Local key management in opportunistic networks

Abdullatif Shikfa*

Alcatel-Lucent Bell Labs France,
Security Research Group,
Route de Villejust, 91620 Nozay, France
E-mail: abdullatif.shikfa@alcatel-lucent.com
*Corresponding author

Melek Önen and Refik Molva

EURECOM,
2229 Route des Crêtes, BP 193,
06560 Sophia-Antipolis Cedex, France
E-mail: onen@eurecom.fr
E-mail: molva@eurecom.fr

Abstract: Opportunistic networks are a new and specific type of mobile peer-to-peer networks where end-to-end connectivity cannot be assumed. These networks present compelling challenges, especially from a security perspective, as interactive protocols are infeasible in such environments. In this article, we focus on the problem of key management in the framework of content-based forwarding and opportunistic networks. After analysing this issue and identifying specific security threats such as Sybil attacks, we propose a specific key management scheme that enables the bootstrapping of local, topology-dependent security associations between a node and its neighbours along with the discovery of the neighbourhood topology, thanks to the use of pseudonym certificates and encapsulated signatures.

Keywords: key management; opportunistic communication; secure neighbourhood discovery; local security associations; communication networks.

Reference to this paper should be made as follows: Shikfa, A., Önen, M. and Molva, R. (2012) 'Local key management in opportunistic networks', *Int. J. Communication Networks and Distributed Systems*, Vol. 9, Nos. 1/2, pp.97–116.

Biographical notes: Abdullatif Shikfa is a Researcher at the Alcatel-Lucent Bell Labs France in the Security Research Group. Previously, he was a Research Engineer working at EURECOM in the network and security team on the EU project Haggle. His research focused on applied cryptography and security protocols in challenging environments, such as mobile opportunistic networks or RFIDs. He obtained his PhD from the ENST in 2010 and he is an Alumni of Ecole Polytechnique (MSc in 2004), of University of Nice Sophia Antipolis (MSc in 2005) and of ENST-EURECOM (MSc in 2006).

Melek Önen is a Senior Researcher at EURECOM. Her current research interests are the design of security and privacy protocols for ad hoc networks, sensor networks, opportunistic networks and social networks. She obtained her PhD in Computer Science from the ENST in 2005. Her thesis was focusing on securing multicast communications in satellite networks.

Refik Molva is a Professor at EURECOM. His research interests are the design and evaluation of protocols for security and privacy in selforganising systems. He was a Programme Chair or General Chair for security conferences such as ESORICS, RAID, SecureComm, IEEE ICC and security workshops. He is an Area Editor for the *Computer Networks Journal, Computer Communications Magazine* and the *Pervasive and Mobile Computing Journal*. He worked in the Zurich Research Laboratory of IBM as one of the Key Designers of the KryptoKnight Security System.

1 Introduction

Opportunistic networking (The Haggle Project, 2006; Hui et al., 2005) is a new paradigm aiming at enabling communication through highly heterogeneous networks by relying on dynamic transmission of messages whenever communication opportunities arise: such communication thus does not rely on a routing infrastructure and is peer-to-peer in essence. The delay-tolerant paradigm is a suitable approach to address the lack of connectivity and the mobility akin to opportunistic networks. In opportunistic networks, mobility and disconnections are the rule rather than the exception, therefore opportunistic networks are delay-tolerant by nature. The lack of end-to-end connectivity is a key difference between such networks and mobile ad-hoc networks (MANETs). This major constraint implies that it is impossible to establish an end-to-end path from source to destination and forwarding decisions are only based on a local view of the network.

Furthermore, opportunistic networks are more general than MANETs, because dissemination communication is the rule rather than conversational communication. A concept that nicely fits with the dissemination networking model is offered by content-based communication (Carzaniga and Wolf, 2003; Carzaniga et al., 2004) whereby messages are forwarded from source to destinations based on their content instead of explicit addresses. In content-based applications nodes declare their interests through receiver advertisements and simply publish content that they wish to disseminate, rather than explicitly defining destination nodes for packets. Intermediate nodes set up and update their forwarding table based on the receiver advertisements, and take forwarding decisions implicitly by looking up published content in their forwarding table.

The flexibility of content-based opportunistic networks come on the other hand with a high cost in increased exposure in terms of data security. Security services and in particular key management should be revisited to reflect the characteristics of such networks; in particular security services should also be flexible and self-organised. Moreover, privacy protection is particularly challenging due to the content-based

messaging paradigm. The protection of the content with classical security mechanisms would indeed conflict with the forwarding functions since the latter rely on the very content that is being transmitted for their basic operations. An interesting idea to meet the privacy requirements of content-based forwarding in opportunistic networks consists of multiple layer commutative encryption (MLCE) that allows to perform secure operations on encrypted content as proposed in Shikfa et al. (2009a, 2009b). When using MLCE, one needs to encrypt the data with several layers of encryption corresponding to lr next hops. Such a solution therefore calls for an innovative key management scheme that should ensure local and self-organised security associations between a node and its neighbourhood: each node should share a key with all its neighbours that are less than lr hops away. The key management should thus depend heavily on the neighbourhood topology which is fundamental for the multi-layer encryption scheme to work properly. Because of the lack of infrastructure and the peer-to-peer nature of the network, this also means that the neighbourhood topology itself should be securely discovered.

The main goal of this article is therefore to analyse the challenges raised by key management in order to come-up with a dedicated key management solution. This solution should feature local, self-organised and topology-dependent bootstrapping of security associations along with a secure neighbourhood discovery. In order to optimise the performance of the scheme, and to cope with the dependency between topology and security, it is indeed more efficient to perform both neighbourhood discovery and security associations with all lr-hops neighbours 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 neighbourhood. Creation of bogus identities through Sybil attacks would then be a crucial threat against which our scheme is protected thanks to an off-line identity manager (IM) as presented in Shikfa et al. (2010). Compared with Shikfa et al. (2010), this article includes new developments concerning the security evaluation of the solution and a more precise evaluation of the performance of the scheme.

In this article, we first analyse 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-organised and local mechanism that bootstraps security associations with the discovery of the neighbourhood topology thanks to the use of certificates and signatures chains. The proposed scheme relies on two phases: a first step where nodes are connected to an IM that provides them with unique pseudonyms, and a second step where the opportunistic communication takes place and where there is no need for the IM. The pseudonyms are not used as certified identities but only serve the purpose of withstanding Sybil attacks.

2 Problem statement

In this section, we first present the privacy issues pertaining to content-based communication in opportunistic networks. We then define the security requirements of a key management protocol in these specific peer-to-peer networks and present the threat model that we consider.

2.1 Privacy in content-based opportunistic networks

As mentioned in the introduction, content-based forwarding solutions raise entirely new privacy concerns: since nodes may not want to reveal the content of packets to entities other than destination(s), forwarding decisions should be taken over encrypted information. Shikfa et al. (2009a) propose an interesting approach to meet the conflicting requirements between forwarding and privacy in content-based opportunistic networks. The idea of this approach is to use multiple commutative encryption layers in order to ensure end-to-end confidentiality: packets are encrypted with multiple keys where each of them is shared by a different pair of nodes. Thanks to this scheme, intermediate nodes securely compare published content and encrypted interests on the fly.

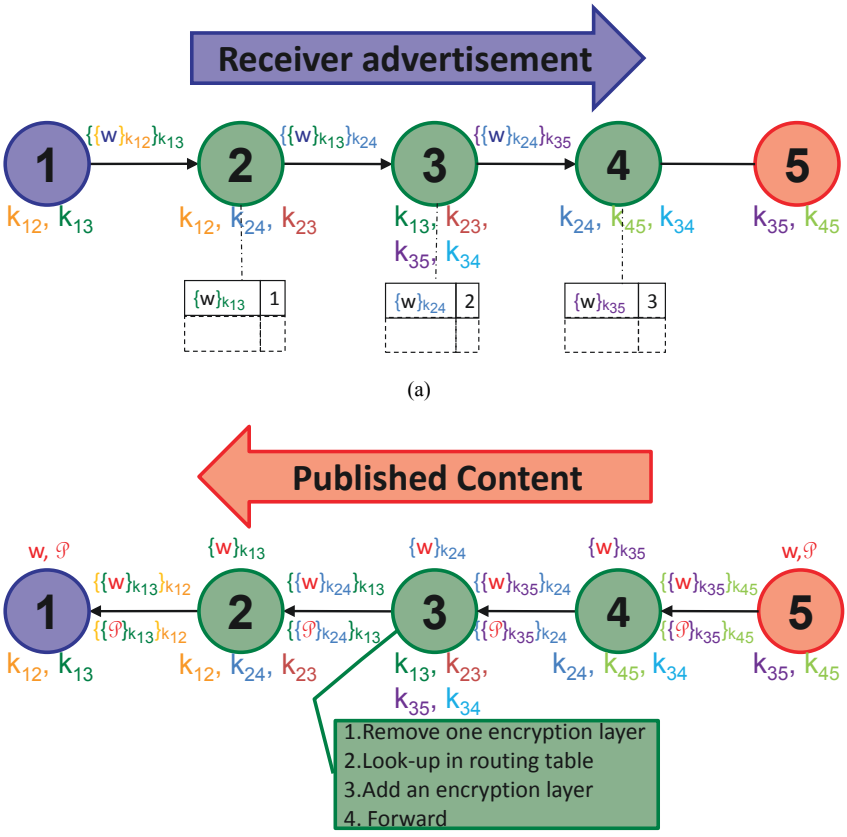
Even though it is impossible to establish an end-to-end path between source and destination, nodes can determine the lr next hops with a local knowledge of the network. Each node establishes a secure channel with nodes that are lr hops away. Moreover, the proposed scheme is commutative in the sense that $\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$, thus layers can be removed in any order.

While sending a new packet, the source first encrypts it lr times with lr different keys, each of them being shared with one of the lr next hops. Thanks to the commutativity of the encryption scheme, whenever an intermediate node receives an encrypted packet, it first removes one layer of encryption, and then compares the encrypted form of receiver advertisements and published content to take a forwarding decision. Before forwarding the packet, in order to ensure the confidentiality at the same level, the same intermediate node adds another encryption layer using the key that is shared with its lrth next hop. By removing old encryption layers and replacing them with the new ones, the same confidentiality degree is always ensured based on only a local knowledge of the network. The security of the scheme of course strongly depends on this lr parameter: the content of a packet can only be discovered if lr consecutive nodes collude.

In order to illustrate this scheme, we define a simple network with five nodes and we set $lr = 2$. Whenever Node 1 wishes to send a receiver advertisement raising its interest on a keyword w, w is encrypted both with k_{12} and k_{13}. When Node 2 receives this packet, it removes one encryption layer using k_{12}, updates its forwarding table with the partial information and finally adds another encryption layer using k_{24} before sending it to Node 3. Each node follows the same procedure. When Node 5 publishes some content with the same keyword w, the transitive and commutative properties of MLCE allow intermediate nodes to correctly forward this packet to Node 1. This scenario is summarised in Figure 1.

In Shikfa et al. (2009a, 2009b) where MLCE is proposed and described, the problem of key management is overlooked. We address this problem by analysing first the general requirements of key management in the context of opportunistic networks and then the more specific requirements of topology-dependent key management, in order to come up with a complete solution dedicated to MLCE.

Figure 1 Multiple layer commutative encryption overview: example with five nodes and $lr = 2$. Each node shares keys with its one and two hops neighbours (as shown below the nodes) (a) receiver advertisement propagation: receiver advertisements are keywords that are encrypted twice according to the keys shared with the next two hops. Intermediate nodes remove one encryption layer, build their routing tables with partially encrypted data, 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 (see online version for colours)



Notes: The payload and the keyword corresponding to the content are encrypted separately. 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.

2.2 Key management requirements

2.2.1 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:

key agreement can only be local. Furthermore, online centralised 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. (2007) 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-organised and should not depend on the identities of the nodes.

2.2.2 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 neighbourhood. Neighbourhood topology also needs to be discovered because of the lack of infrastructure.

The MLCE solution assumes indeed that the topology overlay used for content-based communication is a tree generated in a local and decentralised way. This implies that each node is aware of its lr-hop neighbourhood topology. Securely discovering the neighbourhood topology is yet a non-trivial task which in turn requires security services because nodes should guarantee their claimed hop-distance to their neighbours 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 (Hu et al., 2005a, 2002a) use cryptographic mechanisms and assume that nodes already own verifiable keying materials (e.g., identity certificates).

Hence there is a cyclic dependency between secure neighbourhood discovery and key management in MLCE similar to the dependency cycle between secure routing and security services in MANETs identified in Bobba et al. (2003). In order to take into account this dependency between network topology and security, and in order to avoid running two separate protocols, one for neighbourhood discovery and one for local key management, security associations should thus be locally bootstrapped along with a lightweight neighbourhood discovery solution.

2.3 Threats

2.3.1 Generic attacks

In order to bootstrap security associations and discover the neighbourhood 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 analysed in this article.

2.3.2 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 (Douceur, 2002) 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 summarise, content-based opportunistic networking requires a local and self-organised 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 neighbourhood discovery. Finally, as with any regular protocol, the new key management protocol should be protected from regular network attacks.

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 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 IM anymore and the long term keys are not used by the application. We hereafter describe these two phases in detail.

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 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 fulfil 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.

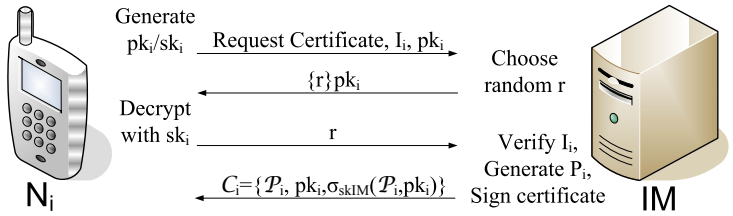
The IM then provides N_i with a certificate $\mathcal{C}_i$ which links the public key of N_i with its pseudonym, by signing these information:

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

The information exchange protocol between the IM and a node N_i is presented in Figure 2. 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.

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.

Figure 2 Summary of the information exchange protocol with the IM (see online version for colours)



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.

3.2 Bootstrapping local security associations

We now assume that all nodes have already performed the setup phase and own pseudonym certificates as mentioned in the previous section.

During the second phase nodes need to establish ephemeral security associations by sharing keys with all their neighbours which are at distance less than lr hops. As mentioned previously, this key agreement depends on the local topology and therefore requires a secure neighbourhood discovery. In order to optimise the number of message exchanges and to cope with the dependency between security and topology, we propose a local key agreement protocol along with neighbourhood discovery: one protocol run provides the initiator with both a correct view of its neighbourhood topology at lr hops distance and shared secrets with all lr-hops or less neighbours in a batch. On the one hand, the neighbourhood discovery mechanism is inspired by secure routing protocols (like Hu et al., 2005b) 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 the Diffie-Hellman key agreement protocol, also called the station to station protocol (Diffie et al., 1992). 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 article 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 neighbours 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 1.

Table 1 Example of security association bootstrapping

N_1 initiates security association request	
N_1	Randomly chooses $r_1 \in \mathbb{Z}^+_{\|G\|}$
	$\sigma_1 = signature_{sk_1}(SARq, 3, \{\mathcal{C}_1\}, \{g^{r_1}\}, \{\})$
$N_1 \rightarrow *$	$< SARq, 3, \{\mathcal{C}_1\}, \{g^{r_1}\}, \{\sigma_1\} >$
Processing of security association request by intermediate nodes	
N_2	Verifies σ_1 and randomly chooses $r_2 \in \mathbb{Z}^+_{\|G\|}$ 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 $r_3 \in \mathbb{Z}^+_{\|G\|}$ 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 $r_4 \in \mathbb{Z}^+_{\|G\|}$ 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

Notes: The initiator N_1 discovers its 3-hop neighbourhood and establishes security associations with three nodes. The underlined font indicates changed message fields, relative to the previous message of the same type.

3.2.1 Initiation of security association request

When a node N_s wants to establish security associations with its neighbours, 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 neighbours. 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 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

$$\sigma_s = signature_{sk_s}(SARq, lr, \mathcal{C}_s, g^{r_s}).$$

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 does 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.

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:

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

$SARp$ 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 $Certificate\_list$.

3.2.4 Key computation

When the reply finally gets back to the initiator of the neighbourhood 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 neighbours 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 neighbourhood.

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 neighbourhood topology and establish security associations with all the nodes in this neighbourhood.

3.3 Summary

The complete mechanism enables to bootstrap security associations along with neighbourhood 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

neighbourhood. The proposed mechanism is local and self-organised 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.

4 Evaluation

In this section, we evaluate the security and performance of the proposed scheme.

4.1 Evaluation of the setup phase

This setup phase, whereby nodes communicate with the IM 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 authorised 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 IM is not required anymore and the proposed scheme enables local and self-organised security associations.

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 analyse the security aspects of this feature.

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 (Stinson, 1995). 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.

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 the man-in-the-middle attack. 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.

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 neighbourhood 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.

4.2.4 On the modification of the STS protocol

As explained in Section 3.2, our protocol for establishing security associations is a modified version of the STS protocol (Diffie et al., 1992). 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 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, Krawczyk et al. (2003) 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 et al. (2003) proposed an alternative scheme called SIGMA (for SIGn and MAc) 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 Krawczyk (2003) 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.

4.3 Evaluation of the neighbourhood discovery mechanism

We now analyse the security of the second main goal of the proposed scheme, which is to securely discover the neighbourhood 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 neighbourhood discovery hops in a greyhole attempt (Hu et al., 2005a); 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 greyhole 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 (Hu et al., 2003) that completely circumvent the deleted nodes and avoid message discarding can be successful and the source node would end up with a fake neighbourhood 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 neighbours by using distance bounding techniques (Capkun and Hubaux, 2006; Shokri et al., 2009), which further mitigates the wormhole threat.

4.4 Performance evaluation

The scheme requires asymmetric cryptography and signature computations to guarantee the local neighbourhood topology. Nevertheless, the design of the mechanism takes into account the need to minimise 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 (ANSI, 2005)]. Tillich and Großschädl (2004) compare the execution time of RSA and ECDSA signatures on various mobile phones. 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 1 GHz Snapdragon processor). 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 smaller 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.

5 Related work

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: Farrell (2007) mentions some requirements of key management in DTN but no solution is proposed, and Asokan et al. (2007) evaluate ID-based cryptography in the context of DTN, but this solution is not suitable for content-based forwarding as mentioned in Section 2.2.1. In the broader area of peer-to-peer key management in MANETs many solutions have been proposed (Van Der Merwe et al., 2007). These solutions can be classified in two main categories:

- Fully self-organised key management, which have been first proposed in Capkun et al. (2003), and further studied in Cagalj et al. (2006), Capkun et al. (2006) and McCune (2009). 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 organised schemes are

inherently vulnerable against Sybil attacks, which is a major issue for MLCE (see Section 2.3.2). Therefore, fully self-organised 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 (Khalili et al., 2003; Wu et al., 2007; Xu and Iftode, 2004; Yi and Kravets, 2002; Zhou and Haas, 1999) or fully (Kong et al., 2001; Luo et al., 2002; Joshi et al., 2005). 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-organised and does not require an additional online authority, therefore it meets the DTN requirements.

5.2 Secure neighbourhood discovery

Secure neighbourhood discovery amounts to secure routing with a fixed number of hops instead of a given destination. Most existing secure routing solutions for MANETs [Ariadne (Hu et al., 2005b), SEAD (Hu et al., 2002b), SRP (Papadimitratos and Haas, 2003)] 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 (Perrig et al., 2002) 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 analysed in Bobba et al. (2003). The authors propose to break the dependency cycle 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 neighbourhood discovery securely thanks to certificates with unique pseudonyms provided by an offline IM. Our approach is therefore close to ARAN (Sanzgiri et al., 2002) 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 is online. Furthermore, ARAN requires signatures on route requests and replies which represents a non-negligible added cost, and ARAN does not use hop-count and can therefore not be used for neighborhood discovery.

6 Conclusions

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-organised and local. This locality also involves a correct view of the neighbourhood topology. We therefore designed a complete solution that enables bootstrapping of security associations along with secure neighbourhood 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 neighbourhood's topology and withstands tampering by malicious nodes. We also proposed the use of an IM 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-organised.

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-organised key management.

Acknowledgements

The authors would like to acknowledge the many helpful suggestions of the three anonymous reviewers and the participants of the 2010 PerCom Conference on earlier versions of this paper. We also thank the editor of this journal.

References

ANSI (2005) 'X9.62:2005 – public key cryptography for the financial services industry, the elliptic curve digital signature algorithm (ECDSA)', American National Standards Institute, November, available at http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.62%3A2005.

Asokan, N., Kostiainen, K., Ginzboorg, P., Ott, J. and Luo, C. (2007) 'Towards securing disruption-tolerant networking', Technical report NRC-TR-2007-007, Nokia Research Center, March, available at http://research.nokia.com/files/NRC-TR-2007-007.pdf.

Bobba, R.B., Eschenauer, L., Gligor, V. and Arbaugh, W. (2003) 'Bootstrapping security associations for routing in mobile ad-hoc networks', in *GLOBECOM '03: Proceedings of the 2003 IEEE Global Telecommunications Conference*, December.

Cagalj, M., Capkun, S. and Hubaux, J-P. (2006) 'Key agreement in peer-to-peer wireless networks', *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, Vol. 94, No. 2.

Capkun, S. and Hubaux, J-P. (2006) 'Secure positioning in wireless networks', *IEEE Journal on Selected Areas in Communications*, February, Vol. 24.

Capkun, S., Buttyán, L. and Hubaux, J-P. (2003) 'Self-organized public-key management for mobile ad hoc networks', *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, pp.52–64.

Capkun, S., Hubaux, J-P. and Buttyan, L. (2006) 'Mobility helps peer-to-peer security', *IEEE Transactions on Mobile Computing*, Vol. 5, No. 1, pp.43–51.

Carzaniga, A. and Wolf, A.L. (2003) 'Forwarding in a content-based network', in *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August, pp.163–174.

Carzaniga, A., Rutherford, M.J. and Wolf, A.L. (2004) '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.

Diffie, W., Van Oorschot, P.C. and Wiener, M.J. (1992) 'Authentication and authenticated key exchanges', *Designs, Codes and Cryptography*, Vol. 2, No. 2, pp.107–125.

Douceur, J.R. (2002) 'The Sybil attack', in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Springer-Verlag.

Farrell, S. (2007) *DTN Key Management Requirements*, June.

Hu, Y-C., Johnson, D.B. and Perrig, A. (2002a) '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*.

Hu, Y-C., Johnson, D.B. and Perrig, A. (2002b) '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.

Hu, Y.C., Perrig, A. and Johnson, D.B. (2003) '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*, IEEE Societies, April, Vol. 3, pp.1976–1986.

Hu, Y-C., Perrig, A. and Johnson, D.B. (2005a) 'Ariadne: a secure on-demand routing protocol for ad hoc networks', *Wireless Networks*, Vol. 11, Nos. 1–2, pp.21–38.

Hu, Y-C., Perrig, A. and Johnson, D.B. (2005b) 'Ariadne: a secure on-demand routing protocol for ad hoc networks', *Wireless Networks*, January.

Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J. and Diot, C. (2005) 'Pocket switched networks and human mobility in conference environments', in *WDTN '05: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, ACM.

Joshi, D., Namuduri, K. and Pendse, R. (2005) 'Secure, redundant, and fully distributed key management scheme for mobile ad hoc networks: an analysis', *EURASIP Journal on Wireless Communications and Networking*, September.

Khalili, A., Katz, J. and Arbaugh, W.A. (2003) '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.

Kong, J., Zerfos, P., Luo, H., Lu, S. and Zhang, L. (2001) '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.

Krawczyk, H. (2003) '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*, Springer, pp.400–425.

Luo, H., Zerfos, P., Kong, J., Lu, S. and Zhang, L. (2002) 'Self-securing ad hoc wireless networks', in *ISCC'02: Proceedings of the IEEE Seventh International Symposium on Computers and Communications*, IEEE Computer Society.

McCune, J.M., Perrig, A. and Reiter, M.K. (2009) 'Seeing-is-believing: using camera phones for human-verifiable authentication', *International Journal of Security and Networks*, Vol. 4, Nos. 1/2, pp.43–56.

Papadimitratos, P. and Haas, Z.J. (2003) 'Securing mobile ad hoc networks', *The Handbook of Ad Hoc Wireless Networks*, pp.551–567.

Perrig, A., Canetti, R., Tygar, J.D. and Song, D. (2002) 'Efficient and secure source authentication for multicast', *ICNP'02: Proceedings of the 2002 IEEE International Conference on Network Protocols*, November.

Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C. and Belding-Royer, E.M. (2002) '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.

Shikfa, A., Önen, M. and Molva, R. (2009a) 'Privacy in content-based opportunistic networks', in *WAINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, IEEE Computer Society, pp.832–837.

Shikfa, A., Önen, M. and Molva, R. (2009b) 'Privacy-preserving content-based publish/subscribe networks', in *Emerging Challenges for Security, Privacy and Trust: Proceedings of the 24th IFIP TC 11 International Information Security Conference, SEC 2009*, Springer Boston, May, pp.270–282.

Shikfa, A., Önen, M. and Molva, R. (2010) 'Bootstrapping security associations in opportunistic networks', in *PERCOM '10: Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications*, IEEE Computer Society.

Shokri, R., Poturalski, M., Ravot, G., Papadimitratos, P. and Hubaux, J-P. (2009) 'A practical secure neighbor verification protocol for wireless sensor networks', in *WiSec '09: Proceedings of the Second ACM Conference on Wireless Network Security*, ACM, pp.193–200.

Stinson, D.R. (1995) *Cryptography: Theory and Practice*, CRC Press.

The Haggle Project (2006) Available at http://www.haggleproject.org/index.php.

Tillich, S. and Großschädl, J. (2004) 'A survey of public-key cryptography on j2me-enabled mobile devices', in *Computer and Information Sciences – ISCIS 2004*, Springer-Verlag, pp.935–944.

Van Der Merwe, J., Dawoud, D. and McDonald, S. (2007) 'A survey on peer-to-peer key management for mobile ad hoc networks', *ACM Computer Surveys (CSUR)*, Vol. 39, No. 1.

Wu, B., Wu, J., Fernandez, E.B., Ilyas, M. and Magliveras, S. (2007) 'Secure and efficient key management in mobile ad hoc networks', *Journal of Network and Computer Applications*, Vol. 30, No. 3.

Xu, G. and Iftode, L. (2004) 'Locality driven key management architecture for mobile ad-hoc networks', in *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, IEEE, October.

Yi, S. and Kravets, R. (2002) 'Key management for heterogeneous ad hoc wireless networks', in *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, IEEE Computer Society, pp.202–205.

Zhou, L. and Haas, Z.J. (1999) 'Securing ad hoc networks', *IEEE Network Magazine*.