# Acimn protocol: A protocol for Anonymous Communication In Multi hop wireless Networks

**Volker Fusenig** **Dagmara Spiewak** **Thomas Engel**

University of Luxembourg
Faculté des Sciences, de la Technologie et de la Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
Email: {Volker.Fusenig, Dagmara.Spiewak, Thomas.Engel}@uni.lu

## Abstract

While communicating wireless every node residing in communication range of the sending node is able to eavesdrop the communication. The content of the messages can be kept confidential by encryption. On the other hand the communication partners are still known. Even if no personal information like source and destination address is included in the message an attacker might reveal the communication partners with the help of traffic analysis. In this paper the Acimn protocol is introduced which enables anonymous communication in multi hop wireless networks. It bases on the combination of the dining cryptographers networks and layered encryption. Both, the message overhead and the overhead due to cryptographic algorithms, is kept small while the protocol offers protection against traffic analysis attacks.

*Keywords:* Anonymity, Privacy, Mobile Computing.

## 1 Introduction

The awareness for the protection of privacy increases. To preserve privacy, the communication partners have to be hidden to nonparticipants. In today's Internet it is possible to determine who talks to whom and also how often, even if the communication is encrypted. In recent years, methods were developed to make the communication anonymous, but they did not get in place mainly because of the poor throughput. In multi hop wireless networks it is even more difficult to keep the communication partners anonymous. Privacy protection in such scenarios will become more important with the new applications in such environment like IP telephony and car to car communication. Because of the wireless communication, all nodes in sending range can listen to the communication. By using the default routing protocols, these nodes can easily determine the sender and the receiver of every message. Since the existing tools for anonymity are build for the usage in wired internet communication, they are not applicable without modification in a wireless environment.
In this paper a protocol for anonymous communication in multi hop wireless networks is described. By the usage of this protocol the sender and the receiver of a message are hidden to all other nodes of the network. Even the nodes on the path from the originator to the destination only know their successor on the path. Every other node of the network actually does not recognize that messages are sent over the network if additionally dummy traffic is produced. Furthermore it is impossible to track messages on their way to the destination node, because all messages have the same size and change their content on every one hop of communication. Due to the fact that the messages are sent encrypted over the network, only the sender and the destination node can read the message.
The rest of the paper is organized as follows: in the section 2 the two main techniques, *Mixes* and *DC nets*, for providing anonymity are presented. The underlying models are introduced in section 3. In section 4 the operation of the protocol is presented. Section 5 shows the performance of the Acimn protocol followed by a security analysis in section 6. Finally section 7 summarizes and concludes the paper.

## 2 Related Work

There are two basic strategies to achieve anonymity in computer networks: *Mixes* and *DC-nets*. The later introduced protocol is based on the usage of both techniques. The functionality of these techniques will be presented in the next two sections. Afterwards an overview of protocols for anonymous communication in multi hop wireless networks is given.

### 2.1 Mixes

The first *mix* was designed by Chaum (Chaum 1981) to hide the communication participants in an electronic mail system. A mix collects pieces of mails sent by different participants, modifies the from address and sends the pieces in a re-sorted order to the destination or to another mix.
Based on this idea, several tools which provide anonymity were presented (Danezis et al. 2003, Reiter & Rubin 1998, Shields & Levine 2000, Berthold et al. 2000). A technical innovation was provided by the introduction of *Onion Routing* (Goldschlag et al. 1996, Syverson et al. 2000). Onion Routing generalizes the mix protocol in such a way that it can handle any communication in the Internet. *Tor*, the *Second-Generation Onion Router* (Dingledine et al. 2004), is an enhancement of the Onion Router fixing some problems of the Onion Router 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 incrementally by negotiating a symmetric key by executing a Diffi-Hellman handshake with each Onion Router in the circuit. To extend the circuit, the client $A$ sends a *relay extent* message to the last Onion Router $B$ on the circuit, specifying the address of the next Onion Router $C$ and the first part of the Diffi-Hellman handshake $g^x$ encrypted by the public key $PK_C$ of $C$. $B$

sends the message to $C$ in order to extend the circuit. $C$ responds to $B$ with the second part of the Diffi-Hellman handshake $g^y$. $B$ wraps it and sends it back to $A$. The client can now send messages at a fixed size over the built up circuit where the messages are unwrapped by the symmetric key at each Onion Router. With this protocol the sender $A$ of the messages remains anonymous, because it uses no public keys to authenticate itself. Additionally $A$ knows that it is handshaking with the right Onion Router $C$ because it encrypted the first part of the handshake with the public key of the Onion Router $C$ and only $C$ has the negotiated key. When the circuit is established, $A$ can communicate anonymously as long as at least one node in the circuit is honest.
In wireless networks there exist two basic problems that makes the usage of mixes ineffective. Because of the wireless communication every node in sending range of another node can eavesdrop the communication. To avoid that an attacker tracks messages over a cascade of mixes, each mix has to collect several messages and has to forward them in a resorted order. However each node has only a small number of other nodes in sending range so that the time for collecting enough messages for resorting would cause a high delay. Not only that high delays are not acceptable for a lot of applications, the delays make a communication impossible if the wireless network is dynamic.

## 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 how participants of a group can communicate anonymously among each other. The generalized approach works with any number of participants in a group greater than two. If the group consists only of two participants, only a nonparticipant listener is unable to distinguish between the sender and the receiver of a message. In a group with more than two participants neither a participant inside the group nor a nonparticipant can detect the sender or the receiver of a message.
To build up a DC-net group, each participant has to share a secret key with each other participant. Only one key bit is used in every round and the keys are only used once. In order to simplify matters, it is assumed that only one participant sends in every round. Possible reservation techniques can be found in (Bos & den Boer 1990, Chaum 1988). In the i-th round each participant calculates the sum modulo two of the i-th bit of all pairwise shared keys. The participant $A$, 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 participants. All other participants send the sum modulo two. After receiving the bits of all participants, each participant 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 $A$.
In the example of figure 1 the nodes $N_1$ and $N_2$ share a 1, the nodes $N_1$ and $N_3$ share a 0, and the nodes $N_2$ and $N_3$ share a 1. Assumed that node $N_1$ has reserved the current round and wants to send a 1, it builds the sum modulo two of the shared bits 1 and 0, inverts it to 1, and sends it to the other nodes. The nodes $N_2$ and $N_3$ build the sums modulo two resulting in 0 for node $N_2$ and 1 for node $N_3$ and publish them. The sum modulo two of these published bits is the bit 1 which was sent by node $N_1$.
As an extension of this protocol Chaum presents *traps* with the purpose of finding participants which disrupt the system by preventing others from sending

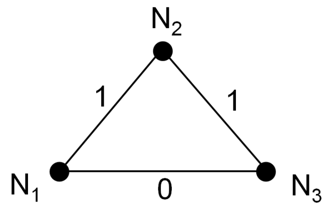


Figure 1: DC net functionality

messages. To build a trap one participant 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 participant disrupts this message, the trapper can prove the disruption by publishing the secret key. If the secret key bits of all honest participants used in this round are disclosed, the disrupter can be identified. Later Waidner (Waidner & 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 are 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 are reactive such as DSR (Johnson et al. 2004) and AODV (Perkins et al. 2003), this means that the nodes of the network do not store information on the network topology. So if a node $A$ wants to send a message to node $B$ it has to discover a route from $A$ to $B$.
The route request (*RREQ*) in ANODR (Kong & 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 be subsequently decrypted by all the nodes on the path. After having established a path to the destination the source can send a data messages 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 participants of the network, because it

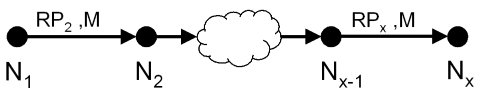


Figure 2: ANODR communication

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 2 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 destination to the source by examining the length of the onion: the onion grows with the hop distance to the source.

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 & 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 symmetric 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 Underlying models

In this section the assumptions on the models are presented.

### 3.1 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 preconfiguration.

All nodes in the network have a pair of a public and private key that need not be signed by a trusted authority. The public key will be used as identifier and the node can authenticate itself by using the private key. These keys only have to be created once when entering the network. For routing a proactive routing protocol is used. Therefore every node manages a relatively up-to-date routing table.

### 3.2 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, so that in the worst case the communication of the hole network can be captured. Additionally it is possible that an attacker compromises a node of the target network, while it is assumed that not all nodes are compromised. The adversary does not have unbounded computational power, otherwise no reasonable cryptographic solutions are expected to work.

## 4 Protocol

The following protocol is designed for wireless networks. The wireless medium creates new challenges: every participant in sending range of two nodes $A$ and $B$ can listen to their communication and each participant only has a limited number of possible direct communication partners. If a node wants to communicate with another nodes that are not in sending range it has to send the message multi hop to the destination node. The presented protocol shows a way how each communication partner stays anonymous in such a scenario. The anonymity is based on DC-nets combined with layered encryption.

### 4.1 Configuration

The wireless network is divided into groups of three nodes, where every node in such a group is in sending range of the other nodes in its group. Each node can be a member of several groups. For example in the network depicted in figure 3 the node $N_4$ is 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 $N_9$). For the protocol it is essential that a node knows the routes to the communication partners and additionally the public keys of the nodes on the route. The path finding algorithm is not focus of this paper. It can be performed like in Global State Routing (Chen & Gerla 1998) with the difference that every node stores the complete path and not only the next node on the path to the destination node. This is necessary to hide the destination node to the nodes on the path. From now on it is assumed that every node has a valid route to every communication partner.

Every node has a public and private key pair. The public key is used as the unique identifier in the network. Additionally, every sending node maintains a list of nodes with the associated symmetric keys (the key exchange will be explained later). Associated with the symmetric key they store a counter, the counter encrypted with the symmetric key, and the identifier of the next node of the channel. The counter is used to identify the symmetric key for decrypting a received message. By storing the encryption of this counter, less computational power is needed for the identification. Details of this mechanism will be explained in the subsequent sections.

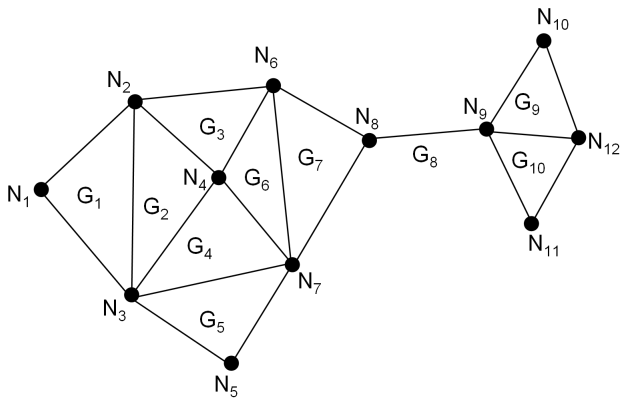


Figure 3: Example network

## 4.2 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 participant 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, the wireless network is divided into groups of three nodes as described in section 4.1. Nodes of these groups share a private key, 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 protocol described in section 2.2. By adding dummy traffic the communication can be completely hidden to nonparticipating nodes. From now on it is assumed that for every communication the presented technique is used.

## 4.3 Key Exchange for multi hop communication

If a node wants to communicate anonymously with another node in the network, it has to establish private keys with every node on the path to the destination node. Due to the fact that in the routing table all nodes on the path to the destination node are stored, a message can be generated using this path in order to exchange the keys with all nodes at once.

In the first step, a symmetric key $SK$ is generated for every node on the route. In conjunction with a symmetric 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 to determine which symmetric key to use for encrypting the received message. After creating these keys the sending node generates a message $M_x$ by encrypting the symmetric key $SK_x$ with the public key $PK_x$ (the node identifier) of the last node $N_x$ on the path. Subsequently, the sending node encrypts the concatenation of the node identifier $PK_i$ and the symmetric key $SK_i$ of the node $N_i$ with the corresponding node identifier $PK_i$. The concatenation of the node identifier $N_{i+1}$ of the successor node and the message $M_{i+1}$ is encrypted with the symmetric key of node $N_i$. The concatenation of both encryptions builds the message $M_i$. This process is repeated from the destination node $N_x$ to the first node on the path $N_1$ and results in message $M_1$.

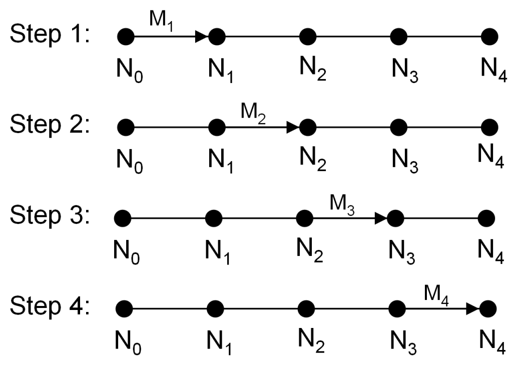


Figure 4: Key exchange

$$\begin{aligned} M_x = &\ [PK_x, SK_x]_{PK_x} \\ M_{X-1} = &\ [PK_{x-1}, SK_{x-1}]_{PK_{x-1}} \\ &\ [[PK_x, SK_x]_{PK_x}]_{SK_{x-1}} \\ M_1 = &\ [PK_1, SK_1]_{PK_1}[M_2]_{SK_1} \end{aligned}$$

The message $M_1$ is sent to the first node on the path. After decrypting the node identifier and the symmetric 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 symmetric key. It stores the symmetric 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 can obtain its key for the symmetric encryption (see figure 4).

To hide the path length of the anonymous communication, the messages need to have a fixed length. This can be achieved by putting the actual encrypted node identifier concatenated with the encrypted symmetric key at the end of the message so that the message length remains the same. To notify the last node on the path that it is the destination node a special mark in the message has to be set. This can be done by setting the next node identifier to zero. By adding dummy bits in the message and limiting the maximum length of the path, a fixed message size for all messages can be guaranteed. In order to avoid that the destination node is revealed by not forwarding the key exchange message it forwards a dummy key exchange message to extend the path by a few hops.

## 4.4 Communication protocol

To send data anonymously using the Acimn protocol the path to the destination node has to be known (see section 4.1). The symmetric keys used in the communication protocol are established by only one key exchange message that is sent over the communication path as described in section 4.3. Both during the key establishment phase and during the communication phase the one hop communication is executed using the DC net approach as described in section 4.2.

If node $N_0$ wants to send the data $D$ to the destination node $N_x$ multi hop over path $P$ it builds a communication message as follows: For all $0 < i \leq x$ the message $M_i$ is build by the concatenation of the encrypted message counter of node $N_i$ on the path, the encrypted message counters $P_i$ of the following nodes on the path, and the data destined for the destination node encrypted successively by all the symmetric 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:

$$\begin{aligned} P_x &:= [-1, P_{x+1}]_{SK_x} \\ P_i &:= [MC_{i+1}, P_{i+1}]_{SK_i} \end{aligned}$$

The data is encrypted layer by layer using the exchanged symmetric keys:

$$\begin{aligned} D_x &:= [D]_{SK_x} \\ D_i &:= [D_{i+1}]_{SK_i} \end{aligned}$$

After creating the message, the first node sends the message $M_1$ into the group, which contains the node with the node identifier $N_1$, representing the first node on the path, using the one hop communication protocol described in section 4.2. 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 symmetric 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. The next node of the path $N_{i+1}$ is stored together with the symmetric key $SK_i$ (see section 4.1). Node $N_i$ sends the newly build message $M_{i+1}$ to node $N_{i+1}$. This process is repeated until the message $M_x$ reaches the destination node $N_x$. While decrypting the message counters $P_x$ the Node $M_x$ detects that it is the destination node. By enlarging the path with dummy nodes the real destination node can be covered.
In almost the same manner as in the key exchange protocol (section 4.3) a fixed message size can be guaranteed. This can be done by limiting the maximum hop distance of the communication partners and fixing the amount of data sent in each message. To obtain the same message size for all $P_i$, for $0 < i \leq x$, every node $N_i$ on the path adds random bits at the end of $P_{i+1}$. When using a block cipher these random bits have no effect on the the other encrypted message counters.
To make the communication protocol robust to message loss, a series of message counter is stored. In this series the encryptions of the message counters $[MC], [MC + 1], \cdots, [MC + s]$ 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 $[MC + i + 1], \cdots, [MC + i + s]$.

### 4.5 Response channel

The basic Acimn protocol only provides a communication channel in one direction. In this section a procedure for the backward communication is presented.
If node $N_0$ wants to receive a message anonymously from node $N_x$ it chooses a path starting at node $N_0$ that goes through node $N_x$ and ends again in $N_0$. Node $N_0$ then initiates a key exchange with the nodes on this path as described in section 4.3. After this key exchange node $N_0$ can send messages to $N_x$ (see section 4.4). To receive messages anonymously from node $N_x$, node $N_0$ has to give the path in form of a list of encrypted message counter $P_x$ to $N_x$.
Assume that node $N_0$ sends an anonymous message to $N_x$. If $N_0$ wants a response to this anonymous message it sends the backwards path $P_x$ in the data field $D$ to $N_x$. Because every message counter of the backwards path can only be used once, node $N_0$ may send several backwards paths to $N_x$ if it wants to receive more than one message of $N_x$. Node $N_x$ can now send a message with data $D$ to $N_0$ by using the backwards path $P_x$. In the following the construction of the message is shown, where $x < i < m$ with $N_m := N_0$. The backwards path is build by node $N_0$ as follows:

$$\begin{aligned} P_i &:= [MC_{i+1}]SK_{i+1}[P_{i+1}]_{SK_i} \\ P_m &:= [MC_m]SK_m. \end{aligned}$$

The data field is constructed by:

$$\begin{aligned} D_x &:= [D]_{SK_x} \\ D_i &:= [D_{i-1}]SK_i. \end{aligned}$$

$N_x$ now constructs the following message:

$$M_x := [MC_{x+1}]SK_{x+1}, P_{x+1}, D_x$$

At every node on the path the message is replaced by:

$$M_i := [MC_{i+1}]_{SK_{i+1}}, P_{i+1}, D_i$$

Node $N_m = N_0$ receives the message $M_m$ and rebuilds the data $D$ by decrypting the received $D_{m-1}$ step by step with the symmetric keys $SK_{m-1}, \cdots SK_x$.
Compared to the communication protocol the nodes on the path execute the same computations as during the communication from $N_0$ to $N_x$. Only $N_0$ and $N_x$ has to execute different computations.

## 5 Performance evaluation

Due to the use of the DC net and the Mix net approach a message overhead is generated. By using the DC net approach in every communication step three messages have to be exchanged for sending one message by one hop. So to send data of size $d$ messages with the data size of $3 * d$ has to be exchanged. The traffic used for the exchange of the symmetric keys between the nodes in a DC net group and the reservation of communication slots is negligible.
While using layered encryption only during the key establishment phase a message overhead is generated. For this one key exchange message is sent over the whole communication path. The data size of the key exchange message depends on the used key lengths. While using a 3DES key length of 112 Bit and a RSA key length of 384 Bit with an upper hop limit of 20 hops the key exchange message has only $l = 1178$ Byte length. This key exchange message is sent over $n$ hops, where $n$ is the length of the communication path. If the communication path is used to send the data size $d$ the mean overhead produced is $\frac{d}{m}$. Together with the overhead of the one hop communication protocol the overall overhead for data with size $d$ is $2 + \frac{3*l}{d}$.
Based on this values the upper bound for the message overhead is $O(5)$ in the case when every communication path is used only once for sending a packet of size 1 kilobyte. On the other hand the lower bound is $\Omega(2)$ in the case when the communication paths are used to send messages of size $d$ for $d \to \infty$.
There are three different calculations performed during the communication. To clarify the computational power to perform the calculation the encryption and decryption rate of the different algorithms is identified. The reference hardware is a Dell Latitude D610 laptop with an Intel Pentium M processor with 2 GHz and 1 GByte main memory.

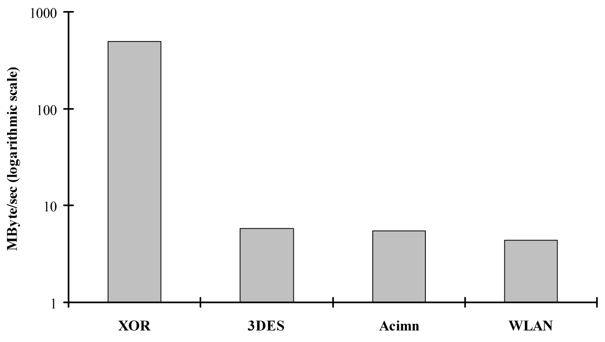


Figure 5: Throughput of Acimn during communication

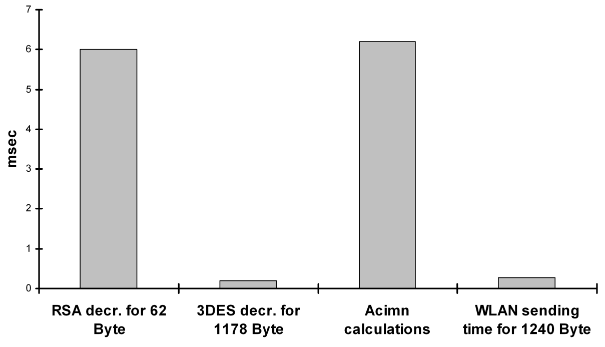


Figure 6: Time to process a key exchange message

While using the DC net approach in every communication step there is a bit by bit XOR computation of the message and the shared secret keys of all DC net group members. After having received the messages of all group members each participant calculates again the XOR of the received messages resulting in the original message. These operations do not consume much computational power. On the reference hardware a throughput of 497 MByte/sec is achieved.
During the multi hop communication every node on the path performs a symmetric decryption. As reference the 3DES algorithm is used. The gained throughput for the encryption and decryption is 5.74 MByte/sec. So both, the XOR and the 3DES algorithm, reach a better throughput than ordinary wireless hardware based on the 802.11g standard. A comparison of the different throughputs is shown in figure 5.
For the asymmetric encryption and decryption the RSA algorithm is used. The throughput of the encryption is 205.13 KByte/sec and the throughput of the decryption is 10.08 KByte/sec. But only a small part that consists of the public key of the node and the new shared symmetric key has to be decrypted using the RSA algorithm. These keys together have approximately 62 Byte, so that the RSA decryption only takes 6 msec while the rest of the message is decrypted with the more efficient 3DES algorithm. The comparison of the time to send a key exchange message using 802.11g standard and the time to process the Acimn calculations is shown in figure 6.

## 6 Security analysis

Acimn provides anonymity through the use of different techniques.
Messages sent on one link can not be backtracked to the sender because of the use of the DC net approach. A passive attacker only can detect that a message is sent inside a group but it can neither determine which node is the originator of the message nor the content of the message because the messages are encrypted.
Additionally the path between the sending and the destination node is hidden by using layered encryption. Therefore an inner node of the path does not know any of the predecessor nodes nor any of the successor nodes except the direct successor on the path. A single passive attacker is able to detect neither the sender nor the receiver of a message because the messages are encrypted and the included personal information is not readable. A powerful passive attacker which can detect all messages sent is also not able to determine the destination or the path, because the messages change their appearance at every hop. If there is only little traffic the probability that messages belong to one communication path gets higher so that a powerful attacker can spot the path on which the sender and the receiver of a message are located. Dummy traffic helps to avoid these situations.
Since messages are encrypted, the messages all have the same size, and messages change their appearance on every hop to the destination no personal information can be gained. Only the destination node is able to read the content of the message.
The network is not notedly more vulnerable to denial of service attacks. Due to the limited group number of three in the on hop communication the messages sent using the protocol is only three times higher than without using it. Indeed there is asymmetric decryption during the key establishing phase at every node. But the message is only forwarded by an intermediate node if the decryption contains its identifier. For that the attacker has to correctly encrypt the messages. If a replay of messages is avoided by buffering key exchange messages the attacker has to do the same computations than all the nodes it wants to disrupt. The following list gives different attacks and their influence to Acimn.

**Denial of Service attack** The aim of this attack is to destroy the availability of the target device or target service. Due to the efficient calculations of the Acimn protocol the impact of a DOS attack is not noticeable higher compared to normal wireless communication.

**Replay attack** By resending messages an attacker can try to get access to a session or resources. In Acimn the encrypted message counter prevents such a replay attack. Only the node that established the communication path can build the encryption of the incremented message counter.

**Message coding attack** By analyzing the input and output traffic of a node an attacker can match packets by their coding or appearance. Because of the decryption of the packets at every hop and the fixed packet size no mapping is possible while using Acimn.

**Collusion attack** At least one node on the Acimn path is needed to prevent colliding attackers from revealing the communication partners.

**Packet volume and packet counting attack** Through counting the packets going through a node and measuring the amount of traffic send an attacker can track packets on the way to the destination node. Acimn can not prevent from this attack overall. But anyway it makes this attack difficult since the communication direction is unclear and with additional dummy traffic a reliable counting is not possible.

**Message delaying attack** Delaying packets has no effect on the execution of Acimn.

**Flooding attack** In flooding attacks the attacker sends many packets over the target node in order to separate packets of the victim going over this node. But to reveal the communication partners the attacker has to separate the packets at every link on the path from the source to the destination while the DC net approach of Acimn additionally complicates the analysis for the attacker.

**Intersection attack** If an attacker observes a node for some time it can detect habits of the users. Because of the encryption used in Acimn the attacker can not observe the content of messages. Even it can not be sure if the victim node is communicating because of the DC net approach.

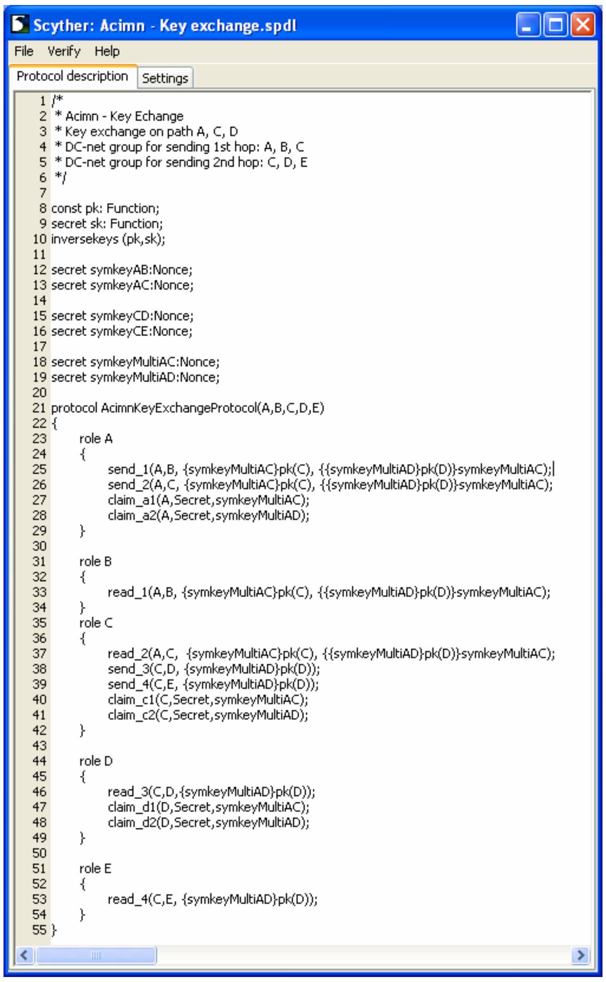
**Timing/Latency attack** In a timing attack the latency of a reply of the destination node is calculated. This value is compared with a list of earlier calculated latencies of devices. Due to the fact that for the backward communication a new path has to be establishes it is not possible to map traffic going from node $A$ to $B$ to the backwards traffic from $B$ to $A$. Because of this the attacker is not able to calculate the latencies.

**Clogging attack** In clogging attacks the attacker observes the messages between the destination node $A$ and the predecessor node $B$. Then it chooses a node $C$ of the network randomly and floods this node with packets. If the traffic between $A$ and $B$ collapse the node $C$ belongs to the communication path with high probability. Acimn can not totally prevent from this attack but makes it more complex because for one hop communication the group of potential senders and receivers can only be narrowed to three nodes.

For the symmetric and asymmetric encryption and decryption any cryptographic algorithm as well as any key size can be used. Acimn is not linked to specific ones. If it is assumed that the used cryptografic algorithm for the asymmetric encryption can not be broken, an attacker can not receive the symmetric keys that are exchanged during the key exchange phase. If the cryptographic algorithm for the symmetric encryption is secure, then also the data during the communication remains confidential. Both are proved with the Scyther security protocol checker (Cremers 2006). The Scyther code of the key exchange protocol is shown in figure 7 and the output is shown in figure 8.

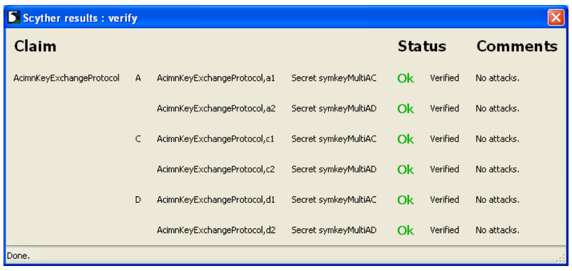
## 7 Conclusion

Acimn, an anonymous routing protocol was presented. It bases on the dining cryptographers network for every one hop communication step combined with layered encryption for multi hop communication. With the response channel also a backwards communication is possible where the responding node receives an anonymous path for sending its data.
While using the Acimn protocol the loss of performance is maintainable due to the use of symmetric encryption during the communication. The message overhead is limited to $O(5)$ and is at least only $\Omega(2)$. The protocol is secure against most of the traffic analysis attacks. With a more complex clogging attack it might be possible to reveal the communication partners. But for this a high amount of traffic has to be sent between the communication partners



```
/*
 * Acimn - Key Echange
 * Key exchange on path A, C, D
 * DC-net group for sending 1st hop: A, B, C
 * DC-net group for sending 2nd hop: C, D, E
 */

const pk: Function;
secret sk: Function;
inversekeys (pk,sk);

secret symkeyAB:Nonce;
secret symkeyAC:Nonce;

secret symkeyCD:Nonce;
secret symkeyCE:Nonce;

secret symkeyMultiAC:Nonce;
secret symkeyMultiAD:Nonce;

protocol AcimnKeyExchangeProtocol(A,B,C,D,E)
{
	role A
	{
		send_1(A,B, {symkeyMultiAC}pk(C), {{symkeyMultiAD}pk(D)}symkeyMultiAC);
		send_2(A,C, {symkeyMultiAC}pk(C), {{symkeyMultiAD}pk(D)}symkeyMultiAC);
		claim_a1(A,Secret,symkeyMultiAC);
		claim_a2(A,Secret,symkeyMultiAD);
	}

	role B
	{
		read_1(A,B, {symkeyMultiAC}pk(C), {{symkeyMultiAD}pk(D)}symkeyMultiAC);
	}
	role C
	{
		read_2(A,C,  {symkeyMultiAC}pk(C), {{symkeyMultiAD}pk(D)}symkeyMultiAC);
		send_3(C,D, {symkeyMultiAD}pk(D));
		send_4(C,E, {symkeyMultiAD}pk(D));
		claim_c1(C,Secret,symkeyMultiAC);
		claim_c2(C,Secret,symkeyMultiAD);
	}

	role D
	{
		read_3(C,D,{symkeyMultiAD}pk(D));
		claim_d1(D,Secret,symkeyMultiAC);
		claim_d2(D,Secret,symkeyMultiAD);
	}

	role E
	{
		read_4(C,E, {symkeyMultiAD}pk(D));
	}
}
```

Figure 7: Scyther code



```
Claim                                                                                      Status            Comments

AcimnKeyExchangeProtocol  A  AcimnKeyExchangeProtocol,a1  Secret symkeyMultiAC  Ok  Verified  No attacks.

                             AcimnKeyExchangeProtocol,a2  Secret symkeyMultiAD  Ok  Verified  No attacks.

                          C  AcimnKeyExchangeProtocol,c1  Secret symkeyMultiAC  Ok  Verified  No attacks.

                             AcimnKeyExchangeProtocol,c2  Secret symkeyMultiAD  Ok  Verified  No attacks.

                          D  AcimnKeyExchangeProtocol,d1  Secret symkeyMultiAC  Ok  Verified  No attacks.

                             AcimnKeyExchangeProtocol,d2  Secret symkeyMultiAD  Ok  Verified  No attacks.
Done.
```

Figure 8: Scyther output

over a longer period of time. This can be prevented if multiple path exist so that a blocking of only one node does not tend to traffic slump. A precise treatment will be part of future work.

## References

Berthold, O., Federrath, H. & Köpsell, S. (2000), Web MIXes: A system for anonymous and unobservable Internet access, *in* H. Federrath, ed., 'Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability', Springer-Verlag, LNCS 2009, pp. 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. & 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, pp. 320–327.

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

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

Chen, T.-W. & Gerla, M. (1998), Global state routing: A new routing scheme for ad-hoc wireless networks, *in* 'IEEE International Communications Conference, ICC98, June 1998, Atlanta, GA, USA', IEEE, pp. 171–175.
**URL:** *http://www.ics.uci.edu/ atm/adhoc/paper-collection/gerla-gsr-icc98.pdf*

Cremers, C. (2006), Scyther - Semantics and Verification of Security Protocols, Ph.D. dissertation, Eindhoven University of Technology.

Danezis, G., Dingledine, R. & Mathewson, N. (2003), Mixminion: Design of a Type III Anonymous Remailer Protocol, *in* 'Proceedings of the 2003 IEEE Symposium on Security and Privacy'.

Dingledine, R., Mathewson, N. & Syverson, P. (2004), Tor: The second-generation onion router, *in* 'Proceedings of the 13th USENIX Security Symposium'.

Goel, S., Robson, M., Polte, M. & Sirer, E. G. (2003), Herbivore: A Scalable and Efficient Protocol for Anonymous Communication, Technical Report 2003-1890, Cornell University, Ithaca, NY.

Goldschlag, D. M., Reed, M. G. & Syverson, P. F. (1996), Hiding Routing Information, *in* R. Anderson, ed., 'Proceedings of Information Hiding: First International Workshop', Springer-Verlag, LNCS 1174, pp. 137–150.

Jiang, S., Vaidya, N. H. & 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)', Vol. 2, IEEE, pp. 873– 877.

Johnson, D. B., Maltz, D. A. & Hu, Y.-C. (2004), The dynamic source routing protocol for mobile ad hoc networks (dsr), Internet-draft, IETF MANET Working Group. Expiration: January 2005.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt*

Kong, J. & 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, pp. 291–302.

Perkins, C. E., Belding-Royer, E. M. & Das, S. R. (2003), Ad hoc on-demand distance vector routing protocol, Internet-draft, IETF MANET Working Group. Expiration: August 17, 2003.
**URL:** *http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt*

Reiter, M. & Rubin, A. (1998), 'Crowds: Anonymity for web transactions', *ACM Transactions on Information and System Security* **1**(1).

Seys, S. & 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, pp. 133–137.

Shields, C. & 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', ACM Press, New York, NY, USA, pp. 33–42.

Sirer, E. G., Polte, M. & Robson, M. (2001), Cliquenet: A self-organizing, scalable, peer-to-peer anonymous communication substrate, Technical Report TR2001, Cornell University, Computing and Information Science.

Sy, D., Chen, R. & Bao, L. (2006), Odar: On-demand anonymous routing in ad hoc networks, *in* 'The Third IEEE International Conference on Mobile Ad-hoc and Sensot Systems (MASS)'.

Syverson, P., Reed, M. & Goldschlag, D. (2000), Onion Routing access configurations, *in* 'DARPA Information Survivability Conference and Exposition (DISCEX 2000)', Vol. 1, IEEE CS Press, pp. 34–40.

Waidner, M. & 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, p. 690.

Zhang, Y., Liu, W., Lou, W. & Fang, Y. (2006), Mask: Anonymous on-demand routing in mobile ad hoc networks, *in* 'Transactions on Wireless Communications', Vol. 21, IEEE, pp. 2376–2385.