Privacy in Content-Based Opportunistic Networks

Abdullatif Shikfa
EURECOM
France
Email: Abdullatif.Shikfa@eurecom.fr

Melek Önen
EURECOM
France
Email: Melek.Onen@eurecom.fr

Refik Molva
EURECOM
France
Email: Refik.Molva@eurecom.fr

Abstract—**In this paper, we present security primitives required to achieve privacy in content-based opportunistic networks. We define three privacy models adapted to content-based networking and detail what are the requirements that the security primitives have to achieve in order to fit in each of these models. We also propose an original approach based on multiple layer commutative encryption that features full privacy content-based networking.**

I. Introduction

Opportunistic networks are based on a novel communication paradigm that aims at overcoming the limitations of communication services built upon the widely used concept of end-to-end connectivity. Indeed, users have nowadays "islands of end-to-end connectivity" at home, at the office or in hotspots. However, they are also likely to sporadically be in range of many other users while in between and, in spite of enjoying ever increased connectivity, they cannot benefit of end-to-end communications over several different technologies at a time.

Opportunistic and autonomic networking ([4]) is designed to solve the problem of communication in the presence of intermittent network connectivity, and, to this end, has the following requirements:

- **relaxed end-to-end connectivity:** opportunistic networking aims at transmitting a message over any communication medium available. To achieve such a goal, forwarding decisions are taken on-the-fly so that packets eventually reach their destination but establishing an end-to-end path is not possible.
- **collapsed architecture:** in order to benefit from various communication architectures, packets created to take advantage of opportunistic networking 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 ([2], [3]) 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.

Privacy is a crucial issue in content-based networking. Advertisements and published content are namely forwarded through various intermediate nodes that may not be trusted by publishers or receivers; moreover, trust relationship are loose in such a heterogeneous environment. Receivers do not want any other node (especially untrusted ones) to know what their interests are because these information threaten their privacy. Thus, nodes should be able to correctly build their forwarding tables based on encrypted advertisements and they further should correctly forward encrypted content based on these forwarding tables. Hence, nodes require mechanisms that allows to take content-based forwarding decisions without accessing the content in clear. In [5], Lilien et al. present the challenges in privacy and security of opportunistic networks but, to the best of our knowledge, we are the first to study the problems of privacy in content-based opportunistic networks. The main contributions of this paper are the following:

- We present security primitives required to achieve privacy in content-based networks. We define three privacy models adapted to content-based networking and detail what are the requirements that the security primitives have to achieve in order to fit in each of these models,
- We propose an original design that features complete content-based opportunistic networking with strong privacy enforcement.

In the next section, we first describe the opportunistic content-based network model, and derive from it two main security primitives required to preserve privacy. In section III, we formally define three privacy models and then detail them regarding the two security primitives. Section IV analyzes two basic approaches that attempt to solve the problem of privacy enforcement in content based networking whereas section V presents our original scheme.

II. Problem statement

A. Reference model

Content-based networks are usually represented by a non-oriented graph. Nodes are either:

- receivers, in which case they are denoted by A_i,
- intermediate nodes, which are denoted by B_k,
- publishers denoted by C_j.

Receivers A_i send their interest as receiver advertisements RA_i toward the network, publishers C_j publish some content

This work has been supported by the HAGGLE and SOCIALNETS projects, grant agreement number 27918 and 217141, funded respectively by the EC sixth framework program theme FP6-IST-2004-2.3.4 for Situated and Autonomic Communications and the EC seventh framework programme theme FP7-ICT-2007-8.2 for Pervasive Adaptation. See http://www.haggleproject.org/ and http://www.social-nets.eu/ for further details.

PC_j, and finally intermediate nodes B_k forward both advertisements and published content thanks to their forwarding table FT_k. Receiver advertisements are simple keywords (like "restaurant" for example) whereas published content is composed of two parts: control information (a simple keyword as well) and the payload itself (possibly long and complex). Only the control information, that we denote by CI_j, are relevant in the forwarding process.

It is worth noting that this classification of nodes in three categories is purely functional: it is, indeed, possible that one node assumes the three roles depending on the communication that is undergoing.

Content-based applications do not define explicit destination for packets, whether receiver advertisements or published content. Intermediate nodes only use the keywords to take correct forwarding decisions to the next hop, there is no end-to-end route. Hence, receivers advertise their interest and intermediate nodes update their forwarding table based on these receiver advertisements. This way, when an intermediate node receives a published content it simply has to look-up for a match between the control information of the published content and its forwarding table. When such a match is found the content is forwarded accordingly.

In the next section, we detail the privacy issues of content-based forwarding and the primitives required to solve them.

B. Security primitives

Since advertisements and published content are forwarded through opportunistic intermediate nodes that are not necessarily trusted, security has to be enforced with several operations. First, receiver advertisements have to be encrypted to enforce privacy. However, encrypted advertisements should be forwarded toward the network and intermediate nodes should be able to build forwarding tables based on these encrypted information. Therefore such applications require a dedicated encryption operation that allows some networking operations over encrypted data. Furthermore, in order to optimize bandwidth usage, similar advertisements should first be aggregated and forwarded into a single packet. Therefore, intermediate nodes should be able to first compare encrypted packets and aggregate them into one packet if they are equivalent and finally forward this single packet.

Similarly, a content publisher may encrypt the published content for privacy purposes. In this case, the encryption operation performed by the publisher should also provide some similar properties as the one for advertisements. Indeed, whenever a content is received, whether it is encrypted or not, an intermediate node should be able to take a forwarding decision over this content based on its forwarding table.

To summarize, forwarding decisions are directly taken over the content of the packet but content publishers or receivers may not wish to reveal this content to some intermediate nodes whose only task is forwarding. In order to ensure networking together with security, intermediate nodes require two main secure forwarding primitives:

- **secure setup of forwarding tables**: in order to correctly forward packets, nodes must construct their forwarding tables based on encrypted receivers' advertisements;
- **secure look-up**: based on its forwarding table, an intermediate node must be able to take correct forwarding decisions whenever it receives encrypted content.

The design of these two security primitives can differ with respect to the application security requirements and mainly with respect to the level of privacy. The mechanisms required to achieve a certain level of privacy inherently depend on the level of trust between intermediate nodes on the one hand, and receivers or content publishers on the other hand. A trusted node is defined as a node which can access the content of the data (even if it is not the destination) whereas an untrusted node should deal with it only in an encrypted way. In other words, data privacy has to be enforced against untrusted nodes but not against trusted ones.

In the next section, we define several privacy models and the requirements for forwarding primitives to fit in each model.

III. Security definitions

A. Privacy models

As explained in the previous section, the design of the two secure forwarding primitives depends on the level of privacy required by the application. A content publisher or a receiver may or may not want to reveal some content or some interests respectively to the intermediate nodes. After analyzing several different scenarios in content-based applications, we came up with three main privacy models:

- **model 1, privacy oblivious**: this model refers to the case where publishers and receivers do not require privacy at all. Therefore, information is simply sent in clear and intermediate nodes proceed as in standard content-based applications.
- **model 2, intra-community privacy**: in this model, the level of privacy depends on nodes' relationship: some intermediate nodes may be trusted and some others not. The trust relationship can for example be based on some community membership. In this case, members of the community can access the content of the packet and proceed as in standard content-based application but other nodes cannot access it. Hence communication between community members should be encrypted, but the two forwarding primitives are performed on the cleartext.
- **model 3, full privacy**: as opposed to model 2, this model refers to the case where nodes do not trust any other node. Therefore, intermediate nodes should be able to process encrypted packets without having access to the content of these packets.

In opportunistic networks, messages are forwarded through different sub-networks with different characteristics, hence it is possible to consider hybrid models for such paths. We limit ourselves to these three main models in this paper, and think about transitions between them as future work.

In the next section, we discuss the design of the two secure forwarding primitives that are secure setup of forwarding tables and secure look-up based on these three privacy models.

B. Privacy-aware setup of forwarding tables and look-up

In order to correctly and efficiently perform network operations, an intermediate node B_k first needs to build a forwarding table, denoted by FT_k based on the received RA_i and further uses this table to take forwarding decisions whenever it receives a content PC_j. The design of these two primitives strongly depends on the privacy models described in the previous section. We therefore analyze these two problems for each of these models.

- **model 1, privacy oblivious**: when no privacy is required at all, both RA_i and CI_j are received by intermediate nodes in clear. In this case, the building of forwarding tables FT_k and the look-up operations are the classical ones used in content based networking. Therefore, whenever B_k receives a RA_i, it first looks if such an entry or an equivalent one exists in its forwarding table. If this is not the case, then B_k adds an additional row in its table as follows: $$RA_i \rightarrow A_i.$$ If, on the other hand, B_k finds an equivalence between the received RA_i and another one in its forwarding table denoted by $RA_{i'}$ then B_k aggregates this information and updates the row corresponding to $RA_{i'}$ as follows: $$(RA_i \Leftrightarrow RA_{i'}) \rightarrow A_i, A_{i'}.$$ This aggregation operation is a very important optimization from a performance point of view. Once the forwarding table FT_k is updated, B_k can propagate the aggregated advertisement toward the network and correctly make forwarding decisions whenever it receives a packet PC_j. Indeed, the look-up operation consists in comparing the control information CI_j of PC_j with each row in its forwarding table in order to define the next hop for the packet. This case with no privacy can be used as a witness case.
- **model 2, intra-community privacy**: in this model, recipients and publishers only trust B_k if they belong to the same community. In this case, B_k is able to decrypt any packet originating from members of its community. For example, suppose that A_1 and B_1 belong to the community ($community1$). A_1 sends its encrypted interest RA_1 to B_1, such that only members of $community1$ can decrypt it. Other nodes, like A_2 for example, cannot discover A_1's interest, but B_1 is able to setup its forwarding table exactly like in model 1 because it can decrypt RA_1. Similarly, when C_1 sends some encrypted data to B_1, if B_1 belongs to the same community, then it can have access to the control information CI_1 and perform a correct look-up without revealing any extra information to potential eavesdroppers.
- **model 3, full privacy**: in this model, every node becomes a potential adversary. This implies that A_i or C_j do not trust any intermediate node B_k and therefore they encrypt their advertisements or content packet respectively. To build its forwarding table, B_k should first be able to detect whenever two encrypted advertisements RA_i and $RA_{i'}$ are equivalent without decrypting them as opposed to the case in model 2. The only information that it should get from this process is the matching between them, it should never be able to get more information on the interests. This aggregation process is required to optimize the forwarding table but it imposes an additional challenge from a security point of view. Similarly, the content publisher encrypts its packet PC_j and B_k should be able to find whether the encrypted control information CI_j within this packet matches one of the encrypted entries of its forwarding table FT_k or not. Therefore, B_k always knows where to forward the packet without knowing neither the content of the message nor the corresponding advertisement.

Now that we have clearly defined the privacy models for each security primitive, we analyze some basic approaches to solve these problems, and then propose a complete privacy preserving approach.

IV. Basic approaches and their drawbacks

A. Hash functions

The first basic idea to solve these problems is to use a cryptographic hash function, as proposed by Propicman in [6]. A cryptographic hash function is a one-way collision resistant function. Receivers (A_1 and A_2) hash their advertisements using a public hash function h and send them to the intermediate node B. B receives $h(RA_1)$ and $h(RA_2)$, compares them and if they are the same, he puts them in the same row of its forwarding table, otherwise he puts them in two different rows. B is therefore able to detect if RA_1 and RA_2 are equivalent or not without learning their actual value (because by the very definition of one-way hash functions, finding x given $h(x)$ is difficult). When C wants to send a message, he also performs a hash function over the control information before sending it. B receives $h(CI)$ from C and he has to do a look up in his forwarding table. This operation can be done directly on hashed values. B can then perform the secure look-up and forward the message as indicated by its forwarding table without accessing the hidden information.

The idea looks seducing and efficient, it almost achieves privacy model 3 and its cost is very low. Yet, it is not secure against dictionary attacks. Since the hash function is public and no secret information is required, any node, including A_1, A_2, B, C or an attacker, can compute the hash of any value. Since the messages are well formated and they have a meaning (which is very different from a pseudo-random sequence), B or another attacker could simply compute the hashes of all words of a dictionary and then identify these hashes with the hashed value exchanged during the protocol. This attack is quite cheap and can easily and efficiently be launched by any node, thus breaking the confidentiality of hashed values. In fact, this method does not even achieve model 2 because of this simple attack.

B. Group security

Another idea is to use group key cryptography in order to achieve intra-community privacy. In this case all nodes belonging to a given community are given a common key, that we call community key. For example, let us suppose that A_1, B and C belong to $community1$ and thus share key k_1. Then, A_1 sends $E_{k_1}(RA_1)$ to B which can decipher it and access RA_1 as opposed to attackers which are not members of $community1$. When C wants to send its message, it sends $E_{k_1}(CI)$ to B which can then decipher it and perform the look-up operation in a classical way and forward the message afterwards. Eavesdroppers have no access to information since it is encrypted but members of the community have access to all information. For example, A_1 can directly decipher $E_{k_1}(CI)$ if she catches it, but this is normal since A_1 and C trust each other.

This mechanism fulfills the goal of model 2 in terms of privacy, but it has some disadvantages. First of all, group key cryptography implies heavy key management to build the groups, add members or revoke some of them. Such an administrative burden should be taken care of and might not be available depending on the network capabilities. Another problem which is inherent to model 2 is that communication occurs only between community members; other nodes are completely excluded. Unless in the case of a controlled environment, this method is therefore not standalone, but it can be used in a hybrid protocol.

V. A preliminary solution: privacy with multiple encryption

A. General idea

We now present an original approach, which solves the problem of full privacy content-based forwarding.

The basic idea behind our solution is to use a Multiple Layer Commutative Encryption (MLCE) in order to meet the privacy requirements of content-based forwarding. MLCE allows intermediate nodes in charge of routing secure traffic to perform secure transformations without having access to the processed data. This feature of MLCE lends itself very well to solving the problem of routing encrypted data.

In multiple layer encryption, data is encrypted several times with different keys. The idea is for a receiver to encrypt its receiver advertisement with $r \geq 2$ layers corresponding to the r next hops using r different keys, and for the publishers to do the same with their published content. An intermediate node B_k en-route can remove only one encryption layer so that the data is always protected by at least $r-1$ layers of encryption. Thus B_k does not have access to data in cleartext, but it performs the setup of forwarding tables and takes forwarding decisions on data encrypted $r-1$ times. Then B_k adds a new encryption layer corresponding to the r^{th} next hop without destroying the other layers and transmits the message.

In order to be able to add and remove layers in any order, the encryption layers all have to use the same cryptosystem, and this cryptosystem has to be commutative. An encryption mechanism E is commutative if, for any data d, any keys k_1, k_2 we have :

$$E_{k_2}(E_{k_1}(d)) = E_{k_1}(E_{k_2}(d)).$$

In the sequel of the paper we denote by E_k a commutative cipher which encrypts its input under key k, and D_k is the corresponding decryption operation. Examples of such ciphers are one-time pad XORs [10], RSA [9] or the Pohlig Hellman cryptosystem [8].

Thanks to the commutativity of the layers, MLCE allows secure look-up as well as setup of efficient and secure forwarding tables based on encrypted data. Furthermore, the source of a packet does not need to pre-establish an end-to-end secure communication with the destinations, it does not even need to know ultimately who the destinations are. This interesting property reflects the philosophy of content-based opportunistic applications and is possible thanks to the flexibility of MLCE.

In order to illustrate our solution we define a simple scenario with one publisher (C), four intermediate nodes (B_1 to B_4) and five receivers (A_1 to A_5). As we mentioned earlier, this classification is purely functional, and the associated logical tree is presented in figure 1. We assume that A_1 to A_4 share a common interest and advertise w while A_5 is interested in a different keyword w'. As for the publisher it wants to publish a content corresponding to w.

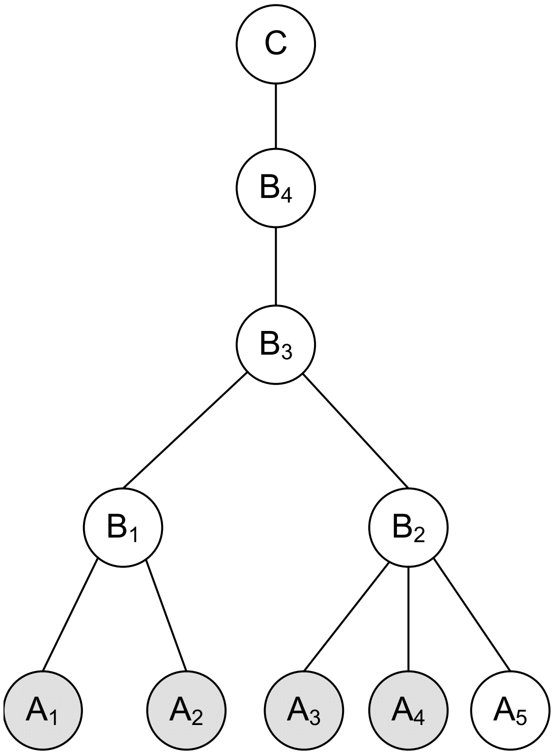

Fig. 1. Network used as illustration

B. Key management

The use of MLCE allows for a local, flexible and easy to manage key distribution. In our scheme, nodes only need to share keys needed for the addition or removal of r layers of encryption, hence each node only needs to have a local view of the network corresponding to the r-hops neighborhood. This scheme thus does not require end-to-end key management which fits the absence of end-to-end connectivity. This scheme is also flexible because a change in the neighborhood only requires to modify the key distribution locally and does not impact nodes which are far away. This scheme is also easy to manage because it does not require an additional security infrastructure like a specific secure channel to exchange keys or a trusted third party.

The number of layers r is a security parameter that has a performance impact, yet, for the sake of simplicity, we present our scheme only for the case $r = 2$ and discuss the choice of the parameter r in section V-E. In the sequel of the paper, we thus suppose that each node shares a key with each of its parent, grandparent, children and grandchildren when they exist. The keys shared between two nodes N and M are denoted by k_{MN} or k_{NM} indifferently. In figure 1, B_1 shares keys with its children A_1, A_2, its parent B_3 and its grandparent B_4 denoted respectively by $k_{B_1A_1}$, $k_{B_1A_2}$, $k_{B_1B_3}$ and $k_{B_1B_4}$. In the next sections we describe how our solution deals with the problems of building secure forwarding table and secure look-up.

C. Advertisements propagation and forwarding table building

When a receiver wants to advertise its interest it first adds two layers of encryption with the keys that it shares with its parent and grandparent. For example A_1 sends to B_1 the following: $[RA_{A_1} = E_{k_{B_1A_1}}(E_{k_{B_3A_1}}(w)); A_1]$.

This subscription filter is propagated upwards and each intermediate node removes one encryption layer, performs some operation to setup its forwarding table, adds a new encryption layer and finally forwards the message to the next hop. Table I illustrates the propagation of A_1's advertisement until the last hop before the content publisher.

A_1	w
$A_1 \rightarrow B_1$	$[E_{k_{A_1B_1}}(E_{k_{A_1B_3}}(w)); A_1]$
B_1	$E_{k_{A_1,B_3}}(w)$
$B_1 \rightarrow B_3$	$[E_{k_{A_1B_3}}(E_{k_{B_1B_4}}(w)); A_1]$
B_3	$E_{k_{B_1,B_4}}(w)$
$B_3 \rightarrow B_4$	$[E_{k_{B_1B_4}}(E_{k_{B_3P}}(w)); B_1]$
B_4	$E_{k_{B_3,P}}(w)$

TABLE I
PROPAGATION OF A RECEIVER ADVERTISEMENT w FROM A_1 TO B_4

The operation performed by intermediate nodes consists in building their forwarding table in order to take appropriate forwarding decisions afterwards. The intermediate node also has to detect equivalent advertisements in order to aggregate them and optimize its forwarding table.

We take the example of node B_3 to illustrate the setup of the forwarding table FT_3. B_3 receives encrypted receiver advertisements corresponding to all receivers. Let us assume it first receives the message $[E_{k_{A_1B_3}}(E_{k_{B_1B_4}}(w)); A_1]$ from B_1. This message indicates that B_1 is interested in keyword $E_{k_{A_1B_3}}(E_{k_{B_1B_4}}(w))$ not for himself but for A_1. B_3 removes an encryption layer by applying $D_{k_{A_1B_3}}$ to the message and then stores in its forwarding table FT_3 a first row $E_{k_{B_1B_4}}(w) \rightarrow B_1(A_1)$ which means that in case B_3 receives content matching $E_{k_{B_1B_4}}(w)$ it has to forward it to B_1 and the next hop after B_1 is A_1. B_3 also forwards $[E_{k_{B_1B_4}}(E_{k_{B_3P}}(w)); B_1]$ to B_4.

When B_3 receives $[E_{k_{A_2B_3}}(E_{k_{B_1B_4}}(w)); A_2]$ it removes a layer by applying $D_{k_{A_2B_3}}$ to get $E_{k_{B_1B_4}}(w)$ which is equal to the first row of FT_3, hence it just updates the list of destination and does not forward the message. Hence after two hops, intermediate nodes are able to detect advertisements' equivalences and aggregate them in one row in their forwarding table.

When B_3 receives $[E_{k_{A_3B_3}}(E_{k_{B_2B_4}}(w)); A_3]$ it also removes one layer (with $D_{k_{A_3B_3}}$) to get $E_{k_{B_2B_4}}(w)$ which is different from the entry in its forwarding table and hence is not aggregated. A new row is added in FT_3 and the message $[E_{k_{B_2B_4}}(E_{k_{B_3P}}(w)); B_2]$ is forwarded to B_4. B_3 also receives messages from A_4 and A_5 and *in fine* its forwarding table is represented in Table II.

FT_{3_1}	$E_{k_{B_1B_4}}(w) \rightarrow B_1(A_1), B_1(A_2)$
FT_{3_2}	$E_{k_{B_2B_4}}(w) \rightarrow B_2(A_3), B_2(A_4)$
FT_{3_3}	$E_{k_{B_2B_4}}(w') \rightarrow B_2(A_5)$

TABLE II
FORWARDING TABLE FT_3 OF B_3

D. Content distribution and secure look-up

Now that the advertisement propagation process has been detailed, we similarly explain the content distribution algorithm. This algorithm roughly follows the advertisement process in the reverse path.

When C wants to publish a content $\mathcal{P}$ it first chooses a keyword that describes the content. This keyword is used as control information and in our example we suppose that this keyword is also w so that it matches some of the receiver advertisements. C then uses MLCE to encrypt the payload and the keyword in the same way as the receivers did for their advertisements, that is to say by adding two encryption layers with the key it shares with its child and grandchild. In the example, the message sent by C to its child B_4 is as follows:

$$[\overbrace{E_{k_{B_3C}}(E_{k_{B_4C}}(w))}^{Control\ \ information};\overbrace{E_{k_{B_3C}}(E_{k_{B_4C}}(\mathcal{P}))}^{payload}].$$

When an intermediate node receives the message, it uses only the control information to do the look-up and takes a forwarding decision. The payload and the control information are decrypted and re-encrypted with the multiple encryption system and follow the reverse path of advertisements so that they can easily be processed by intermediate nodes and eventually reach their destination.

For example when B_3 receives from B_4 the message $[E_{k_{B_3C}}(E_{k_{B_1B_4}}(w)); E_{k_{B_3C}}(E_{k_{B_1B_4}}(\mathcal{P}))]$, B_3 removes one encryption layer by applying $D_{k_{B_3C}}$ in the control information to get $E_{k_{B_1B_4}}(w)$ and then performs the look-up in FT_3. This look-up shows that this content corresponds to the first row of FT_3 and that two grandchildren are interested in it. Hence B_3 sends two messages to B_1, namely:

$$\begin{aligned} & [E_{k_{B_1B_4}}(E_{k_{A_1B_3}}(w)); E_{k_{B_1B_4}}(E_{k_{A_1B_3}}(\mathcal{P}))], \\ \texttt{and} \quad & [E_{k_{B_1B_4}}(E_{k_{A_2B_3}}(w)); E_{k_{B_1B_4}}(E_{k_{A_2B_3}}(\mathcal{P}))]. \end{aligned}$$

This concludes the processing done at B_3 for this packet. For a better understanding of the whole process from publisher

Step	Event notification
C	$[w, \mathcal{P}]$
$C \rightarrow B_4$	$[E_{k_{B_3C}}(E_{k_{B_4C}}(w)); E_{k_{B_3C}}(E_{k_{B_4C}}(\mathcal{P}))]$
B_4	$[E_{k_{B_3C}}(w); E_{k_{B_3C}}(\mathcal{P})]$
$B_4 \rightarrow B_3$	$[E_{k_{B_3C}}(E_{k_{B_1B_4}}(w)); E_{k_{B_3C}}(E_{k_{B_1B_4}}(\mathcal{P})]$
B_3	$[E_{k_{B_1B_4}}(w); E_{k_{B_1B_4}}(w)]$
$B_3 \rightarrow B_1$	$[E_{k_{B_1B_4}}(E_{k_{A_1B_3}}(w)); E_{k_{B1,B4}}(E_{k_{A_1B_3}}(\mathcal{P}))]$
B_1	$[E_{k_{A_1B_3}}(w); E_{k_{A_1B_3}}(\mathcal{P})]$
$B_1 \rightarrow A_1$	$[E_{k_{A_1B_3}}(E_{k_{A_1B_1}}(w)); E_{k_{A_1B_3}}(E_{k_{A_1B_1}}(\mathcal{P}))]$
A_1	$[w, \mathcal{P}]$

TABLE III
EVOLUTION OF A MESSAGE PUBLISHED BY C ON ITS PATH TO A_1

to receiver, table III illustrates the propagation of a published content related to keyword w from publisher C to receiver A_1.

In the next section, we analyze the security and the performance of the scheme.

E. Evaluation

In a work evaluating the security of cryptosystems in the multi-user setting [1], Bellare et al. have shown that if a cryptosystem is secure in the sense of indistinguishability, then the cryptosystem in the multi-user setting, where messages are encrypted using different keys, is also secure. Thus, the proposed scheme is at least as secure as a one layer encryption.

We now show that the proposed framework fits in the third level privacy model whereby every node becomes a potential adversary and thus intermediate nodes are not trusted. 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 or eavesdroppers at all times, even if these nodes share the same interest.

From a performance perspective, the proposed scheme only requires two encryptions at the receiver, one encryption and one decryption at intermediate nodes, and two decryptions at the publisher. The memory cost is related to the key distribution algorithm: each node shares a key with its parent and grandparent and a key with each of its children and grandchildren. This calls for a key distribution protocol which is well adapted to the opportunistic model because it can be performed locally and does not require an end-to-end setup.

Furthermore, thanks to the secure aggregation of advertisements, forwarding tables are also optimized. Indeed, any intermediate node is able to compare encrypted advertisements and discover equivalences in order to optimize its forwarding table, which also improves the performance of the look-up operation. This aggregation occurs only after two hops though, but this is a minor drawback.

In the proposed framework, the security mechanism presented relies on the use of two encryption layers in order to simplify its description. However it also means that if two consecutive nodes, a node and its parent, collude and hence share their own keying material, they can decrypt their children nodes' interest. In order to prevent such attacks, the number of encryption layers can be increased as described in [7]. Therefore, the privacy of the scheme depends on the choice of the number of encryption layers denoted by r. A larger value for r implies a larger number of nodes to collude to break it. However, if r is very large, then the number of keys stored at each node becomes very large and the key distribution protocol becomes less flexible. More importantly, the aggregation is possible only after r hops, hence the choice of r is a trade-off that depends on the scenario and the topology of the network.

To put it in a nutshell, this scheme enforces, at a very low cost, full privacy all the way since intermediate nodes (and even the root C) do not know what the final destination of the information (except the node before it) is, they just know in which direction to forward the packet.

VI. CONCLUSION & FUTURE WORK

In this paper we presented the analysis of privacy issues in content-based opportunistic networking. We defined three privacy models that adapt to different networking scenarios and achieve different levels of privacy. We also identified two main secure forwarding primitives which are necessary to privacy aware content-based networking operations, namely **secure look-up** and **setup of forwarding tables**, and we have detailed the requirements that each of these primitives should fulfill in order to fit in each privacy model.

Finally, we presented an original approach based on multiple encryption that achieves full privacy for content-based opportunistic networks. This scheme is based on multiple layer commutative encryption and preserves privacy of receivers very efficiently in a decentralized way, without need for end-to-end connectivity.

As a future work, we intend to develop this scheme by improving its flexibility regarding the advertisements' format. We would like indeed to extend receiver advertisements to encompass logical expressions of several interests.

REFERENCES

[1] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multiuser setting: Security proofs and improvements. In *Eurocrypt 2000*, pages 259–274. Springer Verlag, 2000.

[2] A. Carzaniga, M. J. Rutherford, and A. L. Wolf. A routing scheme for content-based networking. In *IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[3] A. Carzaniga and A. L. Wolf. Forwarding in a content-based network. In *SIGCOMM*, pages 163–174, 2003.

[4] Haggle project, 2006. http://www.haggleproject.org/index.php.

[5] L. Lilien, Z. Kamal, V. Bhuse, and A. Gupta. Opportunistic networks: The concept and research challenges in privacy and security. In *NSF Intl. Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN 2006)*, Miami, March 2006.

[6] H. A. Nguyen, S. Giordano, and A. Puiatti. Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman). *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–6, 18-21 June 2007.

[7] M. Önen and R. Molva. Secure data aggregation with multiple encryption. In *Wireless Sensor Networks, 4th European Conference, EWSN 2007*, Lecture Notes in Computer Science, pages 117–132, Delft, The Netherlands, 29-31 January 2007. Springer.

[8] 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.

[9] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[10] D. R. Stinson. *Cryptography: theory and practice*. CRC Press, Boca Raton, Florida, 1995.