To appear in *Journal of Computer Security*, IOS Press.

Digital Payment Systems with Passive Anonymity-Revoking Trustees*

Jan Camenisch and Ueli Maurer
Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland
e-mail: {camenisch | maurer}@inf.ethz.ch

Markus Stadler
Union Bank of Switzerland, Ubilab, Bahnhofstrasse 45, 8021 Zurich, Switzerland
e-mail: Markus.Stadler@ubs.com

Anonymity of the participants is an important requirement for some applications in electronic commerce, in particular for payment systems. Because anonymity could be in conflict with law enforcement, for instance in cases of blackmailing or money laundering, it has been proposed to design systems in which a trustee or a set of trustees can selectively revoke the anonymity of the participants involved in a suspicious transaction. From an operational point of view, it can be an important requirement that such trustees are neither involved in payment transactions nor in the opening of an account, but only in case of a justified suspicion. In this paper we present an efficient anonymous digital payment systems satisfying this requirement. The described basic protocol for anonymity revocation can be used in on-line or off-line payment systems.

Keywords: Digital payment systems, electronic money, cryptography, privacy, anonymity revocation.

1 Introduction

In most presently-used payment systems the protection of the user's privacy relies exclusively on administrative and legal measures. Using cryptographic tools, in particular blind signature schemes [6], it is possible to design electronic payment systems that allow the customers to remain anonymous (e.g. [1, 5, 7, 10]), without affecting the other security requirements such as the unforgeability of money. However, while protecting the honest customers' privacy, the anonymity also opens the door for misuse by criminals, for instance for perfect blackmailing [22] or for money laundering.

*A preliminary version of this paper appeared in *Computer Security — ESORICS 96*, E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, volume 1146 of *Lecture Notes in Computer Science*, pages 33–43. Springer Verlag, 1996.

Therefore, in order to make anonymous payment systems acceptable to governments and banks, they must provide mechanisms for revoking a participant's anonymity under certain well-defined conditions. Such anonymity revocation must be possible only for an authorized trusted third party or a set of such parties. In this paper we refer to trusted third parties as *trustees*. In a concrete scenario a trustee could be a judge or a law enforcement agency.

The concept of *anonymity-revocable payment systems*, sometimes called fair payment systems, was introduced independently in [2] and [21]. This concept should not be confused with the kind of anonymity revocation described in [8], where it is considered that parties involved in a transaction can later reveal their identities if they wish to do so. The customer's privacy can be compromised neither by the bank nor by the payee, even if they collaborate, but the trustee or a specified set of trustees can (in cooperation with the bank) revoke a customer's anonymity. It is understood that the trustee(s) will satisfy a revocation request only if there exists sufficient evidence that a transaction is not lawful.

The first proposed anonymity-revocable systems are either inefficient because they are based on the cut-and-choose paradigm [2, 21], or they require the participation of the trustee in the opening of accounts or even in the withdrawal transactions [3, 4, 16, 21]. From an operational point of view, it is an important requirement that a trustee can be passive, i.e., that he need not be involved in regular transactions nor when a customer opens a new account. The goal of this paper is to present an efficient and secure anonymous digital payment systems satisfying this requirement.

Independently of this work, Frankel, Tsiounis and Yung proposed a different solution to this problem in [14]. A comparison with their scheme is given in Section 8.

The outline of the paper is as follows. In Sections 2 we give a brief overview of digital payment systems. Different types of anonymity revocation are introduced and discussed in Section 3. Section 4 summarizes the basic cryptographic primitives underlying the payment schemes. The basic on-line scheme is presented in Section 5 and extensions to off-line schemes and to multiple trustees are discussed in Section 6 and 7, respectively. Section 8 compares our schemes with other systems having passive trustees.

2 Digital payment systems

An electronic payment system consists of a set of protocols involving three interacting parties: a bank, a customer (the payer), and a shop (the payee). The customer and the shop have accounts with the bank. The goal of the system is to transfer money in a secure way from the customer's account to the shop's account. It is possible to identify three different phases: a *withdrawal* *phase* involving the bank and the customer, a *payment phase* involving the customer and the shop, and a *deposit phase* involving the shop and the bank. In an *off-line* system, each phase occurs in a separate transaction, whereas in an *on-line* system, payment and deposit take place in a single transaction involving all three parties.

The bank, the shop and the customer have different security requirements. The bank must ensure that money can be deposited only if it has previously been withdrawn. In particular, double-spending of digital money must be impossible. The shop, upon receiving a payment in an off-line system, must be assured that the bank will accept the payment. Finally, the customer must be assured that the withdrawn money will later be accepted for a payment and that the bank is not able to claim that the money has already been spent (called a framing attack), i.e., falsely accuse him of double-spending. Furthermore, the customer may require that his privacy be protected. We refer to [5] for a detailed discussion of security requirements for payment systems.

Anonymous electronic payment systems (e.g. [1, 5, 7, 10]) are based on a cryptographic mechanism called a blind signature scheme [6]. Such a signature scheme allows a signer (the bank) to sign a message (a coin) without seeing its content. Furthermore, while anyone, including a shop or the bank, is able to verify such a signature, even the bank is not able to link a particular signature with a particular instance of signing a message. In order to implement an anonymous payment system based on a blind signature scheme, any message signed (blindly) by the bank with the secret key corresponding to a particular public key is agreed to have a certain value (e.g. \$10). Different denominations can be realized by using a different public key for every denomination.

An obvious problem with digital money is that it can in principle be spent more than once. In an on-line system, double-spending can be prevented by the bank by checking the record of previous deposits. This requires that all deposit transactions (at least within the validity period of coins) are stored by the bank. In an off-line system, double-spending cannot be prevented, but it is possible to design systems that allow to revoke a customer's anonymity when the money is spent more than once. This can be achieved by assuring that the customer's identity is properly encoded in the signed message and by having the customer answer to a challenge during the payment such that the identity can be computed from the answers to two different challenges. Alternatively, the anonymity revocation mechanism shown in this paper could be used, but this is not the main purpose of presenting the mechanism.

3 Anonymity revocation by a trustee

Anonymity revocation by a trustee means that, when the need arises, the trustee can link a withdrawal transaction with the corresponding deposit transaction. There are two types of anonymity revocation, depending on which kind of information is available to the trustee:

- *Withdrawal-based* anonymity revocation: Based on the bank's view of a withdrawal transaction, the trustee can compute a piece of information that can be used (by the bank or a payee) to recognize the money when it is spent later. This type of anonymity revocation can for instance be used in case of blackmailing. When the owner of an account is forced to withdraw money and to transfer it to an anonymous criminal, the account owner could secretly inform the bank and the trustee could be asked to compute a value that can be put on a blacklist so that the money can be recognized when it is spent. This corresponds to putting the serial-number of a conventional bank-note on a blacklist.
- *Payment-based* anonymity revocation: Based on the bank's view of a deposit transaction, the trustee computes a parameter that can be linked by the bank with the corresponding withdrawal. This may for instance be needed when the suspicion of money laundering arises.

One of the security requirements of such a payment system is that the trustee must be capable only of anonymity revocation but that he cannot play a different role in the system. Thus, if the trustee's secret key was compromised, only the anonymity of customers would fall, but the system would remain secure from the bank's point of view. In particular, the trustee must be unable to forge money.

It is possible to distinguish three different approaches to achieving the above goals, according to the type of the trustee's involvement.

1. The trustee is involved in every withdrawal. In such systems [3, 16] the trustee plays the role of an intermediary during the withdrawal protocol and performs the blinding operation on behalf of the customer. The trustee can then trivially revoke the anonymity if needed.
2. The trustee is involved in the opening of accounts, but not in transactions (e.g. [4]). Such systems are potentially more efficient because normally an account is used for many transactions.
3. The trustee is not involed in any of the protocols of the payment system but is needed only for anonymity-revocation. In such systems the customer proves to the bank that the coin and the exchanged messages

contain information, encrypted under the trustee's public key, that allows to revoke the anonymity. This can in principle be achieved by application of the well-known cut-and-choose paradigm, as described independently in [2] and [21]. However, such a system is quite inefficient as explained in Section 8.

The goal of this paper is to present an efficient anonymity-revocable payment system that allows both types of anonymity revocation and in which, in contrast to the previously proposed efficient systems, the trustee is completely passive unless he is requested to revoke a person's anonymity. In particular, after initially publishing a public key, the trustee need neither be involved in the opening of an account nor in any withdrawal, payment, or deposit transaction.

4 Proofs and signatures based on discrete logarithms

Before defining some cryptographic primitives we explain our notation and introduce the algebraic preliminaries. The symbol $\|$ denotes the concatenation of two (binary) strings, the letter ϵ stands for the empty string, and by $\alpha \in_R \mathbb{Z}_q$ we mean that α is chosen uniformly at random from $\mathbb{Z}_q$. We assume the availability of a collision resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\ell$ (e.g. $\ell = 128$).

Let G be a finite cyclic group of order q and let $g \in G$ be a generator of G, such that computing discrete logarithms to the base g is infeasible. Like for most other cryptographic schemes based on the discrete logarithm problem, a public key in our context is constructed by computing $y = g^x$ for a secret key x chosen at random from $\mathbb{Z}_q$.

We next explain our notation for illustrating protocols (see Figure 1 for an example). The players' names are indicated in boxes on the top line and their lists of inputs are shown in brackets on the next line. The computations performed by the players and the steps of the protocol are shown between the two horizontal lines. Their lists of outputs in an honest execution of the protocol are shown in brackets on the bottom line. Dishonest players are not restricted to storing only their specified output. A player's view consists of the entire list of parameters seen during the execution of the protocol. For the analysis of protocols, in particular of the anonymity of a certain player, it is important to consider the entire view of other player(s). Whenever the protocol specifies that a player must verify a condition, it is assumed that if the verification fails, the protocol is stopped and all parties are informed.

We will make use of extensions of the Schnorr signature scheme [18]. A Schnorr signature for a message m is a pair (c, s) with $c \in \{0,1\}^\ell$ and $s \in \mathbb{Z}_q$ satisfying the verification equation

$$c = \mathcal{H}(m \| g^s y^c).$$

Player A		Player B
(m, g, y)		(g, x)
		$\tilde{r} \in_R \mathbb{Z}_q$ $\tilde{t} := g^{\tilde{r}}$
	$\xleftarrow{\ \tilde{t}\ }$	
$\gamma, \delta \in_R \mathbb{Z}_q$ $t := \tilde{t} g^{\gamma} y^{\delta}$ $c := \mathcal{H}(m \| t)$ $\tilde{c} := c - \delta \pmod q$		
	$\xrightarrow{\ \tilde{c}\ }$	$\tilde{s} := \tilde{r} - \tilde{c}x \pmod q$
	$\xleftarrow{\ \tilde{s}\ }$	
$s := \tilde{s} + \gamma \pmod q$		
(c, s)		

Figure 1: A protocol for obtaining a blind Schnorr signature.

Such a signature can be generated only if one knows the secret key x, by choosing r at random from $\mathbb{Z}_q$ and computing c and s according to

$$c = \mathcal{H}(m \| g^r)$$

and

$$s \equiv r - cx \qquad \pmod q.$$

Basically, a Schnorr signature with respect to a public-key (g, y) is a proof (depending on the message m to be signed) that the signer knows the discrete logarithm of the public key y to the base g.

We now give definitions for two cryptographic primitives for proving knowledge and equality of discrete logarithms, respectively. A (message-dependent) proof of knowledge of the discrete logarithm of a group element h to the base g, denoted $PKLOG(m, g, h)$ consists of a Schnorr signature with respect to a public-key (g, h) for the message $m \| g \| h$, i.e.,

$$PKLOG(m, g, h) = (c, s)$$

with

$$c = \mathcal{H}(m \| g \| h \| g^s h^c).$$

A (message-dependent) proof of equality of the discrete logarithm of h_1 to the base g_1 and the discrete logarithm of h_2 to the base g_2, denoted $PLOGEQ(m, g_1,$

$h_1, g_2, h_2)$, is a pair (c, s) satisfying the following condition:

$$PLOGEQ(m, g_1, h_1, g_2, h_2) = (c, s)$$

with

$$c = \mathcal{H}(m \| g_1 \| g_2 \| h_1 \| h_2 \| g_1^s h_1^c \| g_2^s h_2^c).$$

Such a proof can be computed if and only if one knows the discrete logarithms $\log_{g_1} h_1$ and $\log_{g_2} h_2$ and if they are both equal to the same value x. To generate a proof one first chooses r at random from $\mathbb{Z}_q$ and computes c and s according to $c = \mathcal{H}(m \| g_1 \| g_2 \| h_1 \| h_2 \| g_1^r \| g_2^r)$ and $s \equiv r - cx \pmod{q}$. Note that the message m can be the empty string.

The protocol shown in Figure 1 is a protocol for blindly issuing Schnorr-signatures. It was first proposed in a slightly different version in [17]. The protocol allows a player B to sign a message m chosen by player A without seeing m or receiving any information about m, and without knowing what player A's resulting Schnorr signature (c, s) will be. If both players follow the protocol then the pair (c, s) is a valid Schnorr signature for m:

$$g^s y^c = g^{\tilde{s}+\gamma} y^{\tilde{c}+\delta} = g^{\tilde{r}-\tilde{c}x+\gamma+\tilde{c}x} y^{\delta} = \tilde{t} g^{\gamma} y^{\delta} = t$$

and therefore the verification condition $c = \mathcal{H}(m \| g^s y^c)$ holds. To prove that the protocol is blind, i.e., that the signer's view is statistically independent of the message and the signature (c, s), one has to show that for every possible view and every possible signature there exists exactly one pair (γ, δ) of blinding factors which would result in that particular signature and view. Given any view consisting of $\tilde{r}$, $\tilde{t}$, $\tilde{c}$, and $\tilde{s}$ and any signature (c, s) of a message m, let

$$\begin{array}{lcll} \gamma & = & s - \tilde{s} & \pmod{q}, \\ \delta & = & c - \tilde{c} & \pmod{q}, \quad \text{and} \\ t^* & = & \tilde{t} g^{\gamma} y^{\delta}. & \end{array}$$

It remains to show that $t^* = t = g^s y^c$ is satisfied:

$$t^* = \tilde{t} g^{\gamma} y^{\delta} = g^{\tilde{r}+\gamma+\delta x} = g^{\tilde{r}+s-\tilde{s}+(c-\tilde{c})x} = g^{s+cx} g^{\tilde{r}-\tilde{s}-\tilde{c}x} = g^s y^c = t$$

The last two equalities hold because $\tilde{s} \equiv \tilde{r} - \tilde{c}x \pmod{q}$ and because (c, s) is a valid signature.

5 An efficient anonymous payment system with a passive anonymity-revoking trustee

In this section we describe the on-line payment scheme with a single denomination of coins. An extension to multiple denominations is straightforward. Extensions to off-line payment schemes are discussed in Section 6.

Let us explain the underlying ideas of our scheme. The main components of a coin are (1) a pair (h_p, z_p) satisfying $z_p = h_p^x$, where x is the bank's secret key, (2) a proof (denoted W) of this fact, and (3) a further proof (denoted V) needed to guarantee that anonymity revocation is possible. The proof W is given by the bank and the proof V can be computed by the customer on his own. To achieve anonymity, the proof W must be issued blindly: During withdrawal the customer sends the bank a blinded pair (h_w, z_w), the bank computes a proof corresponding to W based on this pair, and the customer transforms this proof into the proof W. This is achieved by a subprotocol of the withdrawal and is explained in Section 5.2. The anonymity of a coin can be revoked by the trustee if he can link pairs (h_p, z_p) and (h_w, z_w). This is guaranteed by a mechanism explained in Section 5.5.

5.1 *System setup*

To set up the payment system the bank chooses a finite group G of prime order q such that computing discrete logarithms in G is infeasible. Such a group is cyclic and thus every element (except the neutral element) is a generator. Today, the choice $q \approx 2^{170}$ appears to be secure unless the group has a special structure. Three elements g, g_1 and g_2 are chosen by a publicly verifiable pseudo-random mechanism to assure that the discrete logarithms of none of these elements with respect to one another is known. Finally, the bank chooses at random a secret key $x \in \mathbb{Z}_q$ and computes the public key $y = g^x$. The bank publishes G, g, g_1, g_2, and y.

The trustee randomly chooses his secret key $\tau \in \mathbb{Z}_q^*$ and computes and publishes the corresponding public key $y_T = g_2^\tau$.

5.2 *A subprotocol: a modified blind Schnorr signature scheme*

As mentioned before, the pair (h_p, z_p) is obtained blindly by the customer in a subprotocol, referred to as protocol **P** (see Figure 2), during which the bank sees only the pair (h_w, z_w). This protocol is an extension of the blind issuing protocol for Schnorr signatures due to Brands [1]. It is discussed here as an independent protocol (with its own players, input and output parameters) because it is of independent interest and because it will be reused later.

Protocol **P** takes place between two players A and B, substituted in the withdrawal protocol by the customer and the bank, respectively. A's input consists of the pair (g, y), where $y = g^x$ is B's public key, a secret message m, a group element h_w also known to B, and a blinding exponent α. A's output consists of the pair (h_p, z_p) and the proof

$$W = (c, s) = PLOGEQ(m, g, y, h_p, z_p)$$

Player A		Player B
(m, g, y, h_w, α)		(g, x, h_w)
		$z_w := h_w^x$
		$\tilde{r} \in_R \mathbb{Z}_q$
		$\tilde{t}_g := g^{\tilde{r}}$
	$\xleftarrow{z_w,\ \tilde{t}_g,\ \tilde{t}_h}$	$\tilde{t}_h := h_w^{\tilde{r}}$
$\gamma, \delta \in_R \mathbb{Z}_q$		
$h_p := h_w^\alpha$		
$z_p := z_w^\alpha$		
$t_g := \tilde{t}_g g^\gamma y^\delta$		
$t_h := \tilde{t}_h^\alpha h_p^\gamma z_p^\delta$		
$c := \mathcal{H}(m \| g \| h_p \| y \| z_p \| t_g \| t_h)$		
$\tilde{c} := c - \delta \pmod q$		
	$\xrightarrow{\tilde{c}}$	$\tilde{s} := \tilde{r} - \tilde{c}x \pmod q$
	$\xleftarrow{\tilde{s}}$	
$s := \tilde{s} + \gamma \pmod q$		
$W := (c, s)$		
(m, h_p, z_p, W)		

Figure 2: The protocol **P**.

which serves two purposes. On one hand, it is a blind Schnorr signature for the message m, and this blinding is achieved by the blinding exponents γ and δ. On the other hand, W also proves that the pair (h_p, z_p) satisfies $z_p = h_p^x$. From B's point of view, the proof is given for the pair (h_w, z_w). The exponent α for blinding h_p is chosen by player A before engaging in protocol **P**.

The proof that B's view of protocol **P** is unlinkable to (i.e., statistically independent of) A's output (m, h_p, z_p, c, s) is similar to the proof of blindness for the blind Schnorr signature protocol (see Section 4).

5.3 *The withdrawal protocol*

The actual withdrawal protocol, which uses protocol **P** as a subprotocol, is shown in Figure 3. It is based on a fair blind signature scheme due to Stadler [20]. The customer chooses a random exponent α, which plays two different roles in the protocol. On one hand, α serves as the blinding exponent (within protocol **P**) to transform the pair (h_w, z_w) into the pair (h_p, z_p). On the other hand, α is used in the first place to compute $h_w := g_1^{\alpha^{-1}} g_2$ and

$d := y_T^\alpha$, together with a proof, denoted as U, that the two values of α used in computing h_w and d are the same. The value d can be interpreted as a Diffie-Hellman-type encryption of h_p for the trustee and is stored by the bank for possible later anonymity revocation.

For the computation of U we use the fact that by exchanging base and input element of a discrete logarithm computation, the resulting discrete logarithm is inverted modulo the group order:

$$\log_g h \;\equiv\; (\log_h g)^{-1} \qquad (\mathrm{mod}\ q).$$

The coin consists of the coin number $c\#$, the values h_p, z_p, W, and a proof

$$V = PKLOG(\epsilon, g_2, h_p/g_1)$$

that the customer computes before spending the coin. The pair V is a proof that h_p equals $g_1 g_2^\alpha$ for some α known to the customer. This prevents the potential attack that in protocol **P**, seen as an independent protocol, player A could successfully choose $h_p = h_w^\alpha g^\beta$ and $z_p = z_w^\alpha y^\beta$ for some $\beta \neq 0$. Such an attack would allow a cheating customer to avoid later anonymity revocation. However, the customer can generate the proof V only if he chooses $\beta = 0$ and hence the described attack is not successful.

5.4 *The on-line payment protocol*

A coin can be spent by sending it to a shop. The shop verifies the coin and, if it is valid, passes the coin on to the bank. The bank checks the database of all previously spent coins. (By including an expiration date in the coin one can limit the size of this database.) If the coin is new, it is accepted and inserted into the database, and the shop's account is credited. The protocol is shown in Figure 4.

We now discuss why the scheme provides anonymity for the customer. Withdrawal and payment are unlinkable because of the blindness of the subprotocol **P**. However, although the blindness of protocol **P** is unconditional, i.e. information-theoretical, the anonymity of the payment scheme is only computational because of the revocation parameter d. The bank could link withdrawal and payment by testing whether $\log_{y_T} d = \log_{g_2}(h_p/g_1)$, but this is computationally infeasible because the bank does not know $\log_{g_2} y_T$ (see [1] for a discussion of the so-called Decision-Diffie-Hellman problem).

5.5 *Anonymity revocation*

As already mentioned, there are two kinds of anonymity revocation, namely withdrawal-based and payment-based revocation. The latter can be achieved by letting the trustee compute the value

$$(h_p/g_1)^\tau = (g_2^\alpha)^\tau = d$$

Customer		Bank
(g, y, g_1, g_2, y_T)		(g, x, g_1, g_2, y_T)
	$\xrightarrow{\text{identification}}$	
choose a random coin number $c\#$ $\alpha \in_R \mathbb{Z}_q^*$ $h_w := g_1^{\alpha^{-1}} g_2$ $d := y_T^{\alpha}$ $U := PLOGEQ(\epsilon, g_1, (h_w/g_2), d, y_T)$		
	$\xrightarrow{h_w,\ d,\ U}$	
		verify U
$(c\#, g, y, h_w, \alpha)$		(g, x, h_w)
	Protocol P	
$(c\#, h_p, z_p, W)$		
		debit the customer's account
$(c\#, h_p, z_p, \alpha, W)$		(d)

Figure 3: The withdrawal protocol.

from an h_p that is observed in a payment. The computed value d can then be searched in the bank's revocation database containing the transcripts of the withdrawal transactions, including the values d.

Withdrawal-based anonymity revocation is achieved as follows. Given the value d observed in a withdrawal transaction, the trustee computes

$$g_1 d^{\tau^{-1}} = g_1 g_2^{\alpha} = h_p.$$

This value can be put on a blacklist for recognizing the coin later when it is spent. The two types of anonymity revocation are possible because

$$\alpha = (\log_{g_1}(h_w/g_2))^{-1} = \log_{y_T} d = \log_{g_2}(h_p/g_1)$$

holds for every triple (h_w, h_p, d) generated during a legitimate withdrawal transaction.

Customer	Shop	Bank
$(g_1, g_2, c\#, h_p, z_p, \alpha, W)$	(g, y, g_1, g_2)	(g, y, g_1, g_2)
$V := PKLOG(\epsilon, g_2, h_p/g_1)$		
$\xrightarrow{c\#,\ h_p,\ z_p,\ W,\ V}$		
	verify V and W	
	$\xrightarrow{c\#,\ h_p,\ z_p,\ W,\ V}$	
		verify V and W
		verify that $c\#$ was not spent so far
		credit Shop's account

Figure 4: The payment protocol in the on-line scheme.

5.6 *Efficiency considerations*

A coin in the proposed scheme consists of two group elements, two hash values, and two numbers smaller than q. When the group allows for a compact representation of its elements, the signatures can be quite short. For instance, elements of an elliptic curve with order q over a field of cardinality close to q can be represented by two field elements. Hence, for $q \approx 2^{170}$, the total signature length is roughly $6 \log_2 q + 256 \approx 1300$ bits. This could be reduced further to about 1000 bits by a compressed representation of group-elements and by using the same challenge for the proofs V and W.

6 Extensions to off-line payments

In an off-line system, double-spending can only be detected after the fact, but it cannot be prevented. Detecting double-spending is trivial, but identifying the cheating customer requires an additional mechanism in the protocols. In the presented system, a natural solution appears to be to involve the trustee for exposing double-spenders. This solution is unsatisfactory from an operational point of view, when many instances of double-spending occur. In Section 6.1 we describe a modified protocol that allows the bank to identify double-spenders without the help of the trustee.

In Section 6.2 we discuss the use of tamper-resistant hardware (called observer in this context) for preventing double-spending in off-line systems. Since no secure tamper-proof components are currently available, such devices must be combined with techniques for identifying double-spenders discussed before. In contrast to off-line systems without observers, it is acceptable to involve the trustee for identifying cheaters because the breaking of an observer can be considered a rare event. However, it is also possible to use the technique of Section 6.1 in an observer-based system.

6.1 *Enabling the bank to identify double-spenders*

The payment protocol of Figure 5 allows the bank to identify double-spenders, but the anonymity of honest customers is not compromised. The basic idea is to redefine the proof V and to make use of the following fact, which was already used in a similar way in [1] for the purpose of identifying double-spenders. If the value $t = g^r$ generated by the signer for issuing a Schnorr signature (or in a message-dependent $PKLOG$) is used for signing more than one message, it is easy to compute the secret key from the two signatures. Let (c_1, s_1) and (c_2, s_2) denote the two distinct signatures and let $y = g^x$. If

$$g^{s_1} y^{c_1} = t = g^{s_2} y^{c_2},$$

then we have $s_1 + c_1 x \equiv s_2 + c_2 x \pmod q$ and hence

$$x \equiv \frac{s_1 - s_2}{c_2 - c_1} \pmod q.$$

This fact is used by forcing the customer to use the same value $t_p = g_2^{r_p}$ in every potential payment of a given coin. This is achieved by having the customer choose and store r_p during withdrawal and by including t_p instead of the coin number $c\#$ in the proof W. Furthermore, V depends on a message containing information about the shop and the withdrawal transaction, where the latter is achieved by including W, which contains a hash-value of t_p, h_p, and z_p. The counter cnt is included in the hash value because otherwise the shop could deposit a coin twice and, in reply to bank's objection, blame the customer of having it spent twice.

When a coin is spent more than once, the bank can compute the value α of that coin because α serves as the customer's secret key in the payment protocol.

6.2 *Using observers for preventing double-spending*

It is quite unsatisfactory that in anonymous off-line payment schemes the multiple spending of coins can only be detected but not prevented. Chaum et al. [9] proposed as a solution to use so-called *observers*, small tamper-resistant hardware devices that are issued by the bank for every customer.

$$
\begin{array}{lcr}
\boxed{\text{Customer}} & & \boxed{\text{Shop}} \\
(h_p, z_p, W, t_p, r_p, \alpha) & & (g, y, g_1, g_2) \\
\hline
\downarrow & & \downarrow \\
& \xrightarrow{h_p,\ z_p,\ W,\ t_p} & \\
& & \text{verify } W \\
& & c_p := \mathcal{H}(ID_{Shop}, cnt, W) \\
& \xleftarrow{c_p} & \\
s_p := r_p - c_p\alpha \pmod q & \xrightarrow{s_p} & \\
& & \text{verify } g_2^{s_p}(h_p/g_1)^{c_p} \stackrel{?}{=} t_p \\
& & V := (c_p, s_p) \\
\hline
& & \downarrow \\
& & (h_p, z_p, W, t_p, V, cnt)
\end{array}
$$

Figure 5: The payment protocol in the off-line scheme.

Transactions can only be carried out in cooperation with the observer (see Figure 6). In particular, the observer keeps a list of active (withdrawn but not yet spent) coins and refuses to cooperate in spending a coin a second time, i.e., it cooperates only in spending coins contained in its list of active coins.

The following requirements guarantee the customers' privacy (for a more detailed discussion see [11]):

- The observer must not be able to communicate with anyone except the customer. In particular, the observer must not be able to establish a subliminal channel to the bank.
- Even if later the bank gets access to the observer and obtains all its protocol views, the bank must not be able to trace payments. This implies that the observer must not have an internal clock.

In Figure 6 the communication between observer, customer, bank and shop is illustrated. Because each customer's observer is unique, the bank could link the communications (ii) and (iii) with (i). In order to satisfy the conditions stated above, the communication (iv) must be unlinkable with communications (i) to (iii) .

We now describe how the on-line payment scheme presented in Section 5 can be turned into an observer-based off-line scheme. Let ω and $y_o = {g_1}^{\omega}$ denote the observer's secret key and public key, respectively. (Note that each observer has its own secret key/public key pair.) The basic idea is that the customer and the observer share the value α such that neither of them alone

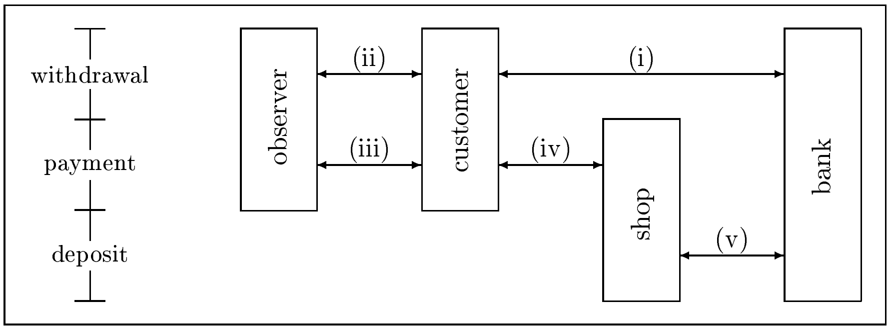

Figure 6: The customer needs the help of the observer for carrying out withdrawal and payment transactions. For each coin the observer only participates once in a payment transaction; hence multiple spending of a coin can be prevented.

knows α. More precisely, α is replaced by the product $\hat{\alpha}\breve{\alpha}$ modulo q, where $\hat{\alpha}$ is chosen (and kept secret) by the observer, and $\breve{\alpha}$ is chosen by the customer. During the withdrawal of a coin, the customer must prove to the bank that the value α is indeed shared with the observer. All operations involving α in the on-line protocol now require the observer's cooperation. Hence the observer can prevent double-spending.

Figure 7 shows the modifications in the subprotocol **P**. The parameter α occurs in the computation of h_p, z_p, and t_h. The observer obtains only values (h_w, z_w, and $\tilde{t}_h$) which the bank already knows; hence no relevant information is leaked to the observer. The resulting subprotocol is called $\mathbf{P}_\circ$, in which parameter $\breve{\alpha}$ replaces player A's input α in protocol **P**.

Figure 8 describes the withdrawal protocol. After the identification, the customer and the observer jointly compute the values h_w and d. Then they jointly construct the two proofs

$$U_1 = PKLOG(\epsilon, y_o, h_w/g_2)$$

and

$$U_2 = PLOGEQ(\epsilon, g_1, h_w/g_2, d, y_T) \ .$$

The proof U_2 is identical to the proof U in the on-line scheme and convinces the bank that d is formed correctly. The proof U_1 convinces the bank that the observer is indeed engaged in the protocol, because knowledge of both $\log_{y_o} h_w/g_2$ (proved in U_1) and $\log_{g_1} h_w/g_2$ (proved in U_2) implies knowledge of $\log_{g_1} y_o = \omega$, the observer's secret key.

An important point in the construction of these two proofs is the additional blinding performed by the customer using the random values r_1 and r_2. This

Observer		Player A in $\mathbf{P}$
$(\hat{\alpha})$		$(g, y, h_w, z_w, \tilde{t}_h, \check{\alpha}, \gamma, \delta)$
	$\xleftarrow{h_w, z_w, \tilde{t}_h}$	
$\hat{h}_p = h_w^{\hat{\alpha}}$ $\hat{z}_p = z_w^{\hat{\alpha}}$ $\hat{t}_h = \tilde{t}_h^{\hat{\alpha}}$	$\xrightarrow{\hat{h}_p, \hat{z}_p, \hat{t}_h}$	$h_p = \hat{h}_p^{\check{\alpha}}$ $z_p = \hat{z}_p^{\check{\alpha}}$ $t_h = \hat{t}_h^{\check{\alpha}} h_p^{\gamma} z_p^{\delta}$
(h_w)		(h_p, z_p, t_h)

Figure 7: In the observer-based system, the computation of h_p, z_p, and t_h within protocol $\mathbf{P}$ must be performed jointly by customer and observer. The modified protocol $\mathbf{P}$ is referred to as protocol $\mathbf{P}_\circ$.

prevents the bank from computing α by using s_1 and $\hat{s}_1$ (or s_2 and $\hat{s}_2$) in case the bank obtained the values $\hat{s}_1$ and $\hat{s}_2$. This could happen for instance when the observer is returned to the bank or when the bank knows the seed of the pseudo-random number generator in the observer.

At the end of the withdrawal protocol the observer and the customer store h_w for the purpose of identifying the coin later in the payment protocol (shown in Figure 9).

The payment protocol is very similar to the protocol in Figure 5, except that the observer and the customer jointly compute the proof V. It is essential that the communication between the customer and the observer is unlinkable with the communication between the shop and the customer so that neither the bank nor the observer obtains useful information about the correspondence of withdrawal and payment transactions. This is achieved by a blinding of the values t_p and c_p. The correctness of the protocol in Figure 9 can be seen as follows:

$$g_2^{s_p}(h_p/g_1)^{c_p} = g_2^{\check{\alpha}\hat{r}_p - \hat{c}_p\hat{\alpha}\check{\alpha} + r_p}(h_p/g_1)^{c_p} = \hat{t}_p^{\check{\alpha}} g_2^{r_p + \delta_p\hat{\alpha}\check{\alpha}} = \hat{t}_p^{\check{\alpha}} g_2^{r_p}(h_p/g_1)^{\delta_p} = t_p$$

and therefore the verification equation hold.

7 Sharing the revocation capability among several trustees

To achieve higher security against fraudulent anonymity revocation, the revocation capability can be shared among several trustees such that only predefined subsets of the trustees are able to cooperatively revoke a customer's

Observer		Customer		Bank
(y_o, g_1, y_T)		(g, y, g_1, g_2, y_T)		$(g, x, g_1, g_2, y_T, y_o)$
	$\xleftarrow{\text{start}}$		$\xrightarrow{\text{identification}}$	
$\hat{\alpha} \in_R \mathbb{Z}_q^*$ $\hat{h}_w := g_1^{1/\hat{\alpha}}$ $\hat{d} := y_T^{\hat{\alpha}}$		choose $c\#$ $\check{\alpha} \in_R \mathbb{Z}_q^*$		
	$\xrightarrow{\hat{h}_w,\ \hat{d}}$			
$\hat{r}_1, \hat{r}_2 \in_R \mathbb{Z}_q^*$ $\hat{t}_1 := g_1^{\hat{r}_1}$ $\hat{t}_2 := g_1^{\hat{r}_2},\ \hat{t}_d := \hat{d}^{\hat{r}_2}$		$h_w := \hat{h}_w^{1/\check{\alpha}} g_2$ $d := \hat{d}^{\check{\alpha}}$ $r_1, r_2 \in_R \mathbb{Z}_q^*$		
	$\xrightarrow{\hat{t}_1,\ \hat{t}_2,\ \hat{t}_d}$			
		$t_1 := \hat{t}_1^{1/\check{\alpha}} g_1^{r_1}$ $t_2 := \hat{t}_2^{1/\check{\alpha}} g_1^{r_2},\ t_d := \hat{t}_d^{1/\check{\alpha}} d^{r_2}$ $c_1 := \mathcal{H}(y_o \Vert h_w/g_2 \Vert t_1)$ $c_2 := \mathcal{H}(g_1 \Vert d \Vert h_w/g_2 \Vert y_T \Vert t_2 \Vert t_d)$		
	$\xleftarrow{c_1,\ c_2}$			
$\hat{s}_1 := (\hat{r}_1 - c_1/\hat{\alpha})/\omega \pmod q$ $\hat{s}_2 := \hat{r}_2 - c_2/\hat{\alpha} \pmod q$				
	$\xrightarrow{\hat{s}_1,\ \hat{s}_2}$			
		$s_1 := \hat{s}_1/\check{\alpha} + r_1 \pmod q$ $s_2 := \hat{s}_2/\check{\alpha} + r_2 \pmod q$ $U_1 := (c_1, s_1),\ U_2 := (c_2, s_2)$		
			$\xrightarrow{h_w,\ d,\ U_1,\ U_2}$	
				verify U_1 and U_2
$(\hat{\alpha})$		$(c\#, g, y, h_w, \check{\alpha})$		(g, x, h_w)
		Protocol $\mathbf{P_o}$		
(h_w)		$(c\#, h_p, z_p, W)$		
				debit account
$(h_w, \hat{\alpha})$		$(c\#, h_w, h_p, z_p, \check{\alpha}, W)$		(d)

Figure 8: The withdrawal protocol in the observer-based system.

Observer		Customer		Shop
$(g_2, \hat{\alpha})$		$(g_1, g_2, c\#, h_w, h_p, z_p, \breve{\alpha}, W)$		(g, y, g_1, g_2)
	$\xleftarrow{\quad h_w \quad}$		$\xrightarrow{\; c\#,\ h_p,\ z_p,\ W \;}$	
reject if coin is not on list of active coins				verify W
$\hat{r}_p \in_R \mathbb{Z}_q$				
$\hat{t}_p := g_2^{\hat{r}_p}$	$\xrightarrow{\quad \hat{t}_p \quad}$			
		$r_p, \delta_p \in_R \mathbb{Z}_q$		
		$t_p := \hat{t}_p^{\alpha} g_2^{r_p} (h_p/g_1)^{\delta_p}$		
			$\xrightarrow{\quad t_p \quad}$	
				$c_p := \mathcal{H}(ID_{Shop}, cnt, W, t_p)$
			$\xleftarrow{\quad c_p \quad}$	
		$\hat{c}_p := c_p - \delta_p \pmod q$		
	$\xleftarrow{\quad \hat{c}_p \quad}$			
$\hat{s}_p := \hat{r}_p - \hat{c}_p \hat{\alpha} \pmod q$				
	$\xrightarrow{\quad \hat{s}_p \quad}$			
		$s_p := \breve{\alpha} \hat{s}_p + r_p \pmod q$		
			$\xrightarrow{\quad s_p \quad}$	
				verify $g_2^{s_p} (h_p/g_1)^{c_p} \stackrel{?}{=} t_p$
delete coin from list				$V := (c_p, s_p)$
				(h_p, z_p, W, cnt, V)

Figure 9: The payment protocol in the observer-based system.

anonymity. Borrowing terminology from the literature on secret sharing, the set of all subsets with the revocation capability is called an access structure.

We first consider the simple case where all trustees must cooperate. This can be achieved by letting each trustee choose a secret key τ_i and defining τ to be the product of the τ_i. Raising a value to the power τ or τ^{-1} during anonymity revocation is achieved by asking all trustees to consecutively compute the τ_i-th or τ_i^{-1}-th powers, respectively. Each trustee can perform the needed computations without revealing the secret τ_i.

In the above solution all trustees must be available and must cooperate for revoking the anonymity. To increase the robustness and the availability of the revocation mechanism, a so-called threshold scheme can be applied: for a threshold t, all coalitions of at least $t+1$ out of n trustees are able to revoke the anonymity, while coalitions of t or less trustees are not. The realization is based on Shamir's secret sharing schemes [19] and Feldman's verifiable secret sharing scheme [13]. A concrete problem in such a realization is that exponentiations with both τ and τ^{-1} must be possible in a distributed manner. A solution is described in [15] for the case $t < n/2$ if all trustees are honest and for the case $t < n/3$ if up to t trustees may be cheating. Another solution is to avoid the exponentiations with τ^{-1}, which occurs only in payment-based revocation. In the setup phase, the trustees collectively compute a second public key $\tilde{y}_T = g_1^{\tilde{\tau}}$. During the payment, the customer must send the additional value $\tilde{d} = \tilde{y}_T^{1/\alpha}$ to the shop. Moreover, the proof $V = PKLOG(\epsilon, g_2, h_p/g_1)$ is replaced by $PLOGEQ(\epsilon, g_2, (h_w/g_1), \tilde{d}, \tilde{y}_T)$. The trustees can thus perform a withdrawal-based revocation by computing

$$(h_w/g_2)^{\tilde{\tau}} = g_1^{\tilde{\tau}/\alpha} = \tilde{y}_T^{1/\alpha} = \tilde{d} \ ,$$

which is now part of the coin.

8 Comparison with other schemes with passive trustees

In this section we compare our on-line scheme with the cut-and-choose based approaches [21, 2] and the recent proposal of [14].

We sketch the scheme of [21] (which, both from the conceptual and the efficiency points of, view is similar to the scheme of [2]). In order to obtain a blind signature on a message m, the customer prepares $2K$ blinded messages, each of which contains m encrypted with the trustee's public key as well as a session identifier encrypted with the trustee's public key. K is a security parameter. These encryptions are probabilistic (i.e., the text is padded with a random string before encryption) in order to prevent decryption by an exhaustive search over a small set of possible values. To check that these messages are properly formed, the bank chooses a random subset of K blinded messages and asks the customer to open all of them, where "open" means presenting the random padding used for encrypting the session identifier. For the purpose of possible later anonymity revocation, the bank stores the corresponding K encryptions of m. Then it blindly signs the remaining K messages that were not opened. Such a coin (a blind signature for the message m) is valid if the bank's signature is valid and if it can be verified that m had correctly been encrypted for the trustee.

In such a system, withdrawal-based revocation can be achieved by asking the trustee to open the encryptions of m which the bank obtained and stored during the withdrawal protocol. Payment-based anonymity revocation can be achieved by asking the trustee to decrypt the encrypted session ID contained in each of the K components of the signature. The probability that a dishonest customer manages to escape payment-based or withdrawal-based anonymity revocation is $1/\binom{2K}{K} \approx 2^{-2K}\sqrt{\pi K}$. To achieve reasonable security, K should be at least 20. Each of the K components consists of a random padding string and a public-key encrypted value. In order to achieve the same security level as in our scheme, the lengths of these two values must be at least 64 and 768 bits, respectively. This results in a total signature length of close to $17,000$ bits, which is about 13 times longer than coins in our scheme (see Section 5.6).

The scheme presented in [14] is based on Brands' payment system [1] and on so-called "indirect discourse proofs." Two such proofs are used to convince the bank and, independently, the shop that the trustee can revoke the anonymity of a coin. Technically, an indirect discourse proof consists of an ElGamal encryption [12] of either the customer's identity (for payment-based revocation) or a unique part of the coin (for withdrawal-based revocation) and a proof that the correct value is encrypted. Conceptually, this technique is similar to that described in this paper, but the system of [14] is less efficient. For instance, coins in [14] are approximately twice as long.

Acknowledgments

Some ideas of this paper are based on results of a previous cooperation with Jean-Marc Piveteau. The authors are grateful to Ronald Cramer and the anonymous referees for providing helpful comments. The first author is supported by the Swiss Commission for Technology and Innovation (KTI), and by the Union Bank of Switzerland.

References

[1] S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI, Apr. 1993.

[2] E. Brickell, P. Gemmel, and D. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *Proceedings of the Sixth Annual ACM-SIAMs*, pages 457–466. Association for Computing Machinery, Jan. 1995.

[3] J. Camenisch, J.-M. Piveteau, and M. Stadler. Faire Anonyme Zahlungssysteme. In F. Huber-Wäschle, H. Schauer, and P. Widmayer, editors, *GISI 95*, Informatik aktuell, pages 254–265. Springer Verlag Berlin, Sept. 1995.

[4] J. Camenisch, J.-M. Piveteau, and M. Stadler. An efficient fair payment system. In *3rd ACM Conference on Computer and Communicatons Security*, pages 88–94, New Delhi, Mar. 1996. Association for Computing Machinery.

[5] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler. An efficient payment system protecting privacy. In D. Gollmann, editor, *Computer Security — ESORICS 94*, volume 875 of *Lecture Notes in Computer Science*, pages 207–215. Springer Verlag, 1994.

[6] D. Chaum. Blind signature systems. In D. Chaum, editor, *Advances in Cryptology — CRYPTO '83*, page 153. Plenum, 1983.

[7] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, Oct. 1985.

[8] D. Chaum. Privacy protected payments — unconditional payer and/or payee untraceability. In D. Chaum and I. Schaumüller-Bichl, editors, *SMART CARD 2000: The Future of IC Cards, Proceedings of the IFIP WG 11.6 International Conference*, pages 69–93. Elsevier Science Publishers B.V. (North-Holland), 1989.

[9] D. Chaum. Achieving electronic privacy. *Scientific American*, pages 96–101, Aug. 1992.

[10] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *Advances in Cryptology — CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer Verlag, 1990.

[11] R. J. F. Cramer and T. P. Pedersen. Improved privacy in wallets with observers. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 329–343. Springer-Verlag, 1994.

[12] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology — CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Verlag, 1985.

[13] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proc. 28th IEEE Symp. Found. Comp. Sc.*, pages 427–437, 1987.

[14] Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect discourse proofs:" Achieving efficient fair off-line e-cash. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology — ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 286–300. Springer Verlag, 1996.

[15] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 354–371. Springer Verlag, 1996.

[16] M. Jakobsson and M. Yung. Revokable and versatile electronic money. In *3rd ACM Conference on Computer and Communicatons Security*, pages 76–87, New Delhi, Mar. 1996. Association for Computing Machinery.

[17] T. Okamoto. Provable secure and practical identification schemes and corresponding signature schemes. In E. F. Brickell, editor, *Advances in Cryptology — CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer-Verlag, 1993.

[18] C. P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.

[19] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, Nov. 1979.

[20] M. Stadler. *Cryptographic Protocols for Revocable Privacy.* Ph.D. Thesis, ETH Zürich, 1996. Diss. ETH No. 11651.

[21] M. Stadler, J.-M. Piveteau, and J. Camenisch. Fair blind signatures. In L. C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 209–219. Springer Verlag, 1995.

[22] S. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computer & Security*, 11(6):581–583, 1992.