Anonymous Communication in Multi Hop Wireless Networks

Volker Fusenig, Dagmara Spiewak, and Thomas Engel
University of Luxembourg
Faculty of Science, Technology and Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
Email: {Volker.Fusenig, Dagmara.Spiewak, Thomas.Engel}@uni.lu

In this paper we present a protocol for unlinkable and anonymous communication in multi hop wireless networks. There exist solutions for unlinkable communication in the internet, such as Tor and Mix networks. These protocols need fixed infrastructure and they are prone to traffic analysis attacks, i.e. where an attacker tries to map communicating devices by performing statistical analysis. Wireless communication eases such kind of attacks because the attacker can listen to a communication while residing in communication range of a device. Our protocol offers unlinkability and receiver anonymity without requiring a fixed infrastructure and keeps the computational and message overhead acceptable.

Keywords: Anonymity, Unlinkability, Wireless networks
ACM Classification: K.6.5

1. INTRODUCTION

There are several reasons for anonymous communication in computer networks. It is not only useful to guarantee the freedom of opinion but also to protect the privacy. This becomes more important the more parties are interested in private information on users. Companies use private information mainly for targeted advertising, where some governmental institutions collect information in the name of crime prevention (EU Commission, 2006). To give the users the ability to decide which information they want to reveal and which information they do not want to reveal they need the chance to communicate anonymously.

In the past there has been some work in the field of anonymous communication (details in Section 2). The most popular technique used for internet communication is Onion Routing for which there have been successful attacks (Bauer *et al*, 2007). The common ground for all anonymity protocols for internet communication is that they need infrastructure, such as onion routers.

In this paper we present a protocol that is designed for the usage in multi hop wireless networks. In this scenario we cannot assume that all nodes are permanently connected to fixed infrastructure as it is for internet communication. Besides this limitation the wireless communication also holds additional challenges: Every node in sending range of a node can listen to its communication. Encrypting this traffic can offer confidentiality and integrity. But it is still possible to determine who communicates to whom.

Manuscript received: 18 July 2008
Communicating Editor: Ljiljana Brankovic

The protocol we present in this paper is able to offer anonymity and unlinkability in such a scenario. Even a strong passive attacker that is able to observe all network communication on all nodes in the network is not able to detect the communication partners. The only chance for the attacker to reveal the communication partners is by taking over all nodes that build the anonymity or unlinkability set. To make the protocol applicable both the computational overhead during the execution and the message overhead has to be small.

The rest of the paper is organized as follows: In Section 2 we give an overview on related work in the area of anonymous communication. In Section 3 we introduce the definitions of anonymity and unlinkability and also the network and adversary model that we will use for the evaluation of the protocol. We present the Acimn protocol in Section 4. In the following Section 5 we analyze the performance of the protocol. We base the measurements of the computational overhead on the execution of a prototype implementation of Acimn. We study the security of Acimn in Section 6. Finally, we conclude our work in Section 7.

2. RELATED WORK

There are two basic strategies to achieve anonymity in computer networks: The use of *proxies* and *DC-nets*. The protocol introduced later is based on the usage of both techniques. We will present the functionality of these techniques in the next two subsections. Afterwards we give an overview of protocols for anonymous communication in multi hop wireless networks.

2.1 Proxies

An easy way to hide the sender of a message in internet communication is to use proxies. For this the sender sends its message to the proxy and not to the destination. The proxy changes the source address of the message to its own address and forwards the message to the destination. The backward communication works in the same way: The proxy receives messages from the destination node and forwards them to the sender node. With a proxy it is possible to hide the sender of a message to the receiver and it is also possible to hide the receiver node. But an attacker can easily break this anonymity by observing the incoming and outgoing traffic of the proxy.

A further development of a proxy is a mix. The first *mix* was designed by Chaum (1981) to hide the communication participants in an electronic mail system. A mix collects encrypted pieces of mail sent by different participants, modifies the source address and sends the decrypted pieces in a re-sorted order to the destination or to another mix. By this the attacker cannot easily match the incoming and outgoing traffic. But by observing the traffic over some time he might be able to break the anonymity by the use of traffic analysis attacks.

Based on this idea, several tools which provide anonymity were proposed (Danezis *et al*, 2003; Reiter and Rubin, 1998; Shields and Levine, 2000; Berthold *et al*, 2000). An improvement was provided by the introduction of *onion routing* (Goldschlag *et al*, 1996; Syverson *et al*, 2000). Onion routing adapts the mix protocol in such a way that it can handle any communication in the Internet. This requires that the protocol introduces only low latency. The cost for reducing the latency is that the anonymity is easier to break in onion routing. *Tor*, the *Second-Generation Onion Router* (Dingledine *et al*, 2004), is an enhancement of onion routing fixing some problems of the onion routing specification. In onion routing clients choose a path to build up a *circuit* over several nodes, called *Onion router*, where each node of the circuit only knows its predecessor and successor. The client constructs the path by negotiating a secret key with all nodes on the path. Afterwards the client can send messages over this path while he has to encrypt its messages one after another with the secret keys of all nodes on the path. If a node on the path receives such a message it can unwrap one layer of encryption and resends the message to the next node on the path. Because of the length of the path it is harder for an attacker to reveal the communication partners. Nevertheless there have been successful attacks on Tor (Bauer *et al*, 2007).

In wireless networks there exist two basic problems that make the usage of both mixes and onion routing ineffective. The first problem is that not every node has a direct communication link to a mix or Tor node. In the case of a mix, even if there is a connection from a node to the mix, it is still impractical for the following reasons. To guarantee unlinkability a mix needs several users in order to mix the messages of these users before resending them. Because in wireless network there is only a small number of potential users the mix has to wait for a long time before it received enough messages. The main problem of onion routing in multi hop wireless networks is that the attacker can observe every communication just by residing in communication range of a node. Because there is no mixing at the onion routers a strong passive attacker is able to follow the messages being forwarded over the communication path to the destination node.

2.2 DC-nets

The *DC-net* (*Dining Cryptographers network*) approach was developed by Chaum and provides untraceability for the sender. In Chaum (1988) he describes this technique that allows nodes of a group to communicate anonymously. The generalized approach works with any number of nodes in a group greater than two. If the group consists only of two nodes, only a node outside the group is unable to distinguish between the sender and the receiver of a message. In a group with more than two nodes neither a node inside nor outside the group is able to detect the sender or the receiver of a message.

To build up a DC-net group, each node has to share secret keys with every other node. Only one key bit is used in every round and the keys are only used once. Only one node is allowed to send in every round. Possible reservation techniques can be found in Bos and den Boer (1990) and Chaum (1988). In the i-th round each node calculates the sum modulo two of the i-th bit of all pairwise shared keys. The node *A*, who has reserved the current round, now sends the inverted sum modulo two if it wants to send a one, and the sum modulo two if it wants to send a zero to all other nodes. All other nodes send the sum modulo two. After receiving the bits of all nodes, each node can build the sum modulo two of the received bits. Because all secret keys are used twice, the sum modulo two results in the bit sent by *A*.

We give the calculations with a group of three nodes N_1, N_2, and N_3 where N_1 and N_2 share the secret key bit S_1, N_1 and N_3 the bit S_2, N_2 and N_3 the bit S_3. Node N_3 reserves the current round and wants to send the message bit M. Each node N_1 calculates the bit $P_i (1 \le i \le 3)$ as follows:

$$N_1 : P_1 = S_1 \oplus S_2$$
$$N_2 : P_2 = S_1 \oplus S_3$$
$$N_3 : P_3 = S_2 \oplus S_3 \oplus M$$

These values P_1, P_2 and P_3 are published and all nodes calculate the sum modulo two of these bits which results in the message bit M:

$$\begin{aligned} N_1, N_2, N_3 : \quad & P_1 \oplus P_2 \oplus P_3 \\ & = S_1 \oplus S_2 \oplus S_1 \oplus S_3 \oplus S_2 \oplus S_3 \oplus M \\ & = M \end{aligned}$$

As an extension of this protocol Chaum presents *traps* with the purpose of finding nodes which disrupt the system by preventing others from sending messages. To build a trap one node chooses a random message and a bit index of a round, encrypts both with a secret key and publishes the encryption. Later the trapper reserves the slot corresponding to the bit index and sends the random message. If one node disrupts this message, the trapper can prove the disruption by publishing the secret key. If the secret key bits of all honest nodes used in this round are disclosed, the disrupter can be identified. Later Waidner and Pfitzmann (1990) improves this method of laying traps.

The DC-net technique is used in CliqueNet (Sirer *et al*, 2001) and Herbivore (Goel *et al*, 2003), both protocols for anonymous communication.

2.3 Anonymous Communication in Ad hoc Networks

There exist different protocols for providing anonymity in mobile ad hoc networks. Most of them provide a method to find a communication path so that the nodes do not have to store information on the network topology. The route request *RREQ* in ANODR (Kong and Hong, 2003) is flooded over the network, where only the destination can determine that the message is destined for it. The RREQ contains beside a cryptographic trapdoor for identifying the destination also an onion that is built up by encrypting the concatenation of the identifier of the initiating node A, the source (also node A), and a nonce N_A that is used as a route pseudonym. Every node that receives such a RREQ adds its identifier, the identifier of the node from whom it receives the RREQ, and another nonce to the onion and encrypts this onion with its own public key. The RREQ is bounced by the destination node and is sent back the path it reached the destination. On the way back to the source the onion in the route reply *RREP* can subsequently be decrypted by all the nodes on the path. After having established a path to the destination, the source can send a data message that contains its route pseudonym. Each node that receives this message can verify if it is on the route by comparing the route pseudonym RP_i of the message and the stored route pseudonyms. If it is on the route it exchanges the route pseudonym with the outgoing pseudonym RP_{i+1} and resends it. This procedure is repeated until the message reaches the destination node. As an improvement of this protocol a technique is proposed that relies on the less cost expensive symmetric encryption and decryption.

Due to the fact that the RREQ is flooded over the network an overhead is generated, especially because every node that receives such a RREQ has to check if it is the destination of this request by opening the trapdoor. Depending on the used trapdoor it might be possible that every node has to decrypt the trapdoor with n–1 different keys, where n is the number of nodes of the network, because it does not know the originator of a message. Because the messages are sent in plain text during the communication a strong passive attacker can easily track messages from the source to the destination. Figure 1 shows that an attacker only needs to detect the sending of a message at the first and the last hop of the communication path. Because the message body does not change the attacker can easily match the source and the destination of the message. If an attacker only receives one RREQ it can estimate the hop distance to the source by examining the length of the onion: the onion grows with the hop distance to the source.

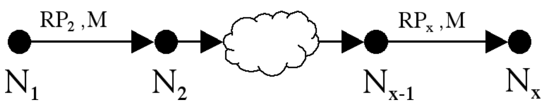


Figure 1: ANODR communication

MASK (Anonymous On-Demand Routing in Mobile Ad Hoc Networks) (Zhang *et al*, 2006) is designed for the usage in military missions. In this solution a trusted authority is needed to configure all participating nodes in advance. Every node receives a set of dynamic pseudonyms that are regularly changed during usage. Similar to ANODR the messages do not change while being sent from hop to hop, so that a tracking of messages is possible.

In ARM (Anonymous routing protocol for mobile ad hoc networks) (Seys and Preneel, 2006) it is required that every pair of nodes of the network shares a secret key and a secret pseudonym. The pseudonyms are used during the RREQ to identify the target node. After having used a pseudonym once it has to be exchanged by a new pseudonym. In this protocol the source of a RREQ has to create a pair of a public and private key, where the public key is used to encrypt the channel identifier at every hop on the path before resending the route request. The private key is included in the RREQ encrypted by the shared secret key. If the length of the path from the source to the destination is *n* then the destination node has to perform *n*–1 decryptions with its received private key to obtain all channel identifiers. To avoid that messages are tracked while traversing the network the messages are decrypted and encrypted at every hop of the path so that they change their appearance.

Another anonymous on demand routing protocol for ad hoc networks is ODAR (Sy *et al*, 2006), that uses Bloom filters (Bloom, 1970). The approach of S. Jiang (Jiang *et al*, 2001) can be used for hiding the source and the destination of a message by choosing a subset of fixed mixes, which are distributed in the ad hoc network.

3. PREREQUISITES

In this section we define the terms anonymity and unlinkability which we will use in the rest of the paper. Furthermore we introduce the network model and adversary model which is used for the evaluation of the protocol.

3.1 Anonymity

There exist definitions for *anonymity* in the literature. We use in this paper the definition of Pfitzmann and Köhntopp (Pfitzmann and Köhntopp, 2000):

> Anonymity of a subject means that the subject is not identifiable within a set of subjects, the *anonymity set*.

For network communication we can build the more precise terms *sender anonymity* with the *sender anonymity set* and *receiver anonymity* with the *receiver anonymity set*.

Anonymity is usually related to an action. In the case of sender anonymity the anonymity is related to the action of sending a message. All users that might have performed this action belong to the corresponding anonymity set. In relation to sender anonymity these are the nodes that might have sent the message. They belong to the sender anonymity set.

To be a member of an anonymity set is not the only information that an attacker can have. Some members of the anonymity set might be more likely to be the originator of an action. For network communication for example the attacker can perform traffic analysis attacks which induce different probabilities for the members of the sender anonymity set to be the sender of a message. This fact leads to the definition of the *degree of anonymity*. This value is not only based on the size of the anonymity set but also on the likelihoods for the members of the anonymity set being the originator of an action. To measure anonymity we use the information theoretic approach that was independently introduced by Serjantov and Danezis (2002) and Diaz *et al* (2002). They define the degree of anonymity as entropies, whose values depend on the likelihoods of the members of the anonymity set for being the originator of an action. We can calculate the degree of anonymity for a system S with the anonymity set $A = \{a_0, \ldots, a_n\}$ by:

$$H(S) = -\sum_{a_i \in A} P(a_i) \log_2 P(a_i).$$

The maximum of this value is reached if all members of the anonymity set are equally likely to be the originator of an action. In this case the degree of anonymity is $\log_2 n$. On the other hand the minimal degree of anonymity is reached if the likelihood for one member of the anonymity set is 1. The resulting degree of anonymity is 0. By normalizing the degree of anonymity by the maximum degree of unlinkability the values only depend on how evenly the likelihoods are distributed and not on the size of the anonymity set. The resulting values are in the interval of [0,1]. This can be useful to evaluate a protocol independently of the topology.

3.2 Unlinkability

We also use the definition of Pfitzmann and Köhntopp (2000) for unlinkability:

> Unlinkability of two or more items of interest (IOI, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not.

This means in the example of network communication that an attacker can detect sender nodes and receiver nodes, but can not detect which sender nodes communicates with which receiver node. The *unlinkability set* consists of all possible relations, i.e. all combinations of sender and receiver nodes. Also for unlinkability there exist pairs of subjects that are related with higher probability than other pairs. Therefore we introduce a measure for the *degree of unlinkability* which is a simplified version of the information theoretic approaches of Steinbrecher and Köpsell (2003) and Franz *et al* (2007).

In the following we concentrate on the term unlinkability in the context of network communication. We define the set of possible sender nodes as $S = \{s_1, \ldots, s_n\}$ and the set of possible receiver nodes as $R = \{r_1, \ldots, r_m\}$. With the notation $l_{i,j}$ we denote the communication link where s_i sent a message to r_j. The set of all possible communication links is $L = \{l_{1,1}, \ldots, l_{n,m}\}$. We define the degree of unlinkability as the entropy of all possible communication links:

$$H(L) = -\sum_{i=1}^{n} \sum_{j=1}^{m} P(l_{i,j}) log_2 P(l_{i,j}).$$

The degree of unlinkability reaches its minimum 0 if the probability of one communication link $l_{i,j}$ is 1 and the probability of all other communication links is 0. In other words the degree of unlinkability is 0 if the attacker can prove that s_i sent a message to r_j. The maximum degree of unlinkability is reached if all communication links are equally likely. It follows a degree of unlinkability of $\log_2 (n \bullet m)$ because there are $n \bullet m$ communication links.

With this definition of the degree of unlinkability we are able to measure the amount of unlinkability that is lost when an attacker makes a specific observation. Just as for the degree of anonymity we can normalize this value by the maximum degree of unlinkability in order to get a value that is independent from the size of the unlinkability set and only depends on how similar the probabilities of the links are. This results in the *unlinkability coefficient*, which is a value in the interval [0,1].

3.3 Network Model

In the following, wireless links are used for communication. These links are assumed to be symmetric, which means that if a node *A* is in transmission range of another node *B*, also node *B* is in transmission range of node *A*. Every node in transmission range of a node *A* can listen to its communication. The network might be mobile, so that nodes can enter and leave the network anytime. Nodes can join the network without configuration.

All nodes in the network have a pair of a public and private key that is used to exchange secret keys for the communication. For routing a proactive routing protocol is used. Therefore every node manages a rather up-to-date routing table where the complete communication path is stored.

3.4 Adversary Model

It is assumed that an attacker node can passively eavesdrop the communication of every node in its sending range. An attacker can control several attacker nodes that do not belong to the unlinkability and anonymity set. In the worst case the communication of the whole network can be captured. Additionally it is possible that an attacker compromises nodes that belong to the unlinkability and anonymity set, while it is assumed that not all nodes are compromised. We assume as in Shannon (1948) that the adversary knows the system being used. The adversary does not have unbounded computational power, otherwise no reasonable cryptographic solutions are expected to work.

4. ACIMN PROTOCOL

In this section we introduce the Acimn protocol which is designed for multi hop wireless networks. It is based on a combination of layered encryption and the DC net protocol.

We assume for the functionality of the protocol that the nodes know the network topology in advance. How this can be achieved is not the focus of this work. One way could be to use a modified version of a proactive routing protocol, such as the Global State Routing protocol (Chen and Gerla, 1998). This is necessary to reveal no information on the destination to the attacker while finding a path to the destination node.

For the cryptographic functions used during the execution of the protocol every node in the network has a pair of public and private keys. These keys are used for the asymmetric decryption and encryption. Additionally, every node maintains a list of combinations of node identifiers and secret keys. Associated to this secret key every node stores a counter, the counter encrypted with the corresponding secret key, and the identifier of the next node on the path. The encryption of the counter is only used to reduce the computational overhead during the communication. More details on these values and how they are exchanged will be explained later in this section.

4.1 One Hop Communication

Because of the wireless communication, every node in sending range of another node can listen to the communication. To avoid that a node is able to track messages sent through the network, the communication has to be covert. In this protocol, the communication is covert by using the DC-net approach for the one hop communication. For this reason we divide the network in groups of three nodes where every node in such a group is in sending range of the other nodes in its group. Nodes can be a member of several groups. For example in the network depicted in Figure 2 the node N_4 is a member of the groups G_2, G_3, G_4 and G_6. Only if there are not more than two nodes in sending range of each other there can be a group of two nodes (in the example network group G_8 only consists of the nodes N_8 and N_9).

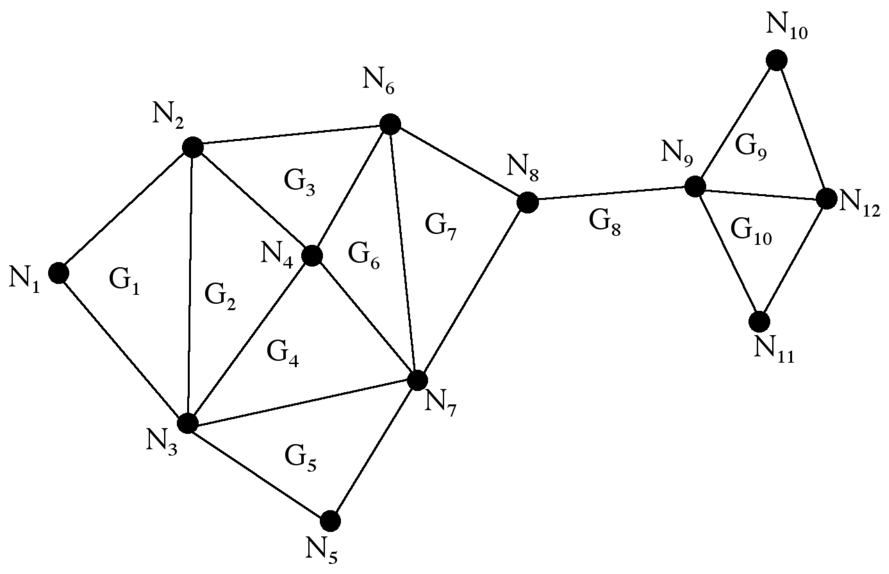


Figure 2: Example network

Pairs of nodes of these groups share private keys, which can only be used once. To minimize the traffic for exchanging keys, a pseudo-random number generator is used and the nodes exchange only the seed for the generation of the same stream of bits. The one hop communication itself follows the DC net protocol described in Section 2.2. From now on we assume that for every communication the presented technique is used.

4.2 Key Exchange for Multi Hop Communication

Our protocol for unlinkable communication mainly bases on layered encryption. Therefore, if a node wants to communicate with another node in the network, it chooses one path of the routing table and exchanges private keys with all nodes on the path with only one key exchange message.

In the first step, a secret key *SK* is generated for every node on the route. In conjunction with a secret key, the sending node stores a counter with initial value zero. This counter will be used as an identifier for the nodes on the path allowing determining which secret key to use for encrypting the received message. After creating these keys, the sending node generates a message M_n by encrypting the secret key SK_n with the node identifier ID_n of the last node N_n on the path. Subsequently, the sending node encrypts the concatenation of the node identifier ID_i and the secret key SK_i of the node N_i with the corresponding public key PK_i. The concatenation of the node identifier ID_{i+1} of the successor node and the message M_{i+1} is encrypted with the secret key of node N_i. The concatenation of both encryptions builds the message M_i. This process is repeated from the destination node N_n to the first node on the path N_1 and results in message M_1.

$$\begin{aligned} M_n &= [ID_n, SK_n]_{PK_n} \\ M_{n-1} &= [ID_{n-1}, SK_{n-1}]_{PK_{n-1}} \\ &\quad [ID_n, [ID_n, SK_n]_{PK_n}]_{SK_{n-1}} \\ M_1 &= [ID_1, SK_1]_{PK_1} [ID_2, M_2]_{SK_1} \end{aligned}$$

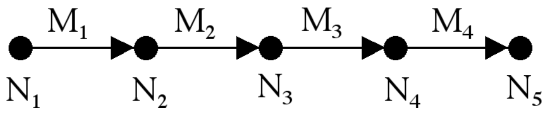


Figure 3: Key exchange

Node N_0 sends the message M_1 to the first node on the path. After decrypting the node identifier and the secret key with its private key, the node can determine if the message is destined for it by comparing the decrypted node identifier with its own node identifier. If the node identifier is correct, it can decrypt the node identifier of the next node N_2 on the path and the message M_2 with the secret key. It stores the secret key SK_1 combined with the identifier of the next hop N_2, a message counter MC_1 that is initialized to zero, and MC_1 encrypted with SK_1. By decrypting and resending the message from node to node on the path, every node obtains its key for the symmetric encryption (see Figure 3).

To hide the destination of the communication we extend the path so that all communication paths have a fixed length. If node N_0 wants to send a message to N_t it builds a path from N_0 over N_t to N_{t+e} so that $t + e$ is the wanted path length. As a result an attacker does not know which node on the path is the destination of the communication. Section 4.5 gives more details on how to select the communication paths so that the attacker receives no information on the communication links.

We use fixed packet lengths during the key exchange phase. The reason for that is that with a fixed packet size the attacker gets no information on the distance of the sender node N_0 to the end node N_{t+e} while observing the message length. A special hint in the key exchange message signals to the end node N_{t+e} that the message does not need to be resent. Because of the fixed path length the initial key exchange messages always have the same size. To keep the message size constant while it is sent over the path, we add dummy bits to the end of the message after taking off one layer of a message.

4.3 Communication Protocol

To send data using the Acimn protocol the path to the destination node has to be known and the secret keys have to be exchanged with all nodes on the path (see Section 4.2). If node N_0 wants to send the data D to the destination node N_t over path P it builds a communication message as follows: For all $0 < i \le t + e$ the message M_i is built by the concatenation of the encrypted message counter MC_i of node N_i , the encrypted message counters P_i of the following nodes on the path, and the data D_i destined for the destination node encrypted successively by all the secret keys of nodes following on the path.

$$M_i := \ [MC_i]_{SK_i}, P_i, D_i.$$

The encrypted message counters P_i are built by successively encrypting them as shown below:

$$\begin{aligned} P_{t+e} &:= \quad [-1]_{SK_{t+e}} \\ P_i &:= \ [MC_{i+1}, P_{i+1}]_{SK_i}, \end{aligned}$$

where $1 \le i \le t + e - 1$. The data is encrypted layer by layer using the exchanged secret keys:

$$\begin{aligned} D_t &:= \ [F_t, D]_{SK_t} \\ D_i &:= \ [D_{i+1}]_{SK_i}, \end{aligned}$$

where F_t is a flag to identify that the message reached the target node and $1 \le i \le t - 1$.

After creating the message, the first node sends the message M_1 into the group, which contains the node with the node identifier of N_1, representing the first node on the path, using the one hop communication protocol described in Section 4.1. Every node N_i on the path, which receives a message sent in a DC-net group, checks if one of the encrypted counters matches the one sent in the message M_i. If this is the case, the node overwrites its counter with the encryption of the incremented counter. It uses the secret key that corresponds to the encrypted counter for decrypting the encrypted message counters P_i and the data D_i . The resulting new values $[MC_{i+1}]_{SK_{i+1}}$, P_{i+1}, and D_{i+1} are used to build the message M_{i+1} that is forwarded on the path. Node N_i sends the newly built message M_{i+1} to node N_{i+1}. This process is repeated until the message M_{t+e} reaches the destination node N_{t+e}. N_t is able to detect that it is the destination of the data by checking the flag F_t and N_{t+e} can detect that it is the destination node by decrypting the message counter P_{t+e}.

In almost the same manner as in the key exchange protocol (Section 4.2) a fixed message size can be guaranteed. This can be done first by fixing the amount of data sent in each message. Second, we keep the same message size for all P_i, for $0 < i \leq t + e$, while adding random bits at the end of P_{i+1} in every communication step. When using a block cipher these random bits have no effect on the other encrypted message counters. The protocol for multi hop communication offers unlinkability even if the message size is not fixed. It is only needed to increase the degree of unlinkability and receiver anonymity as we will see later in Section 6.

To make the communication protocol robust to message loss, a series of message counters is stored. In this series the encryptions of the message counters $[MC],[MC + 1],\ldots,[MC + s]$ are stored. The parameter s has to be chosen in such a way that nearly no message is wrongly discarded and the overhead because of the additional checks is minimized. If a message with message counter $[MC + i]$ arrives the message counter window is moved forward to the counters $[MC + i + 1],\ldots,[MC + i + s]$.

4.4 Response Channel

The basic Acimn protocol only provides a communication channel in one direction. In this section we present a procedure for the backward communication.

If node N_0 wants to receive a message from node N_t while keeping the unlinkability it chooses a path starting at node N_0 that goes through node N_t and ends again in N_0. An easy way is to choose the same path for the backward communication as for the forward communication. In this case the last node on the path N_{t+e} bounces the message back on the same path. For the backward channel node N_0 initiates a key exchange with the nodes on the path as described in Section 4.2. After this key exchange node N_0 can send messages to N_t (see Section 4.3).

To receive messages from node N_t, node N_0 sends a message with random bits in the data field to N_t . N_t exchanges the random bits with the encrypted response and forwards the message on the path. All other nodes perform the same action as before: They check the encrypted message counter, decrypt the path section and the data section with the stored private key.

When N_0 receives the response message it just needs to decrypt the layered encrypted message with the private keys of the nodes on the path from N_t over N_{t+e} to N_0. As a result only the nodes N_0 and N_t perform different computations than during the normal communication protocol. An attacker is not able to detect where the dummy bits are exchanged by the response because of the layered encryption.

4.5 Path Selection

We assume that the adversary knows the algorithms used by the nodes in the network (see Section 3.4). If we assume that the adversary is able to observe the complete communication path we have to ensure that the paths have to be chosen in such a way that the observation of the communication path offers no additional information that helps to identify the communication partners.

Extending the communication path by some dummy nodes as mentioned in Section 4.2 helps to prevent the attacker from detecting the destination of a communication. If we use a constant path length for all communications the attacker receives no information by observing the path length of a communication.

The choice of the nodes on the path can also offer information on the communication partners. If we only concentrate on the multi hop communication protocol the attacker is able to observe the initiator of the communication N_0. The only way to keep the unlinkability and the receiver anonymity is by hiding the destination node in the set of all nodes on the communication path. Because the attacker knows the algorithm which is used to choose the nodes on the path the algorithm has to choose a communication path so that all possible communication links (N_0, N_i), for $1 \leq i \leq t + e$, are equally likely.

An easy algorithm that fulfills this requirement is choosing the nodes on the communication link randomly. Because the likelihood that a node is on the communication path is equal also all possible combinations of (N_0, N_i) for $1 \leq i \leq t + e$ have the same likelihood. Unfortunately this way of choosing a path is not practical in multi hop wireless networks for the following reason: Because not all nodes are connected directly, other nodes are used to connect the nodes on the communication path. These nodes do not belong to the receiver anonymity and unlinkability set, because the attacker might detect that these nodes are not chosen randomly. More nodes on the communication path induce that the chance of a failure and the overhead increases. So by decreasing the number of nodes on the communication path that do not contribute to the receiver anonymity and unlinkability set we can increase the robustness and decrease the overhead.

We propose to use the cheapest path from node N_0 to N_t (in our case this is the shortest path) and extend this path, so that for each following node N_t , $t < i \leq t + e$, the path from N_0 to N_i is also optimal. If it is not possible to find $t + e$ nodes that fulfill this requirement we reduce the path to these nodes, so that the requirement is still fulfilled. So it is possible that the communication path consists of less that $t + e$ nodes. But if we extend the path by nodes N_i where the path from N_0 to N_i is not optimal, a strong attacker is able to detect these nodes. As result these nodes do not necessarily contribute to the receiver anonymity and unlinkability set but increase the overhead and chance of failure. If N_k is the last node on the path we have a receiver anonymity and unlinkability set of size k which results in a degree of receiver anonymity and degree of unlinkability of $\log_2 k$.

If we assume that the network topology is known to the nodes, the sender node N_0 can easily build the cheapest path to the target node N_t by using the Dijkstra algorithm (Dijkstra, 1959). The Dijkstra algorithm initializes the cost of all nodes to ∞ and the cost for N_0 to 0. All nodes are declared as active. In every step it takes the active node N_i with the cheapest cost c_i for the path P_i from N_0 to N_i , and checks for all outgoing edges $e(N_i, N_j)$ if $c_i + cost(e(N_i, N_j)) < c_j$. If this is the case it updates the cost c_j to $c_i + cost(e(N_i, N_j))$ and P_j to $P_i \cup N_i$. We additionally store for each node N_i the length l_i of the cheapest path in order to know if the desired size for the unlinkability set is reached. After having checked all outgoing edges of N_i we declare N_i as passive. The algorithm is repeated until the target node N_i is the cheapest active node which means that P_t is the cheapest path from N_0 to N_t .

To extend the path by nodes N_i, $t < i$, so that the resulting path from N_0 to N_i is optimal we continue the Dijkstra algorithm in such a way, that we mark all active nodes, whose cheapest path goes over N_t . The algorithm ends if a marked active node N_i is reached, whose path length l_i is equal to the desired size of the unlinkability set or if there is no further marked active node. In this case we take the marked passive node N_k, who has the maximum path length l_k.

5. PERFORMANCE

We evaluate the performance of the Acimn protocol with respect to message and computational overhead. We analyze the message overhead theoretically in Section 5.1. In the following Section 5.2 we measure the computational overhead with the use of a prototype implementation of Acimn.

5.1 Message Overhead

Acimn produces message overhead by using the layered encryption technique, by enlarging the path over the target node, and by using DC net for the one hop communication. Subsequently we show how many messages each of these techniques generates.

The layered encryption generates message overhead only during the key exchange phase. For the key exchange the sender node only needs to send one key exchange message over the complete path in order to exchange secret keys that are used during the communication. The size of the key exchange message depends on the algorithms used for the symmetric and asymmetric encryption and also on the length of the communication path. In our prototype we use RSA for asymmetric encryption with public key length of bytes and private key length of bytes. This results in a block length of bytes if the messages are encrypted with the public key. For a path of length t the protocol generates a key exchange message of length $t \cdot 128$ bytes. If we extend a path of t nodes by e additional nodes, the protocol generates additional e messages for the key exchange and the length of the key exchange increases to $(t+e) \cdot 128$. During the communication the protocol produces only e additional messages that are used to guarantee the fixed path length.

For sending a message using DC net at every hop all nodes in the corresponding DC net group have to send one message. Because we fixed the size of the DC net groups to 3 this generates a message overhead of 2 at every one hop communication step. We do not take into account the messages that are used to exchange once the seeds for the pseudo random generators and the messages that are used for the round reservation during the execution of the DC net protocol.

If a communication path is used to send i messages then the total number of additional messages is $3 \cdot (t+e) + (2 \cdot t+3 \cdot e) \cdot i$.

5.2 Computational Overhead

We measure the computational overhead with the help of a prototype implementation of the protocol. The prototype is implemented in Java. We use AES with a key length of 16 bytes for the symmetric encryption and RSA with a private key length of 635 bytes and public key length of 162 bytes. The reference hardware is a Dell Latitude D610 Laptop with a 2GHz Intel Pentium M processor, 1GByte main memory and Windows XP SP2. To evaluate the computational overhead we measure the duration for generating and processing key exchange and communication messages. We average the results over 10000 runs.

In the first place we measure the performance of a key exchange. Figure 4 shows how long it takes to generate and to process a key exchange message. The duration to generate a key exchange message strongly depends on the length of the path, because the sending node has to perform one asymmetric encryption and one symmetric encryption for each node on the path. The growth of the time is nearly linear. The reason for that is that if we add one node to the path the sending node has to perform exactly one more asymmetric encryption and it has to encrypt the body of the message symmetrically. In contrast to the generation of a key exchange message the processing does not increase noticeably when the communication path increases. All nodes only encrypt a small constant part of the message asymmetrically. Only the body of a key exchange message grows linearly if the communication paths become longer. But for the body of the key exchange message

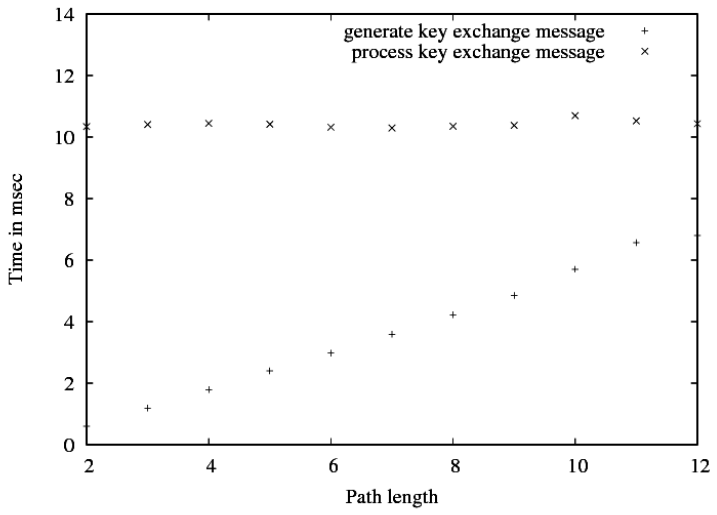


Figure 4: Performance of key exchange

the nodes use efficient symmetric encryption with the exchanged keys. Therefore the duration to process a key exchange message grows linearly with a very small parameter. In our measurements the time to process a key exchange increases from 10.406 ms at a path length of 3 up to 10.432 ms at a path length of 12.

In the same way the time to generate a communication message increases when the communication path becomes longer. For the measurements in Figure 5 we assume a communication message with 1 KByte of data. The sending node encrypts the message symmetrically once for every node on the path. The message itself only grows by 16 bytes if the communication path grows by one node so that in the end the duration of generating a communication message is nearly linear to the length of the communication path. Just as for the key exchange the processing of a

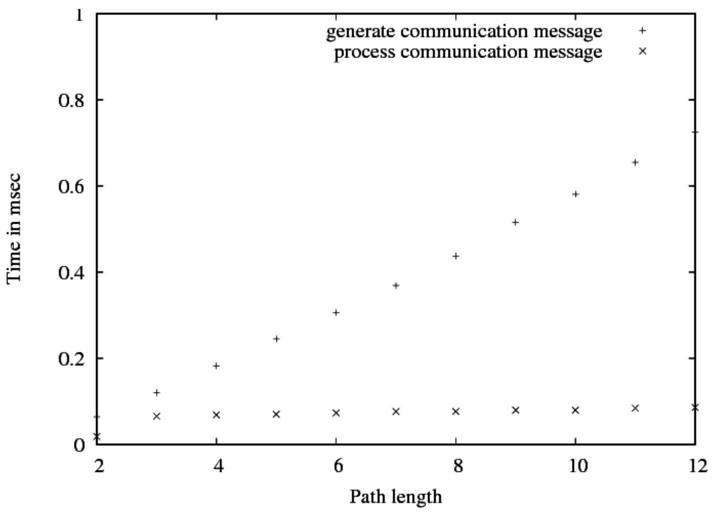


Figure 5: Performance of communication

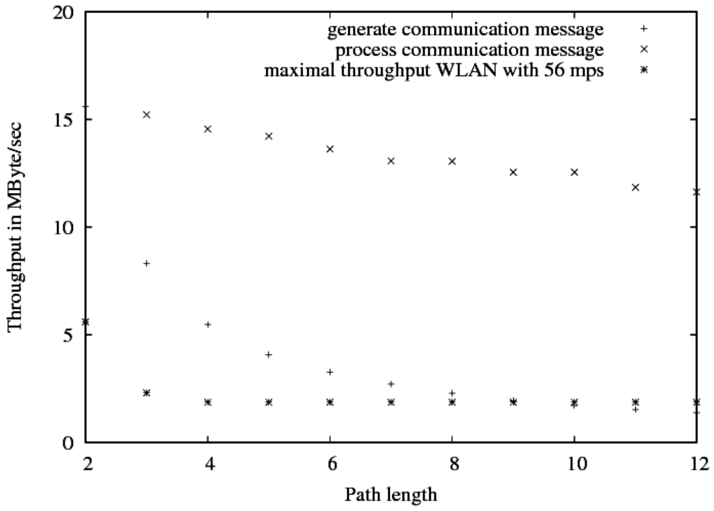


Figure 6: Throughput during communication

communication message only grows slowly. Because the communication messages grow by 16 bytes for each node on the path there is a minimal increase in the time that is needed for the decryption. In our measurements the duration grows linearly from 0.066 ms for a path of 3 nodes up to 0.086 ms for a path of 12 nodes.

Figure 6 shows the throughput that the prototype implementation reached during the measurements. The common WLAN standard with 54 mps offers a maximal real throughput of 5.6 MByte/sec for one hop communication. For multi hop communication the maximum throughput falls to one half for two hop communication and one third for more than two hop communication (Scherer and Engel, 2006). Even at a path length of 12 the protocol offers a better throughput for processing communication messages than WLAN. The throughput of generating communication messages falls under the theoretical maximum throughput of WLAN at a minimum path length of in our measurements. In summary the performance of Acimn is sufficient for using it in practice.

6. SECURITY

We split the analysis of Acimn protocol in three parts. First we show that the protocol used for the key exchange and for the encryption of the data cannot be broken if we assume that the cryptographic functions cannot be broken. Second, we prove that the Acimn protocol offers provable receiver anonymity and unlinkability and show that the one hop communication protocol helps to increase the degree of anonymity and unlinkability exponentially in the best case. Third, we show influences of passive and active attacks on the protocol.

6.1 Algorithms in Acimn

In our prototype we use AES with a key length of 128 bit for symmetric encryption. AES is rated as secure and is recommended by several institutes (i.e. BSI in Germany). For asymmetric encryption we use RSA-1024 in our prototype which has not yet been factored. During the DC net communication we use a one time pad which is generated by a cryptographically secure pseudo random generator. We proved with the Scyther security protocol checker (Cremers, 2006) that under the assumption that the cryptographic algorithms are secure an attacker is not able to gain the secret keys that are exchange during the communication. Details on the Scyther code can be found in Fusenig *et al* (2008a).

6.2 Anonymity and Unlinkability

The protocol consists of two basic strategies: First the multi hop communication protocol with the layered encryption and extended paths; second the one hop communication protocol that bases on DC net.

For the multi hop protocol we can prove that it offers perfect unlinkability and receiver anonymity by showing that all possible observations an attacker can make offers no information on the communication target N_t of a sender node N_0. A strong passive attacker might be able to detect all nodes on the path, for example by using packet counting attacks. But if the nodes on the path are chosen in the same way, e.g. choosing them randomly, the attacker gains no information on the target node. As result for a path of length p there are p different nodes which all have the probability of $\frac{1}{p}$ for being the target node N_t. The resulting degree of receiver anonymity is in this case $\log_2 p$ which is optimal for an anonymity set of size p. Because all messages are forwarded over the complete path there are sending and receiving events at every node on the path. Therefore observing such an event offers no information to the attacker that helps to detect node N_t. Size and content of packets also offer no further information for the attacker: The size of the packets is constant and the content changes at every hop due to layered encryption. Layered encryption is important for hiding the step when a random message is replaced by a reply (see Section 4.4). Since all nodes perform the same computations also the timing of packets offers no additional information to the attacker. More information on the proof can be found in Fusenig *et al* (2008b). As a result if we assume a strong passive attacker as stated in Section 3.4 the minimum degree of unlinkability and receiver anonymity is $\log_2 \frac{1}{p}$ where p is the length of the communication path.

The one hop communication only increases the degree of receiver anonymity from $\log_2 \frac{1}{p}$ to $\log_2 \frac{1}{p+1}$ where p is the path length under the assumption of a strong passive attacker who is able to detect all communication. In this case we have a degree of sender anonymity of 1, because in the first sending group there are 2 of the 3 nodes that might have sent the message. The third node forwards the message that can be detected by the attacker. The degree of unlinkability increases from $\log_2 \frac{1}{p}$ to $\log_2 \frac{1}{2 \cdot (p+1)}$ because there are two possible sender nodes and one more possible receiver node. If we assume a weaker attacker, who has only partial knowledge of the network traffic, the size of the unlinkability set and anonymity set might increase exponentially. This is because at every hop there are two nodes on the path which receives the message and might be the successor of the path.

6.3 Attacks

In the following we give a list of known attacks on anonymity protocols and show the impact of these attacks on Acimn. An overview on traffic analysis attacks can be found in Raymond (2001).

Denial of Service Attack

The aim of a denial of service attack is to destroy the availability of a target device or target service. As seen in Section 5 only the performance during the key exchange is critical. To overcome this problem a way to detect such an attack has to be found. This is part of future work. During the communication, the algorithms used in the protocol offers better throughput than the wireless medium. Therefore the most effective way for a denial of service attack is by attacking the wireless communication, which is not part of this work.

Replay Attack
In this attack the attacker resends packets in order to get access to a session or a service. In anonymity protocols replaying of messages can be used to gain better results for statistical attacks, such as the packet counting attack later in this section. In Acimn the attacker cannot replay messages because of the use of encrypted message counters. Only a node that possesses the corresponding secret key is able to build an incremented version of an encrypted counter.

Message Coding Attack
In a message coding attack the attacker analyzes the input and output traffic of a node and tries to map packets by means of their coding and appearance. Because Acimn uses fixed packet length and layered encryption at every hop an attacker gains no information by performing this attack.

Collusion Attack
For the evaluation of unlinkability and receiver anonymity we assumed a strong passive attacker who is able to observe every communication. The protocol even offers unlinkability and receiver anonymity if some of the nodes on the path collude and every communication is known to the attacker. As long as at least one node on the path except the sender and receiver node does not belong to the colluding attackers the protocol offers unlinkability and receiver anonymity.

Packet Volume Attack
Because Acimn uses fixed packet length, the attacker gains no information from the packet length of an observed packet.

Packet Counting Attack
Packet counting attacks can be successful for several anonymity protocols such as mixes and Tor. In a packet counting attack the attacker counts the number of incoming and outgoing packets at a node in the anonymity system. In cases where the number of incoming and outgoing packets matches, there has been a communication link with high probability. In Acimn this attack reveals in the best case all nodes on the communication path. But it is impossible to detect the target of the communication.

Message Delaying Attack
By delaying messages, the attacker tries to get the system into a state where it is easier to reveal the communication links. Delaying packets has no influence on the execution of Acimn. In the worst case the result of a delaying attack is a denial of service.

Flooding Attack
By sending many packets over the target node the attacker tries to separate packets of the victim that pass the target node. The worst case while using Acimn is when the attacker can reveal all nodes on the path by successively applying this flooding attack on all nodes of the path. But the attacker cannot identify the receiver node out of the nodes on the path.

Intersection Attack
For this attack the attacker observes the victim node for a longer time. The attacker intersects the possible communication partners of every communication and downsizes the anonymity and

unlinkability set. Also this attack only is able to detect the nodes on the path, but not the receiver node of a communication. This technique is the basic function of the disclosure attack (Agrawal and Kesdogan, 2003).

Timing/Latency Attack

The attacker tries to find communication partners with the help of precomputed tables of latencies. If the victim sends a message to another node the attacker maps the measured latency during the attack with the precomputed latencies. Due to the fact that the paths are extended in Acimn the attacker can detect in the worst case only the last node on the path but not the recipient.

Clogging Attack

For clogging attacks the attacker observes the traffic at a node *A* and the predecessor node *B*. He chooses a random node *C* of the network and floods this node with packets. If the traffic on the link from *B* to *A* collapses, node *C* belongs to the path with high probability. By performing this attack successively the attacker is able to detect the originator of the communication to node *A*. Because of the extended paths this attack offers in the worst case only all nodes on the path.

The result of this analysis is that none of the known attacks is able to reduce the minimum degree of receiver anonymity of $\log_2 \frac{1}{p+1}$ and the minimum degree of unlinkability which is $\log_2 \frac{1}{2 \cdot (p+1)}$ for a path length of p. Only if nodes on the path are under control of the attacker is he able to drop the degree of unlinkability and anonymity under these bounds. The attacks were analyzed under the assumption that the attacker is able to perform them in the best way. In reality the offered degree of anonymity and unlinkability will be higher even under attacks.

7. CONCLUSION

We introduced a definition of anonymity and unlinkability and used these definitions to analyze the Acimn protocol. In the worst case Acimn guarantees a minimum degree of unlinkability of $\log_2 \frac{1}{2 \cdot (p+1)}$ and receiver anonymity of $\log_2 \frac{1}{p+1}$ where p is the length of the communication path. This means that the unlinkability set and the receiver anonymity set is linear to the length of the path. In the best case, when the attacker is not able to observe all network communication, these sets increase exponentially.

We measured the performance with the help of a prototype implementation. Acimn introduces moderate computational overhead. In our measurements the computational overhead offers better throughput than multi hop sending in wireless networks for path length up to 9 nodes. The message overhead that is generated during the execution of Acimn is linear to the size of the minimum receiver anonymity set and unlinkability set.

REFERENCES

AGRAWAL, D. and KESDOGAN, D. (2003): Measuring anonymity: The disclosure attack. *IEEE Security & Privacy* 1(6): 27–34.

BAUER, K., MCCOY, D., GRUNWALD, D., KOHNO, T. and SICKER, D. (2007): Low-resource routing attacks against tor. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*. ACM, New York, NY, USA, 11–20.

BERTHOLD, O., FEDERRATH, H. and KÖPSELL, S. (2000). Web MIXes: A system for anonymous and unobservable Internet access. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, 115–129.

BLOOM, B. H. (1970): Space/time tradeoffs in hash coding with allowable errors. *Communications of the ACM* 13(7): 422–426.

BOS, J. and DEN BOER, B. (1990): Detection of disrupters in the dc protocol. In *EUROCRYPT '89: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York, Inc., New York, NY, USA, 320–327.

CHAUM, D. (1981): Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 4*: 2 (February).

CHAUM, D. (1988): The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology* 1: 65–75.

CHEN, T.-W. and GERLA, M. (1998): Global state routing: A new routing scheme for ad-hoc wireless networks. In *IEEE International Communications Conference, ICC '98*, June 1998, Atlanta, GA, USA. IEEE, 171–175.

CREMERS, C. (2006): Ph.D. dissertation. Ph.D. thesis, Eindhoven University of Technology.

DANEZIS, G., DINGLEDINE, R. and MATHEWSON, N. (2003): Mixminion: Design of a Type III anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*.

DÍAZ, C., SEYS, S., CLAESSENS, J. and PRENEEL, B. (2002): Towards measuring anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, DINGLEDINE, R. and SYVERSON, P. Eds. Springer-Verlag, LNCS 2482.

DIJKSTRA, E.W. (1959): A note on two problems in connexion with graphs. *Numerische Mathematik 1*: 269–271.

DINGLEDINE, R., MATHEWSON, N. and SYVERSON, P. (2004):. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, 21–21.

EU COMMISSION (2006): Directive 2006/24/ec of the european parliament and of the council of 15 march 2006. *Office Journal of the European Union L* 105/54.

FRANZ, M., MEYER, B. and PASHALIDIS, A. (2007): Attacking unlinkability: The importance of context. In *Privacy Enhancing Technologies*, BORISOV, N. and GOLLE, P., Eds. Lecture Notes in Computer Science, vol. 4776. Springer, 1–16.

FUSENIG, V., SPIEWAK, D. and ENGEL, T. (2008a): Acimn: A protocol for anonymous communication in multi hop wireless networks. In *Sixth Australasian Information Security Conference (AISC 2008)*, BRANKOVIC, L. and MILLER, M., Eds. CRPIT, ACS, Wollongong, NSW, Australia, 81: 107–114.

FUSENIG, V., STAAB, E., SORGER, U. and ENGEL, T. (2008b): Unlinkable communication. In *Proceedings of the Sixth Annual Conference on Privacy, Security and Trust (PST2008)*.

GOEL, S., ROBSON, M., POLTE, M. and SIRER, E. G. (2003): Herbivore: A scalable and efficient protocol for anonymous communication. Tech. Rep. 2003-1890, Cornell University, Ithaca, NY. February.

GOLDSCHLAG, D.M., REED, M.G. and SYVERSON, P.F. (1996): Hiding routing information. In *Proceedings of Information Hiding: First International Workshop*, ANDERSON, R., Ed. Springer-Verlag, LNCS 1174: 137–150.

JIANG, S., VAIDYA, N.H. and ZHAO, W. (2001): A dynamic mix method for wireless ad hoc networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force)*. IEEE, 2: 873– 877.

KONG, J. and HONG, X. (2003): Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM Press, New York, NY, USA, 291–302.

PFITZMANN, A. and KÖHNTOPP, M. (2000): Anonymity, unobservability, and pseudonymity – a proposal for terminology. In *Workshop on Design Issues in Anonymity and Unobservability*, FEDERRATH, H., Ed. Lecture Notes in Computer Science, vol. 2009. Springer, 1–9.

RAYMOND, J.-F. (2001): Traffic analysis: protocols, attacks, design issues, and open problems. In *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, FEDERRATH, H., Ed. LNCS, vol. 2009. Springer-Verlag New York, Inc., New York, NY, USA, 10–29.

REITER, M. and RUBIN, A. (1998): Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security 1*: 1 (June).

SCHERER, T. and ENGEL, T. (2006): Bandwidth consumption for providing fair internet access in wireless mesh networks. In *The 2006 IEEE International Workshop on Wireless Ad-hoc and Sensor Networks (IWWAN 2006)*.

SERJANTOV, A. and DANEZIS, G. (2002): Towards an information theoretic metric for anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, DINGLEDINE, R. and SYVERSON, P., Eds. Springer-Verlag, LNCS 2482.

SEYS, S. and PRENEEL, B. (2006): Arm: Anonymous routing protocol for mobile ad hoc networks. In *Proceedings of the 20th IEEE International Conference on Advanced Information Networking and Applications - Workshops (AINA 2006 Workshops)*. IEEE, Vienna, AU, 133–137.

SHANNON, C.E. (1948): Communication theory of secrecy systems. *Bell System Technical Journal* 28(4): 656–715.

SHIELDS, C. and LEVINE, B.N. (2000): A protocol for anonymous communication over the internet. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*. 33–42.

SIRER, E.G., POLTE, M. and ROBSON, M. (2001): Cliquenet: A self-organizing, scalable, peer-to-peer anonymous communication substrate. Tech. Rep. TR2001, Cornell University, Computing and Information Science. November.

STEINBRECHER, S. and KÖPSELL, S. (2003): Modelling unlinkability. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, R. Dingledine, Ed. Springer-Verlag, LNCS 2760, 32–47.

SY, D., CHEN, R. and BAO, L. (2006): Odar: On-demand anonymous routing in ad hoc networks. In *The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*.

SYVERSON, P., REED, M. and GOLDSCHLAG, D. (2000): Onion Routing access configurations. In *DARPA Information Survivability Conference and Exposition (DISCEX 2000)*. IEEE CS Press, 1: 34–40.

WAIDNER, M. and PFITZMANN, B. (1990): The dining cryptographers in the disco: unconditional sender and recipient untraceability with computationally secure serviceability. In *EUROCRYPT '89: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York, Inc., New York, NY, USA, 690.

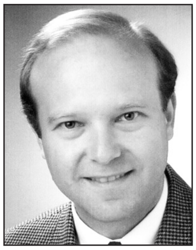
ZHANG, Y., LIU, W., LOU, W. and FANG, Y. (2006): Mask: Anonymous on-demand routing in mobile ad hoc networks. In *Transactions on Wireless Communications*. IEEE, 21: 2376–2385.

BIOGRAPHICAL NOTES

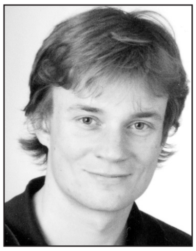
Dagmara Spiewak received her diploma degree in computer science in 2005 from the University of Trier (Germany). She wrote her diploma thesis "Securing and hardening the use of personal digital assistants" in co-operation with the European Space Agency. Since 2005 she has worked as assistant on her PhD in the field of "k-Anonymity in mobile networks" in the computer science deparment of the University of Luxembourg (Luxembourg) in the group of Professor Dr Thomas Engel.

Dagmara Spiewak

Dr Thomas Engel is professor for Computer Networks and Telecommunications at the University of Luxembourg. His team SECAN-Lab (http://wiki.uni.lu/secan-lab) deals with performance, privacy and identity handling in Next Generation Networks. As a member of the European Security Research Advisory Board (ESRAB) of the European Commission in Brussels he advised the Commission on the structure, content and implementation of the FP7 Security Research Programme. He coordinates the European Integrated Project u-2010 with 16 partners on the subject of Next Generation Networks using IPv6. Thomas Engel is a member of the Information and Communication Security Panel ICS of NATO.


Thomas Engel

Volker Fusenig received his diploma degree in computer science in 2005 at the University of Trier, Germany. Currently he is a PhD candidate in the Computer Science and Communication research unit (CSC) of the University of Luxembourg. He is working in the Mesh Sequencer project where his main research interest is in the field of computer security. The focus of his work is anonymous communication, anonymity metrics, privacy and traffic analysis attacks and defenses.


Volker Fusenig