Doubly-Anonymous Crowds: Using Secret-Sharing to achieve Sender- and Receiver-Anonymity*

Stefan Rass†, Raphael Wigoutschnigg, and Peter Schartner
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Carinthia, Austria
{stefan.rass,raphael.wigoutschnigg,peter.schartner@aau.at

Abstract

Anonymous communication is usually achieved by using overlay networks and public-key cryptography. Besides well-established protocols, based on onion routing, various competing solutions have been proposed in the literature. In this work we extend the Crowds system by providing sender and adjustable receiver anonymity. Our solution is robust against a coalition of passive adversaries having intermediate nodes in the network conquered in an attempt to discover the sender and receiver of an observed communication. Absolute privacy for the receiver is achievable asymptotically in the size of the network, which is beyond the control of a sender. We demonstrate that a certain level of anonymity is achievable in finitely large networks and under full control of the sender.

Keywords: Anonymity, anonymous communication, crowds, mix-net, secret-sharing.

1 Introduction

The problem of anonymous communication along with first protocols has been pioneered by the seminal work of Chaum [2], who came up with a most elegant solution nowadays known as the *dining cryptographers problem* [3]. Among the most popular solutions for anonymous communication is *onion routing (OR)*, with TOR [4] being a widely used implementation. Besides this, alternative solutions exist that differ from onion routing by either protecting the sender or receiver only (e.g. Crowds; see Section 2) or by avoiding public-key cryptography (and hence computational intractability assumptions). In this work, we extend the Crowds approach (to be described later) from sender-anonymity only to both, sender- and adjustable receiver-anonymity. We achieve this by running a Mix-net like protocol on top of Crowds, where the public-key encryption used for OR is replaced by a one-round perfectly secure transmission protocol (i.e. multipath transmission). Based on a sufficiently connected network (e.g. an overlay network), we achieve information theoretic security for the receiver under some restrictions. Our intention here is thus improving the Crowds system to protect the receiver's identity as much as it protects the sender. Sender anonymity is directly inherited from the Crowds system. Regarding receiver anonymity, we will prove that in a (finite size) network, the sender can anonymously communicate with a designated receiver in such a way that a passive threshold adversary, upon eavesdropping, can determine the receiver with no better chance than pure guessing. We call the receiver to be "anonymous beyond suspicion" (cf. Section 3.3. Equally strong protection of the receiver's identity can be achieved by combining Crowds with Chaum's Mix-Nets (called a Crowds-Mix; details of Mix-Nets are sketched in Section 2). However, receiver anonymity in the way that we demand in this work is achievable only asymptotically if the network size grows towards infinity. Because the network size is in no way under the initiator's control, our

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 2, number: 4, pp. 27-41

*This paper is an extended version of the work originally presented at the 6th International Conference on Availability, Reliability and Security (ARES'11), Vienna, Austria, August 2011 [1].

†Corresponding author: System Security Research Group, Institute of Applied Informatics, Universitaet Klagenfurt, Universitaetsstrasse 65-67, 9020 Klagenfurt, Austria, Tel: +43(0)46327003715, Email: stefan.rass@aau.at, Web: `http://www.syssec.at/stefan_rass/`

main contribution is thus to show how receiver anonymity can be realized in a given finite size network, with parameters that the sender can control.

This article is a threefold extension to previous work [1], which discussed only one-directional anonymous communication from the sender to the receiver. This issue is resolved in this work by describing a method for "anonymous replies" to the "anonymous transmissions" made possible by the protocol in [1]. In addition, we will discuss the robustness of our protocol w.r.t. node failure, thus filling another gap left out in previous research. Finally, the protocol as described in [1] requires padding, but does not go into details about how this padding should look like. Here, we demonstrate how a carelessly chosen padding can reveal intermediate nodes to the adversary (even the receiver in the worst case). Section 6.2 is devoted to a solution for this issue.

2 Related Work

In 1981, Chaum [2] introduced the idea of using special proxy servers, called *mixes*, to achieve a certain degree of anonymity for the sender and the receiver of electronic mails. The idea is as follows: the sender chooses a sequence $R_1, R_2, \ldots$ of mixes, which are adjacent relay nodes forming a path from the sender to his intended receiver. For the relay node R_i, having the public key pk_i, he encrypts the address of the next hop R_{i+1} in a public-key fashion using pk_i. In doing so, he prevents any relay other than R_i to discover the bit $R_i \rightarrow R_{i+1}$ on his chosen path. Vice versa, R_i gets to know only the next hop and the origin R_{i-1} of his received packet, but can neither discover the first node (i.e. the sender), nor the last node (i.e. the receiver). The actual payload is encrypted with the receiver's public key. If one of the intermediate mixes is unavailable (for whatever reason), then the message cannot be decrypted correctly. The benefit from this is, that the actual data packet looks different at each mix, thus impeding a traffic analysis. The receiver cannot reply unless the sender tells him his identity or he gets a so called *reply block*, which is a chosen and encrypted route from the receiver back to the sender. The actual reply is sent using the very same procedure as before and using the reply block prepared by the sender.

Since the nineties, Chaum's idea has been picked up several times and resulted in remailer systems, designed to send e-mails anonymously. Based on their anonymity properties, they are classified as *type I*, *type II* and *type III* remailers, each of which has different anonymity properties. Known examples are *cypherpunk* [5], *mixmaster* [6], *Babel* [7] and *mixminion* [8].

A noteworthy protocol was published 1996 by Goldschlag, Reed and Syversion [9–11]. This proposal has become known as *onion routing*, because of its layered construction that is closely related to the Mix-net system from Chaum. However, it is primarily designed for TCP-based communication, which allows a wider range of applications. Unlike Chaum's Mix-net system, onion routing is designed for real-time and bidirectional communication, which is more difficult to obtain. Like in the Mix-net system, every onion rounter only knows his predecessor and successor.

During the *connection setup phase* an onion (constructed as in the Mix-net system) is sent through the network to the receiver to open a virtual channel. This onion consists of the routing information (like in the Mix-net system), but contains additional keys for the onion routers to be used with symmetric encryption. The first phase terminates with a virtual channel being set up and each intermediate node storing symmetric keys for subsequent encryption of payload. Replies are sent over exactly the same channel (as opposed to a Mix-net, where the sender can call for a possibly different return path via the reply block). Also, as the protocol is required to exhibit real-time performance, the relay nodes are not able to reorder the messages like in the Mix-net system, which makes traffic analysis easier.

In 2004, the TOR system [4, 12] was developed by the onion routing team to provide a real-time system for bidirectional communication with perfect forward secrecy, because the original onion routing protocol did not provide this. A noteworthy difference to onion routing is that TOR uses an incremental

(telescoping) path-building design and does not use one single onion to build the virtual channel. The sender negotiates the needed session keys with every node on the path.

Another approach, which we extend in the rest of the paper, was published in 1998 by Reiter and Rubin and was named *Crowds* [13]. While mixnets and onion routing use a predetermined path from the sender to the receiver, Crowds employs some sort of "probabilistic routing". Assume a fully connected network having n nodes, and let w.l.o.g. node 1 be the sender and node n be the receiver. Then, if an intermediate node i receives the payload tagged with n as addressee, it randomly decides to forward it either to the receiver with probability $1 - p_f$, or to forward it to another node $j \neq n$ with probability $p_f > 0.5$. It is easy to see that if $0.5 < p_f < 1$, then any node that receives a packet cannot be sure whether the sender has actually been the true origin of the packet. On the other hand, every intermediate node gets to know the true receiver. It follows that the system provides sender anonymity, yet no receiver anonymity. Hereafter, in alignment with the terminology used in the Crowds system, we shall refer to an intermediate relay node as a *jondo* (referring to the usual anonymous person "John Doe").

None of the systems described before has unconditional sender anonymity. Chaum published a protocol with unconditional sender anonymity in 1988, which is called *Dining Cryptographers Protocol* [3]. The disadvantage of this method is the high communication overhead to transmit one bit. His idea was picked up and ended in several advanced protocols including *Herbivore* [14] and P^5 [15].

3 Preliminaries

It is important to remark the difference between the term *sender* and *initiator*. As we are dealing with anonymous communication, we shall refer to the *initiator* as the one instance that starts the process of anonymous communication with some receiver. Contrary to this, a sender in our context can be any entity that transmits data to some other instance. Hence, there is only one initiator (the original sender, often called "Alice") and multiple senders, relaying the message to the final receiver.

3.1 Notions of Anonymity

The following definitions are borrowed from Pfitzmann and Hansen [16]: The *anonymity set* is the set of all possible subjects which could have done an action (sending, receiving, etc.). *Sender anonymity* means that a special subject within the sender anonymity set cannot be identified as the sender of a message (initiator of the communication). *Receiver anonymity* means that a special subject within the receiver anonymity set cannot be identified as the receiver of a message. Two or more items (subjects, messages, actions, etc.) are *unlinkable*, if an attacker cannot reliably recognize any relation among them.

3.2 Attacker Model

We assume a passive attacker with the ability to compromise up to k nodes in our system simultaneously. He is computationally unbounded and is able to read all traffic of the compromised nodes, but does not attempt to modify, block, insert or otherwise actively tamper with the protocol flow.

3.3 Degree of Anonymity

Adopting the notions introduced in Reiter and Rubin [13], three degrees of anonymity will be addressed in this paper. An instance enjoys *absolute privacy*, if an attacker will not even see any subject taking action. Weaker than absolute privacy is anonymity *beyond suspicion*, which is given if the sender (receiver) does not appear more likely to be the initiator (addressee) of a communication than any other subject of the anonymity set. Anonymity up to *probable innocence* is achieved, if from the attackers point of view,

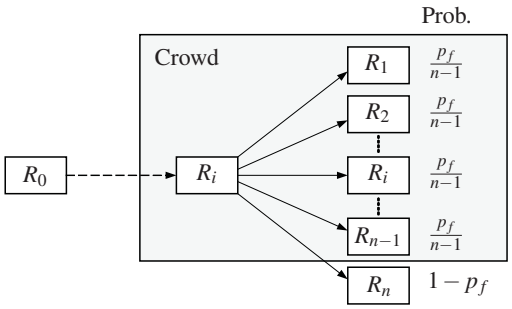

Figure 1: Forwarding in the Crowds system

the sender of a message is not more likely to be the initiator of a message than not. It is easy to see that probable innocence is implied if anonymity is beyond suspicion, and hence the last notion is the weakest of the three.

4 The Crowds system

The *crowds* system was published in 1998 by Reiter and Rubin [13] and differs from the Mix-net system in how the path is chosen through the network. Every node in the system (called jondo, John Doe) can be an initiator of a message or a relay. If a jondo (initiator I) wants to send a message to a receiver R (the receiver is not part of the crowd, e.g. a HTTP server), he randomly selects a jondo and sends the message and the identity of the receiver to this jondo encrypted, so that no eavesdropper is able to see what is being sent. After the arrival of the message the jondo decrypts it and decides what to do next. He throws a biased coin and forwards the message with a probability $p_f > 0.5$ to another randomly chosen jondo, otherwise he forwards directly to the receiver. Figure 1 displays the possible transitions from jondo i to other jondos $1, 2, \ldots, n-1$ (uniformly at random to each of them with probability $\frac{p_f}{n-1}$ or directly to the receiver (with chance $1-p_f$). The value p_f is a fixed system-wide parameter and is known to all jondos.

If transmitted to another relay, this jondo (node i) does the same (throwing a biased coin and forwarding either to another relay or the receiver n). Because of this mechanism, a jondo cannot decide whether his predecessor is the originator of the message or another relay. The communication between the jondos is pairwise encrypted, therefore the incoming message is not linkable to the outgoing message, because another key is used to encrypt the message for transmission to the next hop. This system does not provide receiver anonymity in case of a malicious jondo, because every jondo has to know the receiver address in case of transmitting the message to the designated receiver. However a local eavesdropper is not able to reveal the identity of the receiver nor the initiator because the messages are encrypted hop-to-hop. Sender-receiver unlinkability is guaranteed in the Crowds system.

The sender anonymity of the Crowds system in case of collaborating jondos is *probable innocence*. If the network grows to infinity, then *absolute privacy* is achieved. Because every jondo on the path, the message takes from initiator (original sender) to the receiver, is able to reveal the identity of the receiver. The identity of the receiver is hidden, only if no malicious jondo gets the message for retransmission. If the networks grows, the probability for revealing the identity of the receiver gets lower if the number of malicious nodes remains constant.

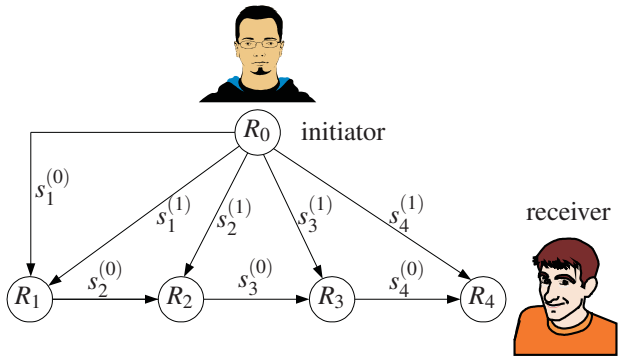

Figure 2: Receiver anonymity using secret-sharing

5 Our Approach

To conceal a message m, we consider a $(2,2)$-sharing scheme for its simplicity and unconditional security. Shamir's polynomial secret sharing [17] is one possibility, yet we use a simple one-time pad like construction by choosing $s^{(0)}$ at random and putting $s^{(1)} := m \oplus s^{(0)}$ with $\oplus$ being the bitwise exclusive-or (the generalization of this towards an (n,n) XOR-sharing is straightforward). Assume a route to be described by a sequence $R_0, R_1, R_2, \ldots, R_r$ (sequence of identities or addresses), where the initiator is denoted by R_0 and the receiver denoted by R_r. Every router R_i (which is a jondo) along the path gets two shares (one from his predecessor R_{i-1} and one from the initiator). The shares are sent using the Crowds-protocol (cf. Figure 2). The routers R_i are called *routing jondos* for simplicity. The initiator and the receiver are indistinguishable from the other routing jondos and therefore are named R_0 and R_r;

In principle, we have the following scheme, according to which a message m is transmitted. Each router R_i for $i = 1, 2, \ldots, r$ along the way receives (at least) two shares $s_i^{(0)}$ and $s_i^{(1)}$ such that $s_i^{(0)} \oplus s_i^{(1)} = R_{i+1} \| s_{i+1}$. The first router R_1 receives $s_1^{(0)}$ and $s_1^{(1)}$ with $s_1^{(0)} \oplus s_1^{(1)} = (R_2, s_2^{(0)})$ such that it can pass onward s_2 to its successor R_2. The *last* router gets $s_r^{(0)}, s_r^{(1)}$ which create $s_r^{(0)} \oplus s_r^{(1)} = null \| m$, where m is the message transmitted. R_r knows that he is the receiver of the message m because no next hop is defined in the address part. Table 1 sketches an example incarnation of this process.

Router	Information
R_1	$s_1^{(0)} \oplus s_1^{(1)} = R_2 \| s_2^{(0)}$
R_2	$s_2^{(0)} \oplus s_2^{(1)} = R_3 \| s_3^{(0)}$
$\vdots$	$\vdots$
R_i	$s_i^{(0)} \oplus s_i^{(1)} = R_{i+1} \| s_{i+1}^{(0)}$
$\vdots$	$\vdots$
R_r	$s_r^{(0)} \oplus s_r^{(1)} = null \| m$

Table 1: Onion-routing like message forwarding with secret sharing in place of public-key encryption

It is most trivial for the initiator to determine all $s_i^{(0)}, s_i^{(1)}$ for $i = 1, 2, \ldots, r$ by a bottom-up computation. Sender anonymity is obtained by using the Crowds system to deliver the shares (see Figure 3). It is easy to generalize the above construction towards using more than two shares, say $s_i^{(0)}, \ldots, s_i^{(t-1)}$ for the i-th forwarding router R_i, in the very same manner as described above (see Figure 3 for an illustration). This variant is the one we are going to consider for our security analysis.

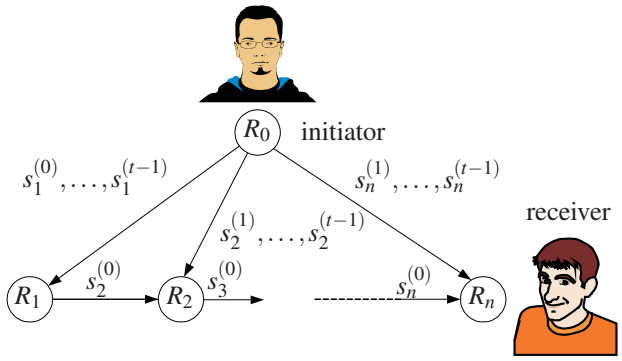

Figure 3: Generalized scheme

6 Security Analysis

Security of anonymous communication is twofold, and we shall consider the sender and receiver anonymity separately.

Sender Anonymity: As we are basically using the Crowds system *as it is*, the security of our scheme, regarding the initiators's anonymity, is inherited and equal to that of the Crowds system.

Receiver Anonymity: We sketch the idea using an example before making things rigorous below. Let us consider R_2 in Figure 2 for illustration: until R_2 receives the information $s_2^{(1)}$, it cannot be sure to be the final receiver. If not, then it can neither be sure that R_3 is the final receiver, because R_2 does not know whether the share $s_3^{(1)}$ indicates R_3 as the final addressee. So, assuming that the paths along which the packets $s_1^{(0)}$ and $s_1^{(1)}$ travel are disjoint, then the receiver's identity remains perfectly hidden. However, since we rely on the Crowds system for sender anonymity, we cannot reliably ensure the paths to be disjoint. The actual problem occurs whenever a node gets shares which can be used to calculate the message. Suppose that, in Figure 2, R_3 gets to know $s_4^{(1)}$ (it knows s_4 anyway), then R_4 is discovered by R_3 as the final receiver. Our task therefore boils down to find out the likelihood of a node getting to see all shares intended for the receiver. Every combination of shares which does not include the ones for the receiver, does not reveal the receivers identity. We will employ some well-known facts from the theory of Markov-chains to investigate this.

According to the scheme, the initiator chooses a path with at least one hop (the receiver), along which the packets travel (for Figure 2, this would be R_1, R_2, R_3, R_4). Recall the generalization depicted in Figure 3, and assume that the relay R_i requires t shares to discover the next hop R_{i+1}. Suppose further that the adversary has a (known) threshold of k and that the network graph is fully connected with encrypted links. It is obvious that the receiver's identity cannot be hidden reliably because all the shares are routed randomly through the crowd and an attacker can acquire all t shares and disclose the receiver. The scenario that we consider is the following:

- the adversary has a k-size subset A of nodes under his control,
- the initiator uses t shares for each hop R along the route to let the node R know to whom it shall pass the packet onwards.

The goal is to choose t sufficiently large such that the chance that an attacker (represented by the set A) gets enough shares (e.g. $s_4^{(0)}$ and $s_4^{(1)}$ or $s_3^{(0)}$, $s_3^{(1)}$ in Figure 2) is negligibly small, considering the random

forwarding regime of the Crowds system. The crowd is taken as a white-box, with the only public parameter $p_f > 0.5$. Technically, each share performs a random walk on the network, whose evolution is governed by the following transition matrix P (with entry p_{ij} indicating the probability of the packet traveling from node i to node j). Without loss of generality, denote the receiver of this share as node n.

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & \dots & n-1 & n \end{array} \\ \begin{pmatrix} \frac{p_f}{n-1} & \frac{p_f}{n-1} & \cdots & \frac{p_f}{n-1} & 1-p_f \\ \frac{p_f}{n-1} & \frac{p_f}{n-1} & \cdots & \frac{p_f}{n-1} & 1-p_f \\ & & & & \\ \frac{p_f}{n-1} & \frac{p_f}{n-1} & \frac{p_f}{n-1} & \frac{p_f}{n-1} & 1-p_f \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \end{array} = (p_{ij}) = P$$

It is important, however, to notice that *all* nodes, including the initiator and the receiver, are part of the crowd and as such indistinguishable peers. In the following, we are interested in the event of t shares ($s_r^{(0)}, \dots, s_r^{(t-1)}$ in our analysis) traversing the enemy's premises A within the Crowds-network.

Let h_{iA} be the probability that a random walk starting from node i ever *hits* the set A of (compromised) nodes within finite time. It is well-known from the theory of Markov-chains (cf. [18]), that the family $h_{1A}, h_{2A}, \dots, h_{nA}$ satisfies the equation system

$$h_{jA} = \sum_{i \in V} p_{ji} h_{iA}, \quad \text{for all } j = 1, 2, \dots, n \tag{1}$$

where $h_{jA} = 1$ for all $j \in A$ (trivially) and $h_{nA} = 0$, since the receiver n will surely not forward his message any further.

Since we consider t independent walks, the chance of all of them traversing A is simply

$$\Pr[t \text{ shares traverse nodes in } A] =: q = \prod_{i=1}^{t} h_{iA}, \tag{2}$$

assuming, without loss of generality, that the initiator uses the (not necessarily distinct) nodes $1, 2, \dots, t$ as entry points to the paths trough the crowd.

Observe that due to the equitable forwarding within the crowd (each node receives the message with equal likelihood), we have $h_{iA} = h_{iB}$ for any two sets A, B of size k that *do not* contain the node i (which is a jondo when analyzing the Crowds system). It follows that we can choose *any* (fixed) starting node i_0 and *an arbitrary* k-size set A_0 with $i_0, n \notin A_0$ (we can exclude the legitimate receiver n for obvious reasons) to determine $h_{i_0 A_0}$ from system (1) as a representative for any other h_{iA} with $i, n \notin A$ and $|A| = k$. In cases where $i \in A$, we have $h_{iA} = 1$.

Unfortunately, the set A is unknown to the initiator, but this causes no problem if t is chosen large. Since $|A| = k$ is fixed, let us choose t much larger than k. In fact, if the initiator uses N of his neighbors as starting points in the crowd, distributing the shares uniformly, then each neighbor gets to transmit a fraction of at least $\lfloor \frac{t}{N} \rfloor$ shares (cf. Figure 4). Since the adversary cannot compromise all of them, we get a refined upper-bound for the quantity q, which is

$$q = \prod_{i=1}^{t} h_{iA} \leq h_{i_0 A_0}^{\lfloor t/N \rfloor},$$

where $h_{i_0 A_0}$ can be computed easily from equation (1). Figure 4 illustrates this graphically: in each step, the receiver is reached with probability $1 - p_f$, and with likelihood p_f, the packet gets forwarded elsewhere at random in the crowd. The adversary has (possibly) compromised a subset A of the initiator's neighbors (or other nodes in the crowd). The bound obtained above tells the likelihood of this random

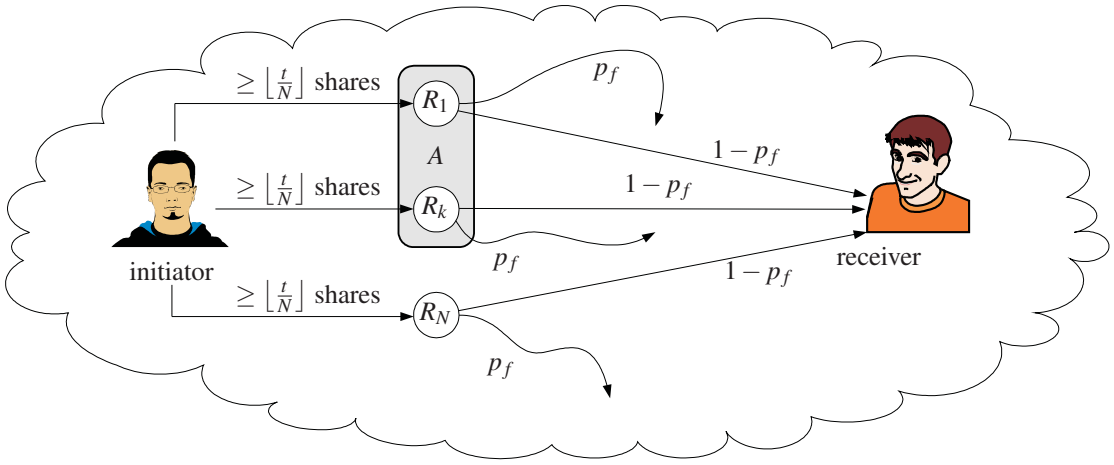

Figure 4: Random walk in the crowd, starting from the initiator's neighbors.

walk to hit A with sufficiently many trajectories so as to enable the adversary to discover some shared secret (address).

By choosing t sufficiently large to make $q < \frac{1}{n}$, we can ensure that the chance for the adversary to disclose the receiver by catching all his shares, is strictly less than correctly guessing the receiver's identity without any prior knowledge. Hence, our scheme can provide receiver anonymity beyond suspicion, except for a negligible chance of breaking the entire scheme. This chance is less promising than pure guessing, and the scheme becomes absolutely secure if the number of shares increases towards infinity. In that sense, our extension improves on the Crowds system, as the latter can achieve the same secrecy only for infinitely large networks.

Summarizing the preceding discussion, we have proved:

Proposition 6.1. *Let a network with n nodes be given and assume the described protocol, the sender enjoys anonymity in terms of probable innocence. Let the adversary be passive with threshold k, and assume that the receiver has $N > k$ neighbors and uses $t > k$ shares for each hop. Then the chance to discover the receiver's identity is no more than $h^{t/N}$, where h is the smallest component of the solution vector of system* (1) *for an arbitrarily chosen set A that does neither contain* 1 *nor n and an arbitrary index $i \notin \{1,n\}$. In particular, if $h < 1$, then receiver anonymity in terms of probable innocence is achieved by using $t \in \Omega(\log n)$ shares.*

6.1 Example

Consider a network with $n = 8$ nodes, and an adversary with threshold $t = 2$. We (arbitrarily) pick $i_0 = 4$, $A_0 = \{5,6\}$ and $p = 0.7$ and solve system (1), based on the 8×8-transition matrix

$$P = \begin{pmatrix} 0.1 & \cdots & 0.1 & 0.3 \\ 0.1 & \cdots & 0.1 & 0.3 \\ \vdots & \ddots & \vdots & 0.3 \\ 0.1 & \cdots & 0.1 & 0.3 \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

The solution vector for our example is $h = (h_{1A}, h_{2A}, \ldots, h_{8A}) = (0.4, 0.4, 0.4, 0.4, 1, 1, 0.4, 0)$. It is trivial to verify that $h_{i_0 A_0} = 0.4$ for *every* choice i_0, A_0 with $i_0 \notin A_0$. Equally trivial is the fact that $h_{i_0 A_0} = 1$

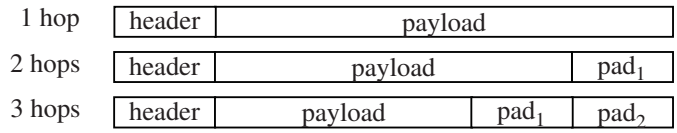

Figure 5: Effect of padding

if $i_0 \in A_0$ and $h_{i_0 A_0} = 0$ if i_0 is the receiver. Suppose that the initiator uses $N = 3 > k$ access points, then choosing $t = 9$ lowers the chance q for the attacker to get all shares that are needed to determine the final receiver down to $q = 0.064 < \frac{1}{n} = 0.125$.

6.2 Keeping a fixed packet size

Following the usual requirements for a Mix-net, demanding all messages to be of equal size, we need padding to avoid a partial discovery of the route due to shrinking packet sizes. Hence, in a network with n nodes, we have an overhead of $O(1)$ bytes for the routing information attached to the payload. At each hop, the forwarding address gets retracted and replaced by a suitable padding. This overhead applies to a share that is sent from routing jondo R_i to router R_{i+1} along the path chosen by the initiator. Regarding the shares that node R_i anonymously receives from the initiator through the crowd, we have $O(t) \cdot O(1)$ such shares of additional traffic for each router R_i with $i = 1, 2, \ldots, r$ where $r \in O(n)$. The overall overhead thus comes to $O(t \cdot n)$, and is therefore polynomial in the size of the network. According to standard security notions in the field of unconditional security (see [19]), our protocol is efficient, as its additional overhead is no more than polynomial in the size of the network. This method is advantageous, since the packet size is equal for all packets and regardless of the number of routing jondos.

The use of padding to keep the packet size fixed throughout the communication may still reveal information to the adversary via the so-far contained amount of padding. In fact, the number of routing jondos is, at any point, given by one plus the number of pads found in the packet (cf. Figure 5). Even worse, the constant padding can have an additional decreasing effect on the chosen threshold. Here, we briefly describe how to fix both issues.

If the routing jondo R_1 gets the required number of shares (as indicated by the *header* field), he finds out that he is the final receiver (in case that e.g. *null* is obtained) or he reconstructs the identity of the next routing jondo. Now, unlike the above version of the protocol, the routing jondo R_1 does not replace this field by an appropriate amount of padding, but sets it to a value pad_1 that he creates from all the information in his possession using a system wide known function f. This new packet is then forwarded to the next routing jondo R_2. This issue can be fixed immediately if the padding is derived from all shares the routing jondo receives.

Let R_i have received the shares $s_i^{(0)}$ (from his predecessor R_{i-1}) and $s_i^{(1)}, \ldots, s_i^{(t-1)}$ from the initiator along other anonymous paths. First, R_i reconstructs $(R_{i+1}, s_{i+1}^{(0)}) = s_i^{(0)} \oplus s_i^{(1)} \oplus \cdots \oplus s_i^{(t-1)}$. Then, he removes the address by replacing $(R_{i+1}, s_{i+1}^{(0)})$ with $(\text{pad}_i, s_{i+1}^{(0)})$. In the previous version of the protocol, this padding would have grown continuously, until its natural limit given by the total packet size (cf. Figure 5). Observe that we cannot just replace R_{i+1} by some constant padding, say *null*: suppose that we use $t = 2$ so that two shares make up the next address. If the address field is replaced by a zero-string, then $s_{i+1}^{(1)}$, transmitted from the initiator, will directly contain the next forwarding address. Still, if the threshold is t, i.e. t shares determine the next hop, then the usage of a constant padding always permits finding out the next address with only $t-1$ (rather than t) shares. We can fix this by making the padding functionally dependent on all the information in the routing jondo's possession.

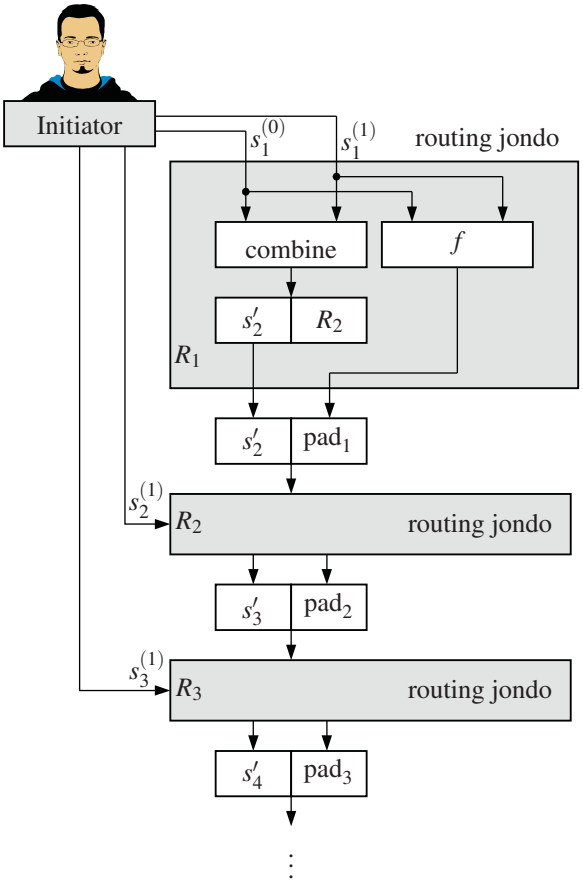

Figure 6: Transmission with pseudorandom padding

The pseudorandom padding $\text{pad}_i := f\left(s_i^{(0)}, s_i^{(1)}, \ldots, s_i^{(t-1)}\right)$ is calculated in such a way that an adversary missing any of the t shares will have entropy of at least $|R_{i+1}|$ bits whenever trying to recover pad_i. In other words, we want to make the padding in the packet from R_i to R_{i+1} looks random (to the adversary), unless the attacker gets all the information that R_i knows or R_{i+1} is about to receive. This ensures that the threshold always remains t (and never decreases down to $t-1$).

Figure 6 illustrates the idea for $t = 2$, i.e. two shares required for determining the next hop. Observe that in a slight abuse of notation for the sake of clarity, the routing jondo R_1 in Figure 6 reconstructs the value R_2 to determine the next hop (in reality, R_1 would get the address R_2 of the next hop R_2).

A simple way to calculate the padding is as follows: assume that addresses are of constant size n, and pick a prime number p having n bit. Define the padding function as

$$f(x_0, \ldots, x_{t-1}) := \left(\sum_{i=0}^{t-1} x_i\right) \bmod p.$$

If any of the x_i (with $i \in \{0, 1, \ldots, t-1\}$ is missing, then the value $f(x_0, \ldots, x_{t-1})$ remains as uncertain as $x_i \bmod p$, because any value is equally possible. It goes without saying that this is not the only possible way of calculating the padding, and any function whose dependence on all input parameters is equally strong in terms of residual entropy upon a missing input is permissible.

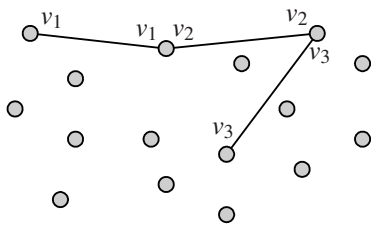

Figure 7: A Crowds-Channel and its virtual channel numbers

7 Bidirectional Anonymous Communication

If this protocol is used to transfer data, the overhead in communication would obviously be enormous. Hence, it appears reasonable to keep the channel open once it has been established.

Fortunately, this virtual channel can be built thanks to the ability of the Crowds system for bidirectional communication. In the Crowds system, a virtual channel is built by assigning (different) virtual channel numbers for each connection between two jondos on the path during the connection phase. If data is sent, the appropriate channel number for the first jondo is chosen by the initiator and stored in the data packet. The first jondo now knows to which jondo he has to forward this message and which channel number he has to use.

Using different channel numbers offers the following advantage: if an attacker gets two different packets, he cannot tell if both belong to the same communication (but belong to the same receiver). If the receiver wants to send data backwards to the original sender (initiator), he only has to use the channel number which is common for him and his predecessor. Along the channel (path), every jondo has to store a relation (in this case a function), which maps the predecessor and its virtual channel number on another jondo and another channel number.

Building up a virtual channel in the Crowds-Mix-system is similar to the Crowds system (see Figure 8). At first, a virtual connection between the initiator R_0 and the first routing jondo R_1 is build up using the Crowds system.

In this example, two Crowds-connections between R_0 and R_1 are set up, where only one connection (the dotted line) remains (see Figure 8a). The other connection is closed after sending the share. In the next step, a virtual Crowds-connection is established between R_1 and R_2 (see Figure 8b). The connection between I and R_2 is closed after sending the share. This is repeated for R_3 and R_4 (cf. Figures 8c and 8d), which yields to the virtual path shown in Figure 9. Figure 9a shows all the Crowds-connections and Figure 9b shows the resulting virtual Crowds-Mix-connection.

This system can easily be modified to use randomly chosen Crowds-connections to create the crowds-mix channel. In this case, the initiator decides (using a designated field within the payload) which connection a routing jondo should use. Alternatively, one can also keep all connections open. In this case, every routing jondo can use every connection to send data (foreward or backward) to make volume attacks on intermediate nodes more difficult.

8 Discussions

In terms of overhead through padding, our protocol is efficient as it adds only a polynomial (in terms of the network size) amount of data to each packet. Furthermore, due to virtual channels being kept open, most of the overhead is needed during the phase when the channel is set up. The transmission itself is not costly once the channel is there. Until then, the overhead is mostly due to the multipath transmission and the Crowds system. We take a closer look at both now.

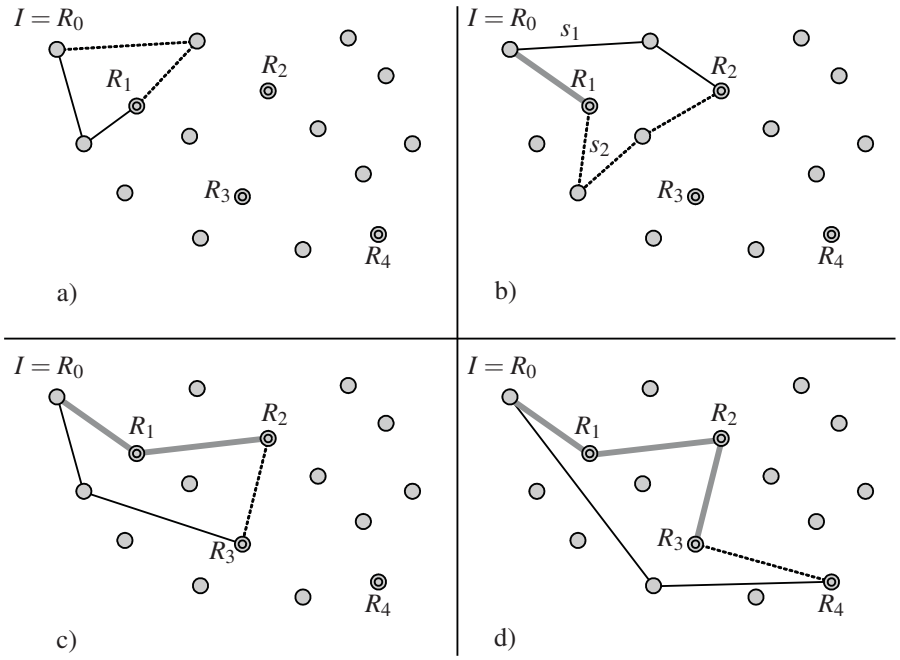

Figure 8: Building up a channel in the Crowds-Mix-system

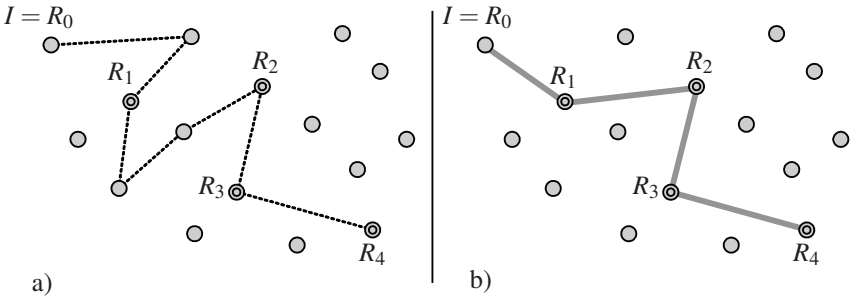

Figure 9: The resulting Crowds-Mix-channel

Performance: Since our construction employs the Crowds system for packet delivery, the delay of the transmission is strongly dependent on the transmission protocol underlying the crowd itself (e.g. TCP/IP, if the internet is the platform for the Crowds-overlay network, but others are equally possible). The average path length of a (single) Crowd-transmission is dependent on the parameter p_f. Obviously, each forwarding is a Bernoulli-trial, succeeding with chance $1 - p_f$ (cf. the transition matrix P of the Markov-chain model), so that the number of hops (equal to the length of the actual path), has the expected value $\frac{1}{1-p_f}$. With t shares emitted by the initiator per hop, and $r-1$ (intermediate) relay nodes between the initiator and the receiver (cf. Figure 2, in which $r = 4, t = 2$), we get $\frac{r \cdot t}{1-p_f}$ for the total number of packet transfers. The expected temporal delay is then by this factor proportional to the delay of the lower-level transmission protocols. Simulations and experimental evaluation (over the internet) are subjects of future work.

System	Sender Anonymity	Receiver Anonymity
Crowds [13]	probable innocence	Pr[absolute privacy] $\to 1$ for $n \to \infty$
Our system	probable innocence	Pr[absolute privacy] $\to 1$ for $t \to \infty$ and $n < \infty$

Table 2: Security comparison

Recovery from node failure: If one of the Crowds-connections between R_i and R_{i+1} goes down, then R_i can re-build the Crowds-channel upon noticing this incident and no action from the initiator is needed. Only if $i = 0$, then the initiator has to build a new Crowds-Connection to R_1. Due to this decentral mechanism, the predecessor attack has a worse chance to succeed. Unfortunately, however, it appears impossible for the network to inherently (and by itself) recover from a lost routing jondo during a transmission. Suppose that R_i breaks down, then R_{i-1} and R_{i+1} would notice this and could try bridging the new gap by themselves. However, this would require R_{i-1} to *know* that he is supposed to forward his packets to R_{i+1} (and vice versa), thus contradicting the anonymity of the overall protocol. Therefore, in case of failure of a routing jondo along the path, the initiator has to re-establish the channel from scratch.

Extensions and Open Problems: Our construction assumes a completely connected network of jondos. Under this assumption and perfectly protected channels, the system is unconditionally secure. Suitable channels can be created in several ways, among them is quantum key distribution (e.g. [20]) or multipath transmission (e.g. [21]). If either one turns out infeasible, then standard public-key cryptography can create a virtually complete graph, with security boiling down to the computational security of the underlying public-key encryption. Our scheme can be generalized in various ways. Achieving security against active adversaries amounts to replacing the proposed (t,t)-secret sharing by a general (t,t')-secret sharing with $t < t'$. It is known [22] that any such scheme permits recovery from at most $\lfloor (t'-t)/2 \rfloor$ modified shares, and hence is secure against active adversaries. The model can be amended appropriately to give the likelihood of no more than $\lfloor (t'-t)/2 \rfloor$ shares traversing the compromised set A. Much less trivial problems arise when the interconnection graph of the jondos is arbitrary and not complete. In that case, we run into the problem of having to consider each k-sized subset $A \subset \{1,2,\ldots,n\}$ separately. Despite the maths remaining simple, the number of such sets can become impractically large. A final generalization may be towards security against external adversaries eavesdropping on the entire traffic in the whole network. The Crowds system is known to be insecure against such adversaries, and fixing this is yet another subject of future research.

9 Conclusions

We extended the Crowds system for anonymous communication from hiding the initiator's identity only to both, sender- and receiver-anonymity. Moreover, even if the receiver does not know from where his incoming queries come from, he can nevertheless respond to them easily, as the protocol can provide a bidirectional anonymous channel. Inspired by ideas used in onion routing, we have shown how multipath transmission and secret sharing can be used to combine Crowds with the Mix-net system to get the best from two worlds (without unifying the disadvantages). Unlike past solutions, our proposal does neither necessarily hinge on the usual computational intractability claims, nor provides security only asymptotically, i.e. only for infinitely large networks; cf. Table 2. Our solution comes at low (in fact only polynomial) additional cost in terms of network overhead, and is thus efficient according to standard definitions [19]. We explicitly assumed a fully connected network, so our protocol is practical only in overlay (logical) networks. If the physical network is loosely connected (e.g. a scale-free graph), then virtual channels, protected by standard cryptographic means, are required. The overall security of the

anonymous communication is in any case determined by the security of the underlying overlay network channels, as our solution puts an unconditionally secure protocol layer on top of this. Besides this and nevertheless, the strength of the anonymity in our approach is fully controllable by the sender. Hence, having a reasonable estimate on the expected power of the adversary in the network, the sender can freely choose the number of shares to use for building the channel and so prevent the adversary from discovering who is talking to whom. Given secure channels, anonymity cannot be breached by a passive threshold adversary. Regarding active attacks, the sender is only required to react on this if one of his chosen routing jondos goes down, in which case a full re-establishment of the channel from scratch is required. Besides this, the protocol is robust against node failure, without the need for the sender or receiver to become active to fix this.

References

[1] S. Rass, P. Schartner, and R. Wigoutschnigg, "Crowds based on secret-sharing," in *Proc. of the 6th International Conference on Availability, Reliability and Security (ARES'11), Vienna, Austria*. IEEE, August 2011, pp. 359–364.

[2] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, February 1981.

[3] D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–75, March 1988.

[4] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proc. of the 13th Conference on USENIX Security Symposium – Volume 13 (SSYM'04), San Diego, California, USA*. USENIX Association, 2004, pp. 303–320.

[5] E. Hughes, "A cypherpunk's manifesto," 1993, http://www.activism.net/cypherpunk/manifesto.html.

[6] L. Sassaman, U. Möller, C. Tuckley, E. Arneson, A. Kirk, P. Palfrader, and L. M. Cottrell, "Mixmaster," 2008, http://mixmaster.sourceforge.net/.

[7] C. Gülcü and G. Tsudik, "Mixing email with babel," in *Proc. of the 1996 Symposium on Network and Distributed System Security (SNDSS'96)*. IEEE, 1996, pp. 2–16.

[8] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *Proc. of the 2003 IEEE Symposium on Security and Privacy (SP'03), Berkeley, California, USA*. IEEE, May 2003, pp. 2–15.

[9] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Proc. of the 1st International Workshop on Information Hiding (IH'96), Hainan Island, LNCS*, vol. 1174. Springer-Verlag, June 1996, pp. 137–150.

[10] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 482 –494, May 1998.

[11] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[12] R. Dingledine, N. Mathewson, and P. Syverson, "Challenges in deploying low-latency anonymity," NRL CHACS, Tech. Rep. Report 5540-625, 2005.

[13] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, November 1998.

[14] S. Goel, M. Robson, M. Polte, and E. Gun Sirer, "Herbivore: A Scalable and Efficient Protocol for Anonymous Communication," Cornell University, Ithaca, NY, Tech. Rep. 2003-1890, 2003.

[15] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "P5: a protocol for scalable anonymous communication," *Journal of Computer Security*, vol. 13, no. 6, pp. 839–876, December 2005.

[16] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization:anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, 2010.

[17] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[18] D. Stirzaker, *Stochastic Processes & Models*. Oxford University Press, 2005.

[19] Y. Desmedt and Y. Wang, "Perfectly secure message transmission revisited," in *Proc. of the 21st Annual Eurocrypt Conference (Eurocrypt'02), Amsterdam, The Netherlands, LNCS*, vol. 2332. Springer-Verlag, April-May 2002, pp. 502–507.

[20] C. Elliott, "The DARPA quantum network," *eprint arXiv:quant-ph/0412029*, December 2004.

[21] M. Fitzi, M. Franklin, J. Garay, and S. H. Vardhan, "Towards optimal and efficient perfectly secure message transmission," in *Proc. of the 4th Conference on Theory of Cryptography (TCC'07), Amsterdam, The Netherlands, LNCS*, vol. 4392. Springer-Verlag, February 2007, pp. 311–322.

[22] R. McElice and D. Sarwate, "On sharing secrets and Reed-Solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, 1981.

Stefan Rass has a master degree in mathematics (with a focus on statistics), as well as in computer science from the Klagenfurt University in 2005, and gained a PhD degree in mathematics in 2009 (with a focus on information-theoretic security). His research interests include general system security, as well as complexity theory and decision theory. He participated in the EU-project SECOQC in the field of Quantum Cryptography and is currently a member of the system security research group at Klagenfurt University.

Raphael Wigoutschnigg has a master degree in computer science (with a focus on information security) from the Klagenfurt University in 2007. His research interests include anonymous communication systems, secret sharing techniques and digital forensics. He is currently a member of the system security research group at Klagenfurt University.

Peter Schartner received his master degree in Telematics (with a focus on information security) from the Technical University of Graz in 1997 and his PhD in computer science (with a focus on security tokens) from Klagenfurt University in 2001. His research interests include key management, security infrastructures and applications for security tokens, especially smartcards and personal digital assistants. He is currently assistant professor in the system security research group at Klagenfurt University.