A Mix-Mediated Anonymity Service and Its Payment*

Elke Franz and Anja Jerichow

Dresden University of Technology
Department of Theoretical Computer Science
D-01062 Dresden

{ef1,jerichow}@inf.tu-dresden.de

Abstract. One measure to provide anonymity for the users of a communication network are mixes whose usage was proposed in several applications recently. However, in practice such concepts are not widely used. One reason may be that the payment for providers, who commercially offer such mix-mediated anonymity service, has not been considered yet. We present detailed protocols for payment schemes that allow anonymous, secure payment of a mix-mediated anonymity service. The schemes aim to achieve confidentiality and integrity for all the user, the provider and the bank.

1 Introduction

In the last decade, the discussion about anonymity and privacy has become more and more important. One requirement that should be supported in secure communication systems is anonymous communication, i.e. to keep secret who is in contact with whom at which time and, maybe, from which location.

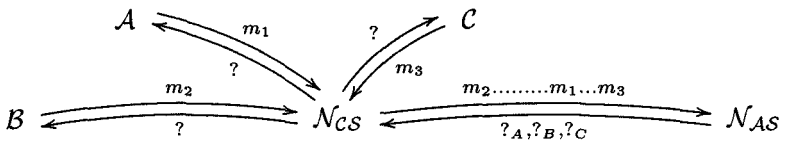
The subject matter of this paper is "mixes" - one anonymity concept. Firstly introduced by Chaum in 1981 for anonymous e-mail transfer [1], the usage of mixes was proposed in several other applications recently. [2], [3], [4], [5], [6], and [7], for example, suggest their usage for providing anonymity in telecommunication networks, mobile communication systems and the internet.

By mixes, sender and recipient anonymity as well as the protection of the communication relation can be achieved. That means, sender and recipient of a message cannot be correlated. Though the mix concept allows that lines are tapped; and all but one of the mixes passed may be corrupted. A mix is called "corrupted" if it tells an attacker the correlation of an incoming and the corresponding forwarded message.

We distinguish between two networks that must be operated independently: one providing the communication service $\mathcal{N}_{\mathcal{CS}}$ and the other providing the anonymity service $\mathcal{N}_{\mathcal{AS}}$. To achieve anonymity, we extend $\mathcal{N}_{\mathcal{CS}}$ by $\mathcal{N}_{\mathcal{AS}}$. Thus, if

* Parts of this work were supported by the German Science Foundation (DFG) and the Gottlieb Daimler- and Karl Benz-Foundation.

Alice ($\mathcal{A}$) wants to communicate with Bob ($\mathcal{B}$), she includes routing information for the usage of $\mathcal{N}_{\mathcal{AS}}$ in her message m_1 and sends it to $\mathcal{N}_{\mathcal{CS}}$, which then forwards it among others to $\mathcal{N}_{\mathcal{AS}}$. After processing the message, $\mathcal{N}_{\mathcal{AS}}$ sends the resulting message back to $\mathcal{N}_{\mathcal{CS}}$ together with other messages. Thus, the provider of $\mathcal{N}_{\mathcal{CS}}$ cannot correlate $\mathcal{A}$ and $\mathcal{B}$ (assuming there is enough traffic in the anonymity network). Moreover, if the attacking model of mixes holds, i.e. all lines can be tapped but at least one mix is trustworthy, then even the mix providers of $\mathcal{N}_{\mathcal{AS}}$ are not able to correlate anything.



For both networks, payment has to be supplied. The payment of communication services is generally supported nowadays. In this paper, we focus therefore on the additional needed payment for the mix-mediated anonymity network that must be operated by different providers as of the communication network. To solve this problem for mixes is even more tricky, since one important requirement on mixes is their independent construction as well as their availability and operation by independent providers. Thus, this paper give answers to the following question:

> How can a user pay for the usage of an anonymity service without revealing his anonymity against all (including the providers of the mixes in $\mathcal{N}_{\mathcal{AS}}$)?

First ideas to achieve payment of anonymous communication were discussed in [8]. We revise some of these and present further results.

Section 2 continues the short overview about the mix network and discusses suitable payment schemes for this anonymity service. Section 3 describes an interactive micro-payment scheme that applies "tick payment" to transfer small amounts of payment. Several payment protocols are described in general. The two most efficient approaches will be applied to the mix concept. Section 4 shows the protocols in detail. Some concerns on attacks and efficiency are discussed in Section 5. The protocols between users, bank and providers to allow secure, anonymous payment are outlined in Section 6. Finally, Section 7 concludes and gives an outlook.

2 Suitable Payment Schemes for a Mix-Mediated Anonymity Service

2.1 Requirements on Mixes

A mix is a network node with cryptographic facilities that hides the relations between communicating users. Mixes can be linked to mix chains. Due to the functionality of a mix, an attacker is not able to trace messages through the mix

network: The mix changes the appearance of the messages by using a suitable cryptosystem. Correlation by the message length is avoided if a length-preserving scheme is used for encryption. Time correlation is avoided as the mix collects all messages in its buffer and re-orders them before they are forwarded. This way, incoming and forwarded messages of a mix are not linkable.

For the purpose of this paper we assume the existence of $\mathcal{N}_{\mathcal{CS}}$. For the anonymity network $\mathcal{N}_{\mathcal{AS}}$ the following assumptions are made: The mixes $\mathcal{M}_i$ of a mix chain $\mathcal{MC}_k$, that a message passes, have independent providers $\mathcal{P}_{\mathcal{M}_i}$. That means

$$\forall k \bullet \exists i, j \bullet (\mathcal{M}_i, \mathcal{M}_j \in \mathcal{MC}_k) \Rightarrow (\mathcal{P}_{\mathcal{M}_i} \neq \mathcal{P}_{\mathcal{M}_j}) \tag{1}$$

whereby a mix chain $\mathcal{MC}$ comprises m mixes with $\bigoplus$ being the concatenation of all used mixes.

$$\forall k \bullet \mathcal{MC}_k \in \mathcal{N}_{\mathcal{AS}} \wedge \mathcal{MC}_k = \bigoplus_{i=1}^{m} \mathcal{M}_i \tag{2}$$

A message consists of several message blocks j. For sending it through the mix chain each message block N_{1j} has to be prepared according to Formula 3.

$$\begin{array}{lll} N_{(m+1)j} := & c_{m+1}(N_j) & \\ N_{ij} := & c_i(r_{ij}, A_{i+1}, N_{(i+1)j}) & (i = 1, \ldots, m) \end{array} \tag{3}$$

where N_j is the message block j for the recipient whose address is A_{m+1}, and N_{ij} is the message block j for mix $\mathcal{M}_i$ with address A_i that is encrypted with the mix' public key c_i.

One can visualize this encryption process as follows: If Alice wants to send a postcard to Bob, she encloses it in an envelope such that only Bob is able to read the content. Additionally she encloses this envelope successively into further envelopes, whereby everyone carries the address of another post office, i.e. a mix address. Like that none of the post offices is in the position to link Alice and Bob.

Let us come back to the subject. Each covering, which is put around the original message must contain random bits r_{ij}. Otherwise an eavesdropper could easily correlate messages as a mix works deterministic. He needs to encrypt only the output with the mix' public key and, following, to compare the result with all messages that arrived before at this mix.

Note: We suggest the usage of a length-preserving scheme. That means all blocks have the same size. More details on this scheme can be found in [1]. Additionally, [9] and [10] give a good introduction to mixes.

2.2 Discussion of Suitable Payment Schemes

Each mix provider must be paid for passing messages. There are two ways: (a) A mix provider is directly paid. (b) Only one mix provider of the mix chain is paid. In the latter, there must exist a general agreement (like for international

telephone calls). For example the money received could be distributed among all mixes according to the number of messages each mix has processed.

There are several ways of payment: a message can be pre-paid, post-paid or paid immediately (pay-now). Note: In the following section, we focus only on the protocols between user and mix provider. The charging process, which involves the bank, will be discussed in Section 6.

The problem with these schemes is that either the user, who paid in advance, has to trust that the mix will really process his message, or the mix, that already processed the message, has to trust that the user will subsequently pay for the service. And last but not least, if an anonymous pay-now system for on-line payment is used, then real-time communication cannot be guaranteed at all (due to the calculation time for on-line verifiction of the payment).

Since we want to keep the advantage of one party over the other as small as possible, we propose an *interactive micro-payment scheme* that will be discussed in the following section.

3 The Interactive Micro-payment Scheme

3.1 The Basic Idea

Instead of paying the full service

- a message is split in small blocks, and
- each block is paid when processed.

By "paying when processed" we understand that the merchant receives the appropriate amount and immediately performs the service (to allow real-time communication). Section 6 will describe why the merchant can trust that encashment of the received money is possible later. Using this scheme, neither merchant nor customer is much ahead of the other party.

The scheme is a *micro-payment* scheme since only small amounts of money, in the following called ticks, will be transferred. It is *interactive* because payment will be included in the actual message and dropped at the mix according to the payment strategy. The interactive micro-payment scheme combines two approaches: the concept of mixes and tick payment.

3.2 Tick Payment

Originally, ticks were used for authentication [11]. In 1995, Pederson investigated tick payment [12] as an efficient way of payment. We apply this principle to the concept of mixes. Since a tick means that only some additional bits are added to the message block, it want be much of a burden to the performance of mixes.

The principle of tick payment is based on a one-way function. This one-way function is a length-preserving permutation f but, nevertheless, called hash function in the following. At the beginning a random number a is chosen. The hash function is applied n times to a. These results in hash values

$$f^1 = f(a), \quad f^2 = f(f^1), \quad \ldots, \quad f^n = f(f^{n-1}) = f(f(f(...f(a)))) \qquad (4)$$

where each hash value f^{n-i} is named a tick t_i (with $i = 0, \ldots, n-1$).

Ticks can be easily used for paying in small amounts. For example between a customer, i.e. a user, and a merchant, i.e. a mix, to pay for anonymous transferred information.

3.3 Generation of a Cheque

The user creates a digital cheque that includes f^n generated according to Formula 4. Other parameters a cheque should contain are a cheque identifier to recognize the ticks that belong to this cheque, the amount-per-tick value, the address of the user's bank for encashment of the cheque, the recipient of the cheque (i.e. the mix' address), and the signature of the user.

This cheque is transferred to and stored at a mix that was chosen by the user. Now, the user sends a tick with each message block. Note, at most n message blocks can be sent and paid.

user		*mix*
`generate` cheque	send cheque(f^n) →	`store` f^n
sends first block	send m_1,t_1 →	`if` $f(t_1) = f^n$ `then` `forward` m_1 `and` `store` $f^{n-1}(= t_1)$

As a general rule, the mix stores a hash value f^{n-j+1}. If it receives the next tick t_j, it applies the publicly known hash function f and compares this result with the stored hash value. If they are equal, the mix has received the correct payment.

sends following blocks	send m_j,t_j →	`if` $f(t_j) = f^{n-j+1}$ `then` `forward` m_j `and` `store` $f^{n-j}(= t_j)$

The mix provider can encash the cheque whenever it is wished. He only has to send the cheque and the last tick received to the bank. Since the bank has supplied the money for generating the value of this cheque, encashment will be no problem (see Section 6 for details on that).

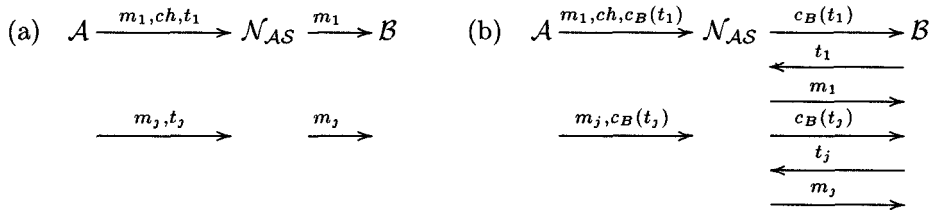
3.4 Different Approaches

There are two aspects to consider:

1. When does the mix provider receive payment, i.e. before or after performing his service (the delivery of a message block)?
2. When is the tick sent, i.e. has the sender or the recipient triggered the payment?

Delivery after Payment. If the sender triggers the payment (P1.a), she includes the digital cheque *ch* and the first tick t_1 in the first block of the message.

Protocol P1:



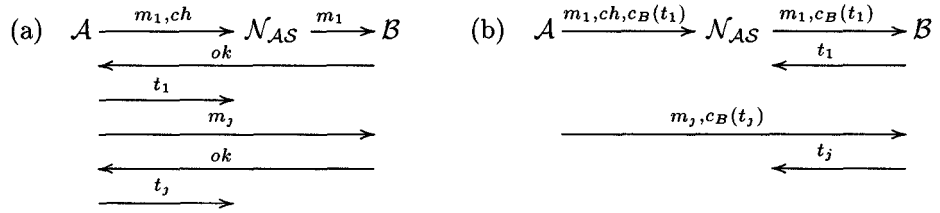
The mix network stores the cheque and forwards the first message block m_1. The following blocks include only the appropriate tick for each message block m_j. Thus the anonymity service is immediately paid for, further steps are not necessary.

If the recipient triggers the payment (P1.b), the first block of the message includes the cheque and the first tick. All ticks must be encrypted with the public key c_B of the recipient such that the mix is not able to get the payment before the recipient has sent back a decrypted tick. The mix stores the cheque and keeps the message block. Then it sends the encrypted tick to the recipient of the message to indicate that a message will be sent after receiving the decrypted tick. Hence, the recipient decrypts the tick and sends it back to the mix network. After receiving this tick, the mix will send the message block.

The following blocks are handled in exactly the same way. However, if the recipient triggers the payment, two additional steps for the transfer of each message block are needed.

Payment after Delivery. If the sender triggers the payment (P2.a), she sends the first message block together with the cheque. The mix network stores the cheque and sends the message block to the recipient. After that the recipient sends an *ok* to the sender of the message. Hence, the sender can be sure that her message block has arrived, i.e. she has a guarantee of message delivery. She then sends the tick to the mix network. For the following blocks, payment is triggered by the sender in the same way.

Protocol P2:



To improve efficiency, t_j could always be sent together with the following message block m_{j+1}. Nevertheless, using this approach the delay by the mix network slows down the data rate.

If the recipient triggers the payment (P2.b), the ticks must be encrypted as well. The cheque is included in the first message block. After receiving a block, the recipient sends back the decrypted tick to the mix network. In practice this happens by encrypting it again. The sender does not get an acknowledgement whether the recipient has got the message blocks. But the mixes will be interested in sending the messages to the recipient to get their payment. This protocol is more efficient than (P2a).

3.5 Remarks

In Section 3.4 we pointed out which measure is a burden to the performance of the system. In summary, the following should be preferred: If the main goal is to achieve efficiency, one should use the protocol for "delivery after payment" where the sender triggers the payment. If one wants to assure that the addressee has really received his message, then the protocol for payment after delivery should be considered. This protocol will still allow an efficient implementation when the recipient triggers the payment.

In Section 2.2, we mentioned two ways of payment. We will now discuss which is applicable for the preferred payment schemes. In case (a) of Sect.2.2, if each provider is directly paid, a digital cheque must be included for each mix of the chain. The problem with that: If a mix continuously receives ticks, it can correlate these ticks and identify the communication relation. Thus, this protocol should only be applied for special services where all users of a mix chain simultaneously send messages (and ticks). I.e. they behave in the same manner and, therefore, belong to the same anonymity group. But this cannot be demanded for all kinds of services. One example where this can be applied are connection-oriented services. For example communication with Real-Time Mixes [13] via ISDN: All users of a local exchange belong to one anonymity group and synchronously establish channels for data transfer (sending dummy traffic if no real data are to be transmitted).

However, if all participants have to behave in the same manner, then more efficient measures for payment can be applied such as a flat-rate scheme. Thus, we will focus on case (b) of Sect.2.2 where the problem described above is not really a problem as long as the mix that receives the cheque is either the first or the last mix of the chain. This is due to the fact that, according to the mix scheme, these mixes may know sender or recipient anyway.

In the following section we will describe detailed protocols for "delivery after payment" with the first mix receiving the cheque and "payment after delivery" with the last mix receiving the cheque.

4 Detailed Protocols

4.1 Delivery after Payment

For protocols with delivery after payment, the first mix is used for accepting the payment that the sender will trigger. Since it is not necessary to pass the cheque

and the ticks through the whole mix network, the extending of the message length can be limited: Only the message N_{1j} prepared for the first mix must be extended for each message block j. After passing this mix, the additional data are removed. The sender generates her message as follows:

first message block

$$\begin{array}{lcl} N_{(m+1)1} & := & c_{m+1}(N_1) \\ N_{i1} & := & c_i(r_{i1}, A_{i+1}, N_{(i+1)1}) \\ N_{11} & := & c_1(r_{11}, A_2, N_{21}, ch_A, t_1) \end{array} \tag{5}$$

following message blocks

$$\begin{array}{lcl} N_{(m+1)j} & := & c_{m+1}(N_j) \\ N_{ij} & := & c_i(r_{ij}, A_{i+1}, N_{(i+1)j}) \\ N_{1j} & := & c_i(r_{1j}, A_2, N_{2j}, id_A, t_j)) \end{array} \tag{6}$$

She then sends the first block N_{11} it to the first mix according to Formula 5.

$$A \xrightarrow{c_1(r_{11}, A_{M_2}, N_{21}, ch_A, t_1)} M_1 \xrightarrow{N_{21}} \ldots \xrightarrow{N_{(m+1)1}} B$$

The cheque ch_A contains an identifier id_A as well as additional parameters (compare with Section 3.3). The mix stores cheque and tick and, at the same time, performs its service, i.e. forwards N_{21}. Thus, t_1 is the payment for the first block. According to Formula 6, every following message block also contains a tick. Moreover, it contains the identifier id_A to assign t_j to the appropriate cheque.

$$A \xrightarrow{c_1(r_{1j}, A_{M_2}, N_{2j}, id_A, t_j)} M_1 \xrightarrow{N_{2j}} \ldots \xrightarrow{N_{(m+1)j}} B$$

The mix can check the correctness of the tick as described in Section 3.2 while passing the processed message to the next address. Thus, the mix can be sure that it will get the payment for its service.

Note: Another possibility to assign a tick to the cheque is that the mix could calculate the predecessor of every received tick by applying the hash function to it. Then it could search for the appropriate cheque according to the calculated value (compare 3.3).

4.2 Payment after Delivery

As motivated in Section 3.5, to improve the efficiency the recipient should trigger the payment for payment-after-delivery protocols and, therefore, the last mix is used for handling the payment.

The sender generates his message as follows:

$$\begin{array}{lll} N_{(m+1)1} & := & c_{m+1}(N_1,\, t_1) \\ N_{m1} & := & c_m(r_{m1}, A_{m+1}, N_{(m+1)1},\, ch_A) \\ N_{i1} & := & c_i(r_{i1}, A_{i+1}, N_{(i+1)1}) \\ & & \\ N_{(m+1)j} & := & c_{m+1}(N_j,\, id_A,\, t_j) \\ N_{mj} & := & c_i(r_{mj}, A_{m+1}, N_{(m+1)j}) \\ N_{ij} & := & c_i(r_{ij}, A_{i+1}, N_{(i+1)j}) \end{array} \tag{7}$$

Though the sender does not get an acknowledge message from the recipient, the guarantee of message receipt is indirectly achieved: The mix network will send the message blocks to the recipient to get the payment for its service in exchange. Here, the user is ahead of the mix chain provider.

$$A \xrightarrow{N_{11}} \ldots \; M_{m-1} \xrightarrow{c_m(r_{m1}, A_{m+1}, N_{(m+1)1},\, ch_A)} M_m \xrightarrow{c_{m+1}(N_1,\, t_1)} B$$

Nevertheless, this advantage is kept small: The next message block is not sent before the mix has received and successfully checked the tick for the previous message block. Therefore, the provider can only be defrauded of one tick.

$$A \xrightarrow{N_{1j}} \ldots \; M_{m-1} \xrightarrow{c_m(r_{mj}, A_{m+1}, N_{(m+1)j})} M_m \underset{c_m(id_A, t_j)}{\overset{c_{m+1}(N_j,\, id_A,\, t_j)}{\rightleftarrows}} B$$

Note: In this protocol, we use identifiers such that the mix knows what cheques the ticks are referring to. The recipient must encrypt tick and identifier, too, when triggering the payment. Otherwise, an eavesdropper could easily link the payment data.

5 Discussion on Attacks and Efficiency

5.1 Discussion of Possible Attacks

This section discusses the possibilities of attacks on anonymity. There are two questions we want to answer:

- Are there possibilities for attacks by others to prevent either the correct payment or the service?
- Are there possibilities for attacks by one of the participating parties to defraud the other party?

To the first concern, the mix could deny its service. Denial-of-service attacks are a problem in general. For example an attacker could try to jam the protocol by intercepting either messages or ticks. To recognize this, additional measures are needed such that the users can check the correct work of the mixes and stop the transmission of ticks if necessary. For example, mixes could make a commitment of the processing of all messages arriving in future, which then is

checkable by the users (see [14] for a detailed discussion on the approach to commitment schemes).

To keep the anonymity of the users, the cheques only include pseudonyms instead of real user information. A pseudonym is a unique name and cannot be connected to the real user identity. But there is one remaining threat for the users of an anonymity service where cheque payment is applied: The cheques are used for many times. So at least the providers are able to collect information about the users even though only under pseudonyms. However, with additional information beyond the anonymity system, the providers could be able to derive information about the real users. We minimized this problem allowing payment transactions only by the first or the last mix of a chain. Here are two more possibilities for avoiding this problem.

At first, the cheques could include not too large amounts. Thus the mixes can only collect information about the limited number of user transactions. It is the responsibility of the users to limit this information.

The other possibility is that a person uses several cheques. A mix provider does then not know what cheques, represented by different pseudonyms, belong to the same person. Thus, linking users and the information yielded by the cheques becomes more difficult. However, this variant is limited by the capability of the mixes: If the mix system is to serve a large number of users, the number of possible cheques for each user is inversely proportional.

Last but not least, charging protocols are needed to assure a correct payment. The usage of tamper proof hardware is the approach that we prefer. Section 6 describes in detail how the forging of cheques can be prevented by this.

The discussed problems show that the only usage of anonymous payment protocols is not sufficient to allow anonymous payment. We must also consider the boundary conditions of the system that provides anonymity.

5.2 Extention of the Message Length by Payment Data

Depending on who triggers the payment, payment data are included in either the outer- or the innermost shell of a message block (compare with Formulae 5, 6 and 7 in Section 4). The traffic load is not really extended if length-preserving schemes are used. Instead, payment data shorten the available space for real user data. Due to this fact an additional message block may be needed. Since a tick is just a small addition to the protocol, this increasing of the traffic load may be negligible.

5.3 Using Hybrid Encrypting Schemes to Improve the Efficiency

The introduced protocols are based on asymmetric encryption schemes. For improving the efficiency, the usage of hybrid schemes is to be preferred.

Such mix schemes that are also length preserving schemes use symmetric encryption for the main part of the message, which allows much faster en- and decryption. The message is split into the real message and a header that includes address information. The mix handles these two parts separately.

The user has to prepare the address part first. Afterwards he encrypts the message with the generated secret keys. I.e. the first block for each mix is encrypted with its public key. It contains a secret key k_{ij} for a symmetric cryptosystem that is then used to encrypt the remaining part of the message. Thus, only one block must be encrypted or decrypted with an asymmetric cryptosystem. All other blocks, the remaining part of the header and the real message are then processed using the symmetric cryptosystem.

Formula 8 shows the generation of the address part.

$$\begin{aligned} R_{(m+1)j} &:= [e_j] \\ R_{ij} &:= [c_i(k_{ij}, A_{i+1}), k_{ij}(R_{(i+1)j})] \quad (i = m, \ldots, 1) \end{aligned} \tag{8}$$

In summary, hybrid cryptosystems improve the efficiency of our protocols.

6 Charging: Protocols between Users, Network and Bank

6.1 General

Former sections discuss the protocols between users and service providers. They are necessary to transmit the payment data to the providers. In this section we also consider the protocol between provider and bank for the encashment of cheques according to [15].

In general, there are two possible protocols for encashment at the bank: on-line and off-line. If an on-line protocol is used, the mix does not perform its service, i.e. the transmission of the appropriate message, before it has received the credit entry from the bank. Thus, it can be sure that it really will get the payment from the users. Also, double-spending detection is possible that way. But there is an important disadvantage: The encashment request of the mix for all message blocks slows down the message transfer. This is for real-time communication not applicable.

The other possibility is to use an off-line protocol. The mix immediately performs its service, i.e. it sends the message to the next mix node or the recipient. Afterwards it sends an encashment request to the bank. Thus, this protocol does not slow down the message transfer. That is why we have chosen this variant for payment in mix-mediated anonymity communications.

But another problem occurs: The mix cannot be sure whether it really will get the payment for its service. To avoid this, we investigated the usage of tamper proof hardware, which will be discussed in the following section. This device also prevents double spending. Therefore, an on-line double spending detection is unnecessary.

6.2 Using a Tamper Proof Hardware Device

To generate the necessary cheques for the providers, every user owns a tamper proof hardware device (TPH). For details about building and using a TPH we refer to [16]. The task of the TPH is to achieve trustworthiness for all in the payment process, i.e. the users, the bank and the providers.

That means:

- for the bank:
  - It is not possible that users overdraw their accounts while creating cheques.
  - The bank is able to recognize cheques that one of its customers created with a TPH. Only these cheques will be accepted.
- for the users:
  - The billing on their accounts will be correctly performed.
  - The made out cheques are valid.
  - The user anonymity is kept against both the provider and the bank.
- for the providers:
  - They can recognize whether the cheques are valid.
  - They know that cheques, generated with a TPH, will be accepted by the bank and that they will get the correct amount for them.
  - The cheques are not usable by others.

The requirements on the TPH can be derived from these points:

- The tamper proof and the correct functionality of the TPH must be guaranteed.
- To keep user anonymity, the identity of the users must not be recognizable from the cheques (untraceability). Moreover, the non-correlation of different cheques must be guaranteed.
- To avoid the usage of forged cheques, there must be ways to decide whether a cheque was created with a TPH. This must be possible for both the bank and the provider.
- The bank must be able to test the correct identity of the provider who wants to cash a cheque.

6.3 Sketching the Charging Process

Every user owns two accounts. One is the real user account at the bank, the other one is an account that is managed by the TPH. The former is designated as bank account, the latter as TPH account. The TPH account is used to make out cheques. For enabling the TPH to generate cheques, it must be loaded. Loading means to transfer money from the bank account to the TPH account. The TPH is used for communication between the user and the bank as well as between the user and the provider. Due to the design of the TPH, i.e. it consists of a user-managed and a bank-managed part to perform user and bank transactions, bank and user are tamper proved against each other (compare with the wallet-observer concept in [17]). Thus, if the necessary functions for the charging process are performed by the TPH, all parties can be sure about the correct execution.

First, the user must load the TPH. This happens to be an on-line process between bank and user, but, in fact, it is independent from using the mix-mediated service. Loading the TPH means to withdraw money from the real

user account. Thus, the user selects the amount that he wants to withdraw. If there is enough money on the bank account, the amount is transferred to the TPH where it is deposited to the TPH account. Now the cheques can be created off-line. If the TPH account is empty, the TPH must be loaded again.

Second, if the user wants to send a cheque, it has to be generated. Thus, the TPH makes out this cheque as described in Section 3.2. The maximum amount of the cheque can be calculated from the number of ticks and the amount per tick. Therefore, the TPH is also responsible for generating the ticks.

Now the cheque can be used for payment of anonymous communication. For enabling the providers and the bank to test the correctness of a cheque, the included data must be signed by the bank. This procedure is also performed off-line in the TPH as the correct withdrawal from the bank account is already done and the TPH allows only cheques in the range of the balance of the TPH account.

As mentioned in Section 6.2, user anonymity must also be kept against the bank. The TPH is responsible that the bank does not have a possibility to trace the cheques. For this reason it uses blind signatures [18]. I.e. the cheques will be blinded before they are signed by the bank transaction part of the TPH. The sign key of the bank is stored in this part of the TPH. Therefore, the signing can also be performed off-line. Then the cheques are sent to the providers. They can test their correctness by checking the bank signature. If the providers have also received the ticks for the cheques, they can cash them at the bank.

On the other hand, the bank must test that the provider who sent the cheque was the intended recipient of the cheque. This can be easily performed: The cheque could include the public key of the provider as sign for the recipient. The bank could use a challenge-response protocol to check the identity of the providers: It sends a random number to the provider. The provider must encrypt this number with its private key. The result is sent back to the bank that tries to decrypt it with the public key included in the cheque. If the bank has also successfully tested its own signature included in the cheque, it can be sure that this cheque is correct, and the money is paid to the provider. Thus, both the bank and the providers can trust the payment anonymous performed by the users.

7 Summary and Outlook

In this paper, we introduced different protocols for payment of an anonymity service provided by mixes. To keep the advantage of one party over the other as small as possible, we suggested the usage of an interactive micro-payment scheme that combines the anonymity scheme "mixes" and tick payment. There exits no implementation of this approach but the application to existing systems like onion routing [19] and mixmaster [2] should be feasible.

In our protocols, we assumed that the sender has to pay for the anonymity service and, therefore, used encryption schemes that maintain the anonymity of the sender. However, the recipient can maintain his anonymity too by us-

ing mixes. However, the application of the protocols to this case seems to be straightforward.

The protocols suggested allow payment before and after delivery. We explained the protocols for asymmetric encrypting schemes. The application to the more efficient hybrid encrypting schemes was sketched and should be feasible.

Hash functions without trapdoor are the basis for the described protocols. Whether trapdoor hash functions are suitable and perhaps more efficient for payment of the mix providers, must also be investigated in future.

We pointed out possible threats to estimate the achieved anonymity of the protocols. We can summarize that it makes sense just to focus on the mix network. The discussion of boundary conditions, especially the relations between users, network and providers, must be considered as well to allow secure anonymous payment.

When the message transfer is not complete, there are problems for both, the user and the provider. Additional functions or protocols steps are necessary to guarantee the security and the fairness of the payment. These extensions and, moreover, possible risks if payment is triggered by the recipient is another point to be further discussed.

Another open question is how to solve the sharing of payment between the sender and the recipient. Attacks, regarding the sharing, are of special interest. On one hand, the users must not pay less than the agreed amount of money. On the other hand, it must be avoided that the providers can get more money than necessary.

In summary, though there are still topics for further investigation, this paper has shown that payment does not necessary break anonymity.

We want to thank Guntram Wicke, Prof. Dr. Andreas Pfitzmann and Andreas Graubner for many helpful discussions.

References

1. David Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM* 24/2 (1981) 84-88.
2. L. Cottrell, "Mixmaster and Remailer Attacks", `http://www.obscura.com/~loki/remailer/remailer-essay.html`.
3. Andreas Pfitzmann, Birgit Pfitzmann and Michael Waidner, "ISDN-MIXes Untraceable Communication with Very Small Bandwidth Overhead", 7th IFIP International Conference on Information Security (IFIP/SEC'91), Elsevier, Amsterdam 1991.
4. David Goldschlag, Michael Reed and Paul Syverson, "Hiding Routing Information", Workshop on Information Hiding, LNCS 1174, Springer-Verlag, Berlin 1996, 137-150.
5. C. Gülcü and G. Tsudik, "Mixing E- mail with BABEL", ISOC Symposium on Network and Distributed System Security, 1996, 2-16.

6. Dogan Kesdogan, Hannes Federrath, Anja Jerichow, and Andreas Pfitzmann, "Location management strategies increasing privacy in mobile communication", 12th IFIP International Conference on Information Security (IFIP/SEC'96), Chapman & Hall, London 1996, 39-48.
7. T. Lopatic, C. Eckert, and U. Baumgarten, "MMIP - Mixed Mobile Internet Protocol", CMS 97 - Communications and Multimedia Security, IFIP TC-6 and TC-11, 22-23 Sept. 1997 in Athens (Greece).
8. E. Franz, A. Jerichow, and G. Wicke, "Payment Scheme for Mixes Providing Anonymity", International IFIP Working Conference on Electronic Commerce 98; June 4-5th 1998, LNCS 1402, Springer-Verlag, Berlin 1998, 94-108.
9. Andreas Pfitzmann and Michael Waidner, "Networks Without User Observability", *Computers & Security* 6/2, 158-166, February 1987.
10. E. Franz, A. Graubner, A. Jerichow, and A. Pfitzmann, "Modelling mix-mediated anonymous communication and preventing pool-mode attacks", 14th IFIP International Conference on Information Security (IFIP/SEC'98), Chapman & Hall, London 1998.
11. Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, (2nd ed.) New York 1996.
12. Torben P. Pedersen "Electronic Payments of Small Amounts", *Security Protocols 1996*, LNCS 1189, Springer-Verlag, Berlin 1997, 59-68.
13. Anja Jerichow, Jan Müller, Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner, "Real-Time Mixes: A Bandwidth-Efficient Anonymity Protocol", *IEEE Journal on Selected Areas in Communications*, Special issue "Copyright and privacy protection", 4(1998).
14. E. Franz, A. Graubner, A. Jerichow, and A. Pfitzmann, "Comparison of commitment schemes used in mix-mediated anonymous communication for preventing pool-mode attacks", Australasian Conference on Information Security and Privacy (ACISP'98) in Brisbane, LNCS, Springer-Verlag.
15. Antoon Bosselaers, Ronald Cramer, Rolf Michelsen, Stig Mjølsnes, Frank Müller, Torben Pedersen, Birgit Pfitzmann, Cristian Radu, Peter de Rooij, Berry Schoenmakers, and Matthias Schunter, "Functionality of the Basic Protocols", CAFE Public Report IHS8341, CWI Amsterdam, October 7, 1995; CAFE (Esprit 7023) Deliverable IHS 8341 (confidential), September1,1995.
16. Andreas Pfitzmann, Birgit Pfitzmann, Matthias Schunter, and Michael Waidner, "Trusting Mobile User Devices and Security Modules", *Computer* 30/2 (1997) 61-68.
17. David Chaum, "Design Concepts for Tamper Responding Systems", Crypto '83, Plenum Press, New York 1984, 387-392.
18. David Chaum, "Blind Signatures for untraceable payments", Crypto '82, Plenum Press, New York 1983, 199-203.
19. Paul F. Syverson, David M. Goldschlag, and Michael G. Reed, "Anonymous Connections and Onion Routing", *IEEE Symposium on Security and Privacy*, IEEE Computer Press, 1997, 44-54.