conti, Jacopo Jacopini, and Andrea Passarella. HiBOp: a History Based Routing Protocol for Opportunistic Networks. In *WOWMOM '07: Proceedings of the Eighth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*. IEEE Computer Society, June 2007.

[BCM$^+$99] Guruduth Banavar, Tushar Chandra, Bodhi Mukherjee, Jay Nagarajarao, Robert E. Strom, and Daniel C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *PICDCS '99: Proceedings of 19th IEEE International Conference on Distributed Computing Systems*, pages 262–272. IEEE Computer Society, 1999.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public Key Encryption with keyword Search. In *Advances in Cryptology - EUROCRYPT 2004*, pages 506–522. Springer Berlin/Heidelberg, 2004.

[BCP08] Chiara Boldrini, Marco Conti, and Andrea Passarella. Exploiting users' social relations to forward data in opportunistic networks: The hibop solution. *Pervasive and Mobile Computing*, 4(5):633–657, 2008.

[BD94] Mike Burmester and Yvo Desmedt. A secure and efficient conference key distribution system. In *Advances in Cryptology - EUROCRYPT 1994*, pages 275–286. Springer Berlin/Heidelberg, 1994.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 26–45. Springer-Verlag, 1998.

[BEGA03] R.B. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh. Bootstrapping security associations for routing in mobile ad-hoc networks. In *GLOBECOM '03: Proceedings of the 2003 IEEE Global Telecommunications Conference*, December 2003.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer-Verlag, 2001.

[BFKW09] Dan Boneh, David Freeman, Jonathan Katz, and Brent Waters. Signing a Linear Subspace: Signature Schemes for Network Coding. In *PKC '09: Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography*, pages 68–87. Springer-Verlag, 2009.

[BGJL06] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006: Proceedings of 25th IEEE International Conference on Computer Communications*, pages 1–11, 2006.

[BGM$^+$08] Chiara Boldrini, Silvia Giordano, Refik Molva, Erik Nordström, Melek Önen, Andrea Passarella, Christian Rohner, Abdullatif Shikfa, and Salvatore Vanini. Haggle deliverable 1.3: Specification of the YOUNG-Haggle, August 2008. http://www.haggleproject.org/deliverables/D1.3_final.pdf.

[BH03] Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8(5):579–592, 2003.

[Bir93] Kenneth P. Birman. The process group approach to reliable distributed computing. *Communications of the ACM*, 36(12):37–53, 1993.

[BKB08] Prithwish Basu, Rajesh Krishnan, and Daniel Brown. Persistent Delivery with Deferred Binding to Descriptively Named Destinations. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.* IEEE, November 2008.

[BKM$^+$09] Erik-Oliver Blass, Anil Kurmus, Refik Molva, Guevara Noubir, and Abdullatif Shikfa. The Ff-family of protocols for RFID-privacy and authentication. In *RFIDSec'09, 5th Workshop on RFID Security*, june 2009.

[BKM$^+$10] Erik-Oliver Blass, Anil Kurmus, Refik Molva, Guevara Noubir, and Abdullatif Shikfa. The Ff-family of protocols for RFID-privacy and authentication. *IEEE Transactions on Dependable And Secure Computing*, 2010. To be published.

[BLB02a] Sonja Buchegger and Jean-Yves Le Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In *Proceedings of the 10th Euromicro Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403 – 410. IEEE Computer Society, 2002.

[BLB02b] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236. ACM, 2002.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer-Verlag, 2001.

[BLV07] Aruna Balasubramanian, Brian Neil Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. *SIGCOMM Computing and Communications Review*, 37(4):373–384, 2007.

[BM02] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[BMC06] Walid Bagga, Refik Molva, and Stefano Crosta. Policy-based encryption schemes from bilinear pairings. In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 368–368. ACM, 2006.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[BSS99] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Elliptic Curves in Cryptography.* Cambridge University Press, 1999.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *SP '07: Proceedings of the 2007 IEEE Sym-*

posium on Security and Privacy, pages 321–334. IEEE Computer Society, 2007.

[CBH03] Srdjan Capkun, Levente Buttyán, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.

[CC03] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. In *PKC '03: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*, pages 18–30. Springer-Verlag, 2003.

[CCC+05] Manuel Costa, Jon Crowcroft, Miguel Castro, Antony Rowstron, Lidong Zhou, Lintao Zhang, and Paul Barham. Vigilante: end-to-end containment of internet worms. In *SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 133–147. ACM, 2005.

[CCH06] Mario Cagalj, Srdjan Capkun, and Jean-Pierre Hubaux. Key Agreement in Peer-to-Peer Wireless Networks. *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, 94(2), 2006.

[CCR03] Xiaoyan Chen, Ying Chen, and Fangyan Rao. An efficient spatial publish/subscribe system for intelligent location-based services. In *DEBS '03: Proceedings of the 2nd international workshop on Distributed event-based systems*. ACM, 2003.

[CH06] S. Capkun and J.-P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24, February 2006.

[CH08] Lukasz Chmielewski and Jaap-Henk Hoepman. Fuzzy private matching (extended abstract). In *ARES 2008: Proceedings of the International Conference on Availability, Reliability and Security*, pages 327–334. IEEE Computer Society, 2008.

[CHB06] Srdjan Capkun, Jean-Pierre Hubaux, and Levente Buttyan. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, 2006.

[CHC+06] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *INFOCOM 2006: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2006.

[CHL04] Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. Efficient ID-based group key agreement with bilinear maps. In *PKC 2004: Proccedings of the*

2004 international workshop on theory and practice in public key cryptography, pages 130–144. Springer Berlin/Heidelberg, 2004.

[CJ03] Thomas Heide Clausen and Philippe Jacquet. Optimized Link State Routing Protocol (OLSR), October 2003. http://www.ietf.org/rfc/rfc3626.txt.

[CMMP06] Paolo Costa, Mirco Musolesi, Cecilia Mascolo, and Gian Pietro Picco. Adaptive Content-based Routing for Delay-tolerant Mobile Ad Hoc Networks. Technical Report RN_06_08, UCL Department of Computer Science, UK, August 2006. http://www.cs.ucl.ac.uk/research/researchnotes/documents/RN_06_08.pdf.

[CMMP08] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 26(5):748 –760, june 2008.

[CP05] Paolo Costa and Gian Pietro Picco. Semi-probabilistic content-based publish-subscribe. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 575–585. IEEE Computer Society, 2005.

[CRS10] Mooi Choo Chuah, Sankardas Roy, and Ivan Stoev. Secure Descriptive Message Dissemination in Disruption Tolerant Networks. In *MobiOpp 2010: Proceedings of the Second ACM/SIGMOBILE International Workshop on Mobile Opportunistic Networking*. Elsevier, February 2010.

[CRW01] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems (TOCS)*, 19(3):332–383, 2001.

[CRW04] Antonio Carzaniga, Matthew J. Rutherford, and Alexander L. Wolf. A Routing Scheme for Content-Based Networking. In *INFOCOM 2004: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004.

[CS05] Yuan Chen and Karsten Schwan. Opportunistic overlays: efficient content delivery in mobile ad hoc networks. In *Middleware '05: Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, pages 354–374. Springer-Verlag New York, Inc., 2005.

[CW03] Antonio Carzaniga and Alexander L. Wolf. Forwarding in a content-based network. In *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication,*, pages 163–174, August 2003.

[DDS] OMG Data Distribution Service. http://www.omgwiki.org/dds/.

[DFGV03] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 257–266. ACM, 2003.

[DFL01] James A. Davis, Andrew H. Fagg, and Brian N. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *ISWC '01: Proceedings of the 5th IEEE International Symposium on Wearable Computers*, page 141. IEEE Computer Society, 2001.

[DGN$^+$07] Franca Delmastro, Silvia Giordano, Hoang Anh Nguyen, Erik Nordström, Melek Önen, Andrea Passarella, Alessandro Puiatti, Christian Rohner, George Theodorakopoulos, and Salvatore Vanini. Haggle deliverable 1.2: Specification of the CHILD-Haggle, August 2007. http://www.haggleproject.org/deliverables/D1.2_final.pdf.

[DGRS03] A. K. Datta, M. Gradinariu, M. Raynal, and G. Simon. Anonymous publish/subscribe in p2p networks. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*. IEEE Computer Society, 2003.

[Dir95] Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, October 1995. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML.

[DLA02] Hongmei Deng, Wei Li, and Dharma P. Agrawal. Routing Security in Wireless Ad Hoc Networks. *IEEE Communications Magazine, Special Topics on Security in Telecommunication Networks*, 40(10):70–75, 2002.

[Dou02] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. Springer-Verlag, 2002.

[DTN] Irtf delay tolerant networking research group (dtnrg). http://www.dtnrg.org/.

[DUP02] Avri Doria, Maria Uden, and Durga Prasad Pandey. Providing connectivity to the saami nomadic community. In *DYD '02: Proceedings of the 2nd International Conference on Open Collaborative Design for Sustainable Development*, December 2002.

[DVOW92] Whitfield Diffie, Paul C. Van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

[ECCD08] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. Delegation Forwarding. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260. ACM, 2008.

[ECD05] ANSI X9.62:2005 - Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute, November 2005. http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.62%3A2005.

[EFLBS06a] Alaeddine El Fawal, Jean-Yves Le Boudec, and Kave Salamatian. Performance Analysis of Self Limiting Epidemic Forwarding. Technical Report LCA-REPORT-2006-127, EPFL, Lausanne, 2006. http://infoscience.epfl.ch/record/94381.

[EFLBS06b] Alaeddine El Fawal, Jean-Yves Le Boudec, and Kave Salamatian. Self-Limiting Epidemic Forwarding. Technical Report LCA-REPORT-2006-126, EPFL, Lausanne, 2006. http://infoscience.epfl.ch/record/89512.

[ESP06] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[Fal03] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[Far07] Stephen Farrell. Dtn key management requirements, June 2007.

[FCAB00] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000.

[FNP04] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology - EUROCRYPT 2004*. Springer Verlag, 2004.

[FR94] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, 1994.

[GA04] Anargyros Garyfalos and Kevin C. Almeroth. Coupon Based Incentive Systems and the Implications of Equilibrium Theory. In *CEC '04: Proceedings of the IEEE International Conference on E-Commerce Technology*, pages 213–220. IEEE Computer Society, 2004.

[Gam88] Diego Gambetta. *Can We Trust Trust?*, pages 213–237. Basil Blackwell, 1988.

[Ge06] Silvia Giordano and Alessandro Puiatti editors. Haggle deliverable 1.1: Specification of first Haggle application and INFANT-Haggle, September 2006. http://www.haggleproject.org/images/b/b0/D11_final.pdf.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178. ACM, 2009.

[GK00] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Mar 2000.

[GLBML01] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for sharing in peer-to-peer networks. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 75–87. Springer-Verlag, 2001.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Special issue of Journal of Computer and Systems Sciences*, 28(2):270–299, 1984.

[Gou08] Xavier Gourdon. *Analyse, Les maths en tête : Mathématiques pour MP* (2^{e} édition)*. Ellipses Marketing, February 2008.

[GR06] Christos Gkantsidis and Pablo Rodriguez Rodriguez. Cooperative security for network coding file distribution. In *INFOCOM 2006: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2006.

[GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In *Information Hiding*, pages 137–150. Springer-Verlag, 1996.

[GRS99] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing. *Commununications of the ACM*, 42(2):39–41, 1999.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT '02: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566. Springer-Verlag, 2002.

[GT02] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[Hag06] The Haggle Project, 2006. http://www.haggleproject.org/index.php.

[HCS$^+$05] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251. ACM, 2005.

[HCY08] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 241–250. ACM, 2008.

[HG08] Julien Haillot and Frédéric Guidec. Content-based communication in disconnected mobile ad hoc networks. In *NOTERE '08: Proceedings of the 8th international conference on New technologies in distributed systems*. ACM, 2008.

[HGM01] Yongqiang Huang and Hector Garcia-Molina. Publish/subscribe in a mobile environment. In *MOBIDE '01: ACM international Workshop on Data Engineering for wireless and mobile access*, pages 27–34. ACM, 2001.

[HGM03] Yongqiang Huang and Hector Garcia-Molina. Publish/subscribe tree construction in wireless ad-hoc networks. In *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management*, pages 122–140. Springer-Verlag, 2003.

[HGM04] Yongqiang Huang and Hector Garcia-Molina. Publish/subscribe in a mobile environment. *Wireless Networks*, 10(6):643–652, 2004.

[HJP02a] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, 2002.

[HJP02b] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*. IEEE Computer Society, 2002.

[HKLM03] Andreas Heinemann, Jussi Kangasharju, O Lyardet, and Max Mühlhäuser. iClouds Ũ Peer-to-Peer Information Sharing in Mobile Environments. In *Proceedings of the 9th International Euro-Par Conference, (Euro-Par 2003), volume 2790 of Lecture Notes in Computer Science*, pages 1038–1045. Springer, 2003.

[HPJ03] Y. C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *INFOCOM 2003: Proccedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1976–1986. IEEE Societies, April 2003.

[HPJ05a] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11(1-2):21–38, 2005.

[HPJ05b] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, January 2005.

[HYCC07] Pan Hui, Eiko Yoneki, Shu Yan Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *MobiArch '07: Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*. ACM, 2007.

[IR02] Ana Lúcia Iacono and Christopher Rose. *Infostations: A new perspective on wireless data networks*, volume 598 of *The Springer International Series in Engineering and Computer Science: Next Generation Wireless Networks*, pages 3–63. Springer Netherlands, 2002.

[JNP05] Deepti Joshi, Kamesh Namuduri, and Ravi Pendse. Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: an analysis. *EURASIP Journal on Wireless Communications and Networking*, September 2005.

[JOW$^+$02] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *ACM SIGOPS Operating Systems Review*, 36(5):96–107, 2002.

[JSB$^+$06] Sushant Jain, Rahul C. Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3):327–339, 2006.

[KBM$^+$07] Rajesh Krishnan, Prithwish Basu, Joanne M. Mikkelson, Christopher Small, Ram Ramanathan, Daniel W. Brown, John R. Burgess, Armando L. Caro, Matthew Condell, Nicholas C. Goffee, Regina Rosales Hain, Richard E. Hansen, Christine E. Jones, Vikas Kawadia, David P. Mankins, Beverly I. Schwartz, William T. Strayer, Jeffrey W. Ward, David P. Wiggins, and Stephen H. Polit. The SPINDLE Disruption-Tolerant Networking System. In *IEEE MILCOM 2007: Proceedings of the 2007 Military Communications Conference*. IEEE, October 2007.

[Kio06] The KioskNet Project, 2006. http://blizzard.cs.uwaterloo.ca/tetherless/index.php/KioskNet.

[KKA03] Aram Khalili, Jonathan Katz, and William A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *SAINT-W '03: Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops).* IEEE Computer Society, 2003.

[KLL04] Hyun-Jeong Kim, Su-Mi Lee, and Dong Hoon Lee. Constant-Round Authenticated Group Key Exchange for Dynamic Groups. In *Advances in Cryptology - ASIACRYPT 2004*, pages 127–140. Springer Berlin/Heidelberg, 2004.

[KP05] Caroline Kudla and Kenneth G. Paterson. Modular Security Proofs for Key Agreement Protocols. In *Advances in Cryptology - ASIACRYPT 2005*, pages 549–565. Springer Berlin/Heidelberg, 2005.

[Kra03] Hugo Krawczyk. Sigma: The 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols. In *CRYPTO '03: Proceedings of the 23rd Annual International Cryptology Conference on Advances in Cryptology*, pages 400–425. Springer, 2003.

[KS05] Stephen Kent and Karen Seo. Security Architecture for the Internet Protocol, December 2005. http://tools.ietf.org/html/rfc4301.

[KZH07] Aniket Kate, Gregory M. Zaverucha, and Urs Hengartner. Anonymity and Security in Delay Tolerant Networks. *SecureComm 2007: Proceedings of the third International Conference on Security and Privacy in Communications Networks and the Workshops*, pages 504–513, September 2007.

[KZL$^+$01] Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *ICNP '01: Proceedings of the IEEE 9th International Conference on Network Protocols*. IEEE Computer Society, 2001.

[LDS03] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[LKBG06] L. Lilien, Z.H. Kamal, V. Bhuse, and A. Gupta. Opportunistic Networks: The Concept and Research Challenges in Privacy and Security. In *Challenges in Privacy and Security, Proceedings of the NSF International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN 2006)*, March 2006.

[LRS05] Hongzhou Liu, Venugopalan Ramasubramanian, and Emin Gün Sirer. Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association, 2005.

[LTH04] Yaping Li, Doug Tygar, and Joseph M. Hellerstein. Private matching. Technical Report IRB-TR-04-005, Intel Research Laboratory Berkeley, 2004. http://www.eecs.berkeley.edu/~tygar/papers/Private_matching.pdf.

[Lun00] Janne Lundberg. Routing security in ad hoc networks. Technical Report Tik-110.501, Helsinki University of Technology, 2000. http://web.informatik.uni-bonn.de/IV/Mitarbeiter/mp/paper/secure_routing/routing security in ad hoc networks - lundberg.pdf.

[Lyn06] Ben Lynn. The pairing-based cryptography library, 2006. http://crypto.stanford.edu/pbc/.

[LZK+02] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, and Lixia Zhang. Self-securing ad hoc wireless networks. In *ISCC'02: Proceedings of the IEEE Seventh International Symposium on Computers and Communications*. IEEE Computer Society, 2002.

[LZM+09] Haitao Liu, Baoxian Zhang, Hussein T. Mouftah, Xiaojun Shen, and Jian Ma. Opportunistic Routing for Wireless Ad Hoc and Sensor Networks: Present and Future Directions. *IEEE Communications Magazine*, 12:103–109, December 2009.

[MC02] René Meier and Vinny Cahill. Steam: Event-based middleware for wireless ad hoc network. In *ICDCSW '02: Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 639–644. IEEE Computer Society, 2002.

[MDM07] Johann Van Der Merwe, Dawoud Dawoud, and Stephen McDonald. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computer Surveys (CSUR)*, 39(1), 2007.

[MdRK04] Ronald Mannak, Huib de Ridder, and David V. Keyson. The human side of sharing in peer-to-peer networks. In *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence*, pages 59–64. ACM, 2004.

[MHM05] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In *WOWMOM '05: Proceedings of the Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks*, pages 183–189, 2005.

[MM02] Pietro Michiardi and Refik Molva. CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121. Kluwer, B.V., 2002.

[MM03] Pietro Michiardi and Refik Molva. A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad hoc networks. In *WiOpt '03: Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 3–5, 2003.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE transactions on information theory*, 39(5):1639–1646, 1993.

[Mpa] The mpala research centre. http://www.nasm.edu/ceps/mpala/main.html.

[MPR09] Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing-Is-Believing: using camera phones for human-verifiable authentication. *International Journal of Security and Networks*, 4(1/2):43–56, 2009.

[MVO91] Alfred Menezes, Scott Vanstone, and Tatsuaki Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 80–89, 1991.

[NGP07] Hoang Anh Nguyen, Silvia Giordano, and Alessandro Puiatti. Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (propicman). In *WOWMOM '07: Proceedings of the Eighth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks.* IEEE Computer Society, June 2007.

[ÖM07] Melek Önen and Refik Molva. Secure data aggregation with multiple encryption. In *EWSN 2007: 4th European Conference on Wireless Sensor Networks*, Lecture Notes in Computer Science, pages 117–132. Springer, January 2007.

[OP01] Lukasz Opyrchal and Atul Prakash. Secure distribution of events in content-based publish subscribe systems. In *SSYM'01: Proceedings of the 10th conference on USENIX Security Symposium.* USENIX Association, 2001.

[OPA07] Lukasz Opyrchal, Atul Prakash, and Amit Agrawal. Supporting Privacy Policies in a Publish-Subscribe Substrate for Pervasive Environments. *JOURNAL OF NETWORKS*, 2(1):17–26, February 2007.

[Ope99] Openssl, 1999. http://www.openssl.org/.

[OPS10] OPS - Open Publish-Subscribe, 2010. http://code.google.com/p/ops/.

[ÖS09] Melek Önen and Abdullatif Shikfa. Haggle deliverable 4.3: Prototype of trust and security mechanisms, December 2009. http://www.haggleproject.org/images/c/c3/D43_final.pdf.

[ÖSM07a] Melek Önen, Abdullatif Shikfa, and Refik Molva. Haggle deliverable 4.1: Preliminary design of trust and security mechanisms, June 2007. http://www.haggleproject.org/deliverables/D4.1_final.pdf.

[ÖSM07b] Melek Önen, Abdullatif Shikfa, and Refik Molva. Optimistic fair exchange for secure forwarding. In *MOBIQUITOUS '07: Proceedings of the 2007 Fourth*

Annual International Conference on Mobile and Ubiquitous Systems: Networking&Services (MobiQuitous). IEEE Computer Society, 2007.

[ÖSM07c] Melek Önen, Abdullatif Shikfa, and Refik Molva. SigNCode: A provably secure homomorphic signature scheme for network coding. Technical Report RR-07-202, Department Network and Security, EURECOM, September 2007. http://www.eurecom.fr/util/publidownload.fr.htm?id=2337.

[ÖSTB08] Melek Önen, Abdullatif Shikfa, George Theodorakopoulos, and Jean-Yves Le Boudec. Haggle deliverable 4.2: Complete design of trust and security mechanisms, December 2008. http://www.haggleproject.org/deliverables/D4.2_final.pdf.

[PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM Computing and Communications Review*, 24(4), 1994.

[PBRD03] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing, July 2003. http://www.ietf.org/rfc/rfc3561.txt.

[PCTS02] Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Song. Efficient and secure source authentication for multicast. *ICNP'02: Proceedings of the 2002 IEEE International Conference on Network Protocols*, November 2002.

[PFH04] Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. Daknet: Rethinking Connectivity in Developing Nations. *Computer*, 37:78–83, 2004.

[PH78] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24(1):106–110, Jan 1978.

[PH03] Panagiotis Papadimitratos and Zygmunt J. Haas. Securing mobile ad hoc networks. *The handbook of ad hoc wireless networks*, pages 551–567, 2003.

[PH09] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. Technical report, TU DRESDEN, December 2009. http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

[PM02] Alain Pannetrat and Refik Molva. Multiple layer encryption for multicast groups. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 137–153. Kluwer, B.V., 2002.

[PPC06a] Luciana Pelusi, Andrea Passarella, and Marco Conti. Beyond MANETs: dissertation on Opportunistic Networking. Technical report, IIT-CNR, May 2006. http://bruno1.iit.cnr.it/˜ andrea/tr/pelusi06_tr.pdf.

[PPC06b] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, November 2006.

[PSH10] PubSubHubbub, 2010. http://code.google.com/p/pubsubhubbub/.

[Res00] Eric Rescorla. *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley Professional, 2000.

[RFJY03] Olga Ratsimor, Tim Finin, Anupam Joshi, and Yelena Yesha. eNcentive: a framework for intelligent marketing in mobile peer-to-peer environments. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pages 87–94. ACM, 2003.

[Riv92] Ronald L. Rivest. The MD5 Message-Digest Algorithm, April 1992. http://www.ietf.org/rfc/rfc1321.txt.

[RR06] Costin Raiciu and David S. Rosenblum. Enabling confidentiality in content-based publish/subscribe infrastructures. In *Securecomm and Workshops, 2006*, August 2006.

[RRH06] Costin Raiciu, David S. Rosenblum, and Mark Handley. Revisiting content-based publish/subscribe. In *ICDCSW '06: Proceedings of the 26th IEEE International ConferenceWorkshops on Distributed Computing Systems*. IEEE Computer Society, 2006.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SDL$^+$02] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*. IEEE Computer Society, November 2002.

[SFWL06] Susan Flynn Symington, Stephen Farrell, Howard Weiss, and Peter Lovell. Delay-Tolerant Networking Security Overview, draft-irtf-dtnrg-sec-overview-02, October 2006. http://tools.ietf.org/html/draft-irtf-dtnrg-sec-overview-02.

[SFWL10] Susan Flynn Symington, Stephen Farrell, Howard Weiss, and Peter Lovell. Bundle Security Protocol Specification, draft-irtf-dtnrg-bundle-security-14, January 2010. http://tools.ietf.org/html/draft-irtf-dtnrg-bundle-security-14.

[SGG03] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia Systems*, 9(2):170–184, 2003.

[SGP04] Katrine Stemland Skjelsvik, Vera Goebel, and Thomas Plagemann. Distributed event notification for mobile ad hoc networks. *IEEE Distributed Systems Online*, 5(8), 2004.

[SH03] Tara Small and Zygmunt J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 233–244. ACM, 2003.

[SHA08] FIPS 180-3: Secure Hash Standard (SHS). National Institute of Standards and Technology, October 2008. http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf.

[SHA12] Cryptographic hash algorithm competition. National Institute of Standards and Technology, 2007-2012. http://csrc.nist.gov/groups/ST/hash/sha-3/index.html.

[Shi07] Robert W. Shirey. RFC 4949: Internet Security Glossary, Version 2, August 2007. http://tools.ietf.org/html/rfc4949.

[Shi10] Abdullatif Shikfa. Security issues in opportunistic networks. In *MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pages 215–216. ACM, 2010.

[Sil86] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Springer, 1986.

[SK05] Aaditeshwar Seth and Srinivasan Keshav. Practical Security for Disconnected Nodes. In *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols (NPSec)*, November 2005.

[SL07] Mudhakar Srivatsa and Ling Liu. Secure event dissemination in publish-subscribe networks. In *ICDCS '07: Proceedings of the 27th International Conference on Distributed Computing Systems*. IEEE Computer Society, 2007.

[SÖM09a] Abdullatif Shikfa, Melek Önen, and Refik Molva. Privacy in Content-Based Opportunistic Networks. In *WAINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, pages 832–837. IEEE Computer Society, 2009.

[SÖM09b] Abdullatif Shikfa, Melek Önen, and Refik Molva. Privacy in context-based and epidemic forwarding. In *WOWMOM '09: Proceedings of the 2009 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks"*, June 2009.

[SÖM09c] Abdullatif Shikfa, Melek Önen, and Refik Molva. Privacy-preserving content-based publish/subscribe networks. In *Emerging Challenges for Security, Privacy and Trust: Proceedings of the 24th IFIP TC 11 International Informa-*

tion Security Conference, SEC 2009, pages 270–282. Springer Boston, May 2009.

[SÖM10a] Abdullatif Shikfa, Melek Önen, and Refik Molva. Bootstrapping security associations in opportunistic networks. In *PERCOM '10: Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications*. IEEE Computer Society, 2010.

[SOM10b] Abdullatif Shikfa, Melek Önen, and Refik Molva. Privacy and confidentiality in context-based and epidemic forwarding. *Computer Communications*, 33(13):1493–1504, 2010.

[SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[SPR$^+$09] Reza Shokri, Marcin Poturalski, Gael Ravot, Panos Papadimitratos, and Jean-Pierre Hubaux. A practical secure neighbor verification protocol for wireless sensor networks. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security*, pages 193–200. ACM, 2009.

[Sti95] Douglas R. Stinson. *Cryptography: theory and practice.* CRC Press, 1995.

[SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 44–55. IEEE Computer Society, 2000.

[TDG$^+$09] Luca Della Toffola, Franca Delmastro, Silvia Giordano, Melek Önen, Andrea Passarella, Daniele Puccinelli, Christian Rohner, Abdullatif Shikfa, and Salvatore Vanini. Haggle deliverable 1.4: Specification of the ADULT-Haggle, July 2009. http://www.haggleproject.org/images/9/94/Deliverable1.4.pdf.

[TG04] Stefan Tillich and Johann Großschädl. A survey of public-key cryptography on j2me-enabled mobile devices. In *Computer and Information Sciences - ISCIS 2004*, pages 935–944. Springer-Verlag, 2004.

[Thi02] Patrick Thibodeau. Pervasive computing has pervasive problems. *ComputerWorld*, 36(41), october 2002.

[URHIK08] Sumair Ur Rahman, Urs Hengartner, Usman Ismail, and S. Keshav. Practical security for rural internet kiosks. In *NSDR '08: Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*, pages 13–18. ACM, 2008.

[VB00] Amin Vahdat and David Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000. http://cseweb.ucsd.edu/~vahdat/papers/epidemic.pdf.

[WCEW02] C. Wang, A. Carzaniga, D. Evans, and A. Wolf. Security issues and requirements for internet-scale publish-subscribe systems. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9*. IEEE Computer Society, 2002.

[Web] IBM Websphere MQ. http://www-01.ibm.com/software/integration/wmq/.

[WLB05] Jörg Widmer and Jean-Yves Le Boudec. Network coding for efficient communication in extreme networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 284–291. ACM, 2005.

[WQA$^+$02] Yi-Min Wang, Lili Qiu, Dimitris Achlioptas, Gautam Das, Paul Larson, and Helen J. Wang. Subscription Partitioning and Routing in Content-based Publish/Subscribe Systems. In *16th International Symposium on DIStributed Computing*, 2002.

[WWF$^+$07] Bing Wu, Jie Wu, Eduardo B. Fernandez, Mohammad Ilyas, and Spyros Magliveras. Secure and efficient key management in mobile ad hoc networks. *Journal of Network and Computer Applications*, 30(3), 2007.

[XI04] Gang Xu and L. Iftode. Locality driven key management architecture for mobile ad-hoc networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. IEEE, October 2004.

[YK02] Seung Yi and Robin Kravets. Key management for heterogeneous ad hoc wireless networks. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 202–205. IEEE Computer Society, 2002.

[YLY$^+$04] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in Mobile Ad Hoc Networks: Challenges and Solutions. *IEEE Wireless Communications*, 11(1):38–47, 2004.

[ZAZ04] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198. ACM, 2004.

[ZCY03] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *INFOCOM 2003:*

Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, volume 3, pages 1987–1997, March-April 2003.

[Zeb02] The zebranet wildlife tracker, 2002. http://www.princeton.edu/ mrm/zebranet.html.

[ZH99] Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 1999.

[Zha06] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37, 1st Quarter 2006.

[Zim95] Philip R. Zimmermann. *The official PGP user's guide.* MIT Press, 1995.

[ZS00] Hu Zhou and Suresh Singh. Content based multicast (cbm) in ad hoc networks. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 51–60. IEEE Press, 2000.