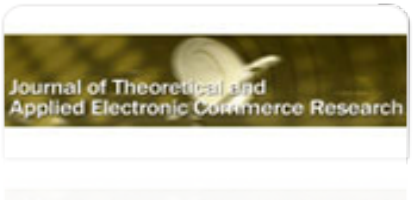

Journal of Theoretical and Applied Electronic Commerce Research
E-ISSN: 0718-1876
ncerpa@utalca.cl
Universidad de Talca
Chile

Neumann, Heike; Schwarzpaul, Thomas
Digital Coins: Fairness Implemented by Observer
Journal of Theoretical and Applied Electronic Commerce Research, vol. 1, núm. 1, april, 2006, pp. 1-15
Universidad de Talca
Curicó, Chile

Digital Coins: Fairness Implemented by Observer

Heike Neumann[1] **and Thomas Schwarzpaul**[2]

[1] Philips Semiconductors
Business Line Identification, CSCC
Georg-Heyken-Straße 1
D-21147 Hamburg, Germany
[2] University of Giessen
Mathematical Institute
Arndstraße 2
D-35392 Giessen, Germany

Abstract

For real-world applications digital coin systems, i.e., off-line payment systems offering not only the unforgeability of conventional coins but also the anonymity of customers making purchases, need to have some additional features.

One of these additional features is the hardware protection of the system, provided by dedicated tamper-resistant devices called **observers** which are used to physically prevent illegitimate copying of coins.

Another essential feature for practical applications is anonymity-revocation mechanisms. Basically, digital coin systems guarantee **perfect** anonymity, i.e., the bank cannot link views of the withdrawal and payments in order to determine whether or not a customer spent her money at a certain shop. The customer's privacy protection is the main difference between coin systems and those based on credit card or cheque based systems. While privacy protection is a desirable property of cash systems, perfect anonymity is not. Perfect anonymity makes possible perfect blackmailing or money laundering. To prevent such "perfect crime" it must be possible to revoke the anonymity of customers in case of need. Anonymity revocation is done by a trusted third party. Cash systems that allow anonymity revocation are called **fair**.

We present a coin system featuring both observer and fairness, showing that both concepts do not interfere with each other and can be implemented simultaneously without loss of security. We prove this claim not only by presenting a fair variant of the Brands' coin system but additionally by outlining a generic framework for fair wallets in which essentially any blind signature scheme can be used. Unlike other fair off-line coin systems, fairness is implemented with the help of the observer, thereby reducing the computational effort during the withdrawal.

Key words: Electronic cash, digital coins, observer, fairness

1 Introduction

Electronic Commerce is one of the most important applications for the Internet and mobile devices. One prerequisite for establishing an (mobile) electronic marketplace is **secure payment**. Many protocols have been proposed to implement different kinds of payments: credit card payments, micropayments, and digital coins.

Cryptographically, the most challenging task is the design of **digital coins**. For every payment system mentioned above we have the requirement that the payment token has to be **unforgeable**. Usually, we achieve this goal by using **digital signatures** of the bank to validate a payment token. But in contrast to credit card payments and micropayments for digital coins we have an additional requirement - the **anonymity** of the payment token, i.e. the bank is not able to link a purchase to a certain customer.

In 1982, D. Chaum presented the notion of **blind digital signatures** that offer the possibility to design electronic coins, [5]. The bank blindly signs a set of data chosen by the customer which guarantees both the unforgeability of the coins and their anonymity, since the bank gets no information about the data it signed.

But blind signatures solve only half of the problem: since digital data can be copied, a customer can spend a valid coin several times (**double-spending**) if the deposit of coins is not done **on-line**. In order to validate each coin on-line, the vendor has to contact the bank in every purchase. In the interests of efficiency, this is highly undesirable. Therefore, we restrict our attention to **off-line systems**, i.e. those in which the vendor has to check the validity of coins without contacting the bank.

Since no vendor can distinguish an original coin from its copy, no cryptographic mechanism can prevent double-spending. In 1988 Chaum, Fiat, and Naor overcame the problem by presenting a **double-spending detection** mechanism, see [4]. A coin is constructed in a manner which allows its owner, the customer, to spend it anonymously once, but reveals her identity if she spends it twice.

From a theoretical point of view this solution is quite elegant, but it is quite impractical. Since 1988 a number of more efficient double-spending detections have been published, but the most efficient of them is still the system presented by S. Brands, see [3].

Nonetheless, all of these mechanisms are able only to **detect** a fraud but not to prevent one. A way to prevent the customer physically from copying his coins is to store essential parts of them in a tamper-resistant device called an **observer**. An example for this kind of tamper-resistant device needed for a payment system is a conventional smart card chip.

Another drawback of the coin systems described so far is the perfect anonymity they provide. A digital coin system is called **perfectly anonymous** if the bank's views of withdrawal and of payment are independent (in a stochastic sense). In practice this implies that the bank can by no means trace a customer or her coins, and is unable to determine which customer performed any given payment. Although perfect anonymity is a theoretically nice property it is not desirable in practice. In [16], [1], [13] the authors describe attacks on a perfect anonymous system: perfect anonymity makes **blackmailing** or **money laundering** possible. In practice, we need some **anonymity revocation** mechanism to prevent criminal activities or to trace the criminals. Obviously, another party different from the bank has to apply this mechanism in order to protect the honest participants within the system. Since customers and the bank have to trust this party in revealing only suspicious identities we call this new party a TTP (= **trusted third party**). Digital coin systems that implement this kind of anonymity revocation method are called **fair**.

Our approach is to modify a coin system with observers so that a TTP can revoke the anonymity of customers in case of need. Since crucial parts of the coins are stored by the observer and cannot be read by their owner, it is not obvious how to design a coin system which simultaneously provides both physical defence (by an observer) against double-spending and the fairness of coins. Based on the works of [9] and [3] we develop a coin system which provides both fairness and observer and prove the security of our construction (provided that the basic primitives are secure). This shows that both concepts can be implemented in one protocol without loss of security. By using observers we can achieve fairness in a more efficient way than [9].

The paper is organized as follows: In section 2 we briefly sketch a coin system with observer, whilst in section 3 we discuss fairness. In section 4 we present a system that features both observers and fairness, and the generic framework is given in section 5.

2 Coins in Wallets with Observers

There are several proposals in literature to model the properties of conventional coins for digital purposes, e.g., [4], [10], [3], [7], [11], [14]. A digital coin system as a conventional one has to provide non-forgery, off-line verification, and untraceability. In detail,

- The unforgeability of digital coins is guaranteed by a digital signature of the bank. Hence, only the bank can generate coins, but everyone is able to verify the correctness of the signature. Forging a coin is as difficult as breaking the digital signature scheme, thus, in general we have computational security.
- To achieve the untraceability of coins the bank computes a "blind" signature instead of a conventional one. This means that the customer and the bank generate a set of data and a corresponding signature so that the bank cannot reconstruct the coin after the withdrawal.

The central problem with off-line coins is the possibility of double-spending, i.e. the lack of freshness of the coins. Although a merchant can verify the validity of the bank's signature, he is not able to decide whether or not a coin has been double-spent. Since the coins provide for the customers' unconditional anonymity, not even the bank knows who double-spent the coin.

The basic approach to solving this problem came from Chaum, Fiat, and Naor, [4], by including the customer's identity within the coin in a way that enables the bank to compute this identification number after a double-spending (and only in that case!). This implies that a payment cannot simply involve sending a coin to the vendor, but also requires the customer has to reveal a part of her identity.

Note, that the coin systems proposed in [4] and [11] cannot **prevent** double-spending, but instead **detect** it afterwards. Since no one can prevent the customer from making copies of the information stored in her computer, there is no other way to deal with the problem of double-spent coins cryptographically.

Accordingly, coins should not be stored by the customer's computer, but by a tamper-resistent hardware device which **erases** the coins after the customer spent them. To assure the customer's anonymity the tamper-resistant device is controlled by the customer's computer. This concept is often called an „observer "and was presented by Chaum and Pedersen in [6]. The observers are distributed by the bank.

To guarantee the prevention of double-spending the bank has to be sure that the observers cannot be tampered with by the customers. This is twofold:

- It must not be possible for the customer to construct valid coins from the information sent by the observer during the communication.
- It must be impossible to physically extract information from the observer which enables the customer to construct valid coins.

On the other hand the customer's anonymity should not be compromised by the observer. Even if the observer is returned to the bank, the bank must not be able to link the customer to her payments. The approach used to achieve this goal is to let the customer control all communication between the observer and the world. This enables the customer to control the information provided by the observer.

For our fair off-line coin scheme we restrict this model slightly. In our model the observer cannot be returned to the bank. This restricts the bank's ability to select the data stored in the observers memory. In practice this can be done by destroying the observer before it is returned to the bank.

The use of an observer is a kind of first line of defence. If the customer cannot manipulate the device the observer can prevent double-spending. If the customer succeeds in tampering with the observer, the double-spending detection identifies the customer afterwards. The customer has to break the observer physically and the electronic cash system mathematically to cheat the bank.

To avoid the attacks described in [2], the observer's internal memory has to be protected by adding some error detection bits. We demonstrate the double-spending detection and the use of an observer in the system of [3].

Throughout the paper, we denote the actors as follows. A user is denoted by $\mathcal{C}$, $\mathcal{B}$ is the bank, $\mathcal{O}$ is an observer, and the vendor is denoted by $\mathcal{V}$.

2.1 Preparations of the Bank

The bank chooses a group G of prime order q, three generators g, g_1, g_2 of G, a secret key $x \in \mathbb{Z}_q$ and two one-way hash functions $H : G^5 \to \mathbb{Z}_q$ and $H_0 : G^2 \times ID \times TIME \to Z_q$ where ID is the set of vendor identification numbers. The bank computes its public key $h := g^x$. All computations are done in G.

2.2 Opening an Account

To open an account the customer authenticates herself to the bank. She secretly and randomly chooses $u_1 \in \mathbb{Z}_q^*$ and sends $g_1^{u_1}$ to the bank. Furthermore, she proves in zero-knowledge her knowledge of u_1. The bank chooses $o_1 \in \mathbb{Z}_q^*$, stores it in the tamper-resistant device, and hands the observer $\mathcal{O}$ over to the customer. The observer computes $A_\mathcal{O} := g_1^{o_1}$ and transmits $A_\mathcal{O}$ to the customer. The bank stores $I := A_\mathcal{O} g_1^{u_1}$ together with the customer's personal data. Note, that the customer does not know the representation of I.

2.3 Withdrawal

In principle, the withdrawal protocol is the generation of a Schnorr-signature, [15], by the bank on a set of data known only to the customer, as shown in figure 1. The observer starts with choosing o_2, computing $B_\mathcal{O}$ and sending $B_\mathcal{O}$ to the user, while the bank chooses w, computes A and B and sends both values to the user.

The user performs some blinding operations and sends a blinded challenge c back to the bank. The bank signs the challenge using its private key and sends the signature back to the user. The user can unblind the bank's signature by computing r'.

$\mathcal{O}$		$\mathcal{C}$		$\mathcal{B}$
				$w \in_\mathcal{R} \mathbb{Z}_q^*$
$o_2 \in_\mathcal{R} \mathbb{Z}_q^*$				$a \leftarrow g^w$
$B_\mathcal{O} \leftarrow g_1^{o_2}$	$\xrightarrow{B_\mathcal{O}}$	$s \in_\mathcal{R} \mathbb{Z}_q^*$	$\xleftarrow{a,b}$	$b \leftarrow (Ig_2)^w$
		$A \leftarrow (Ig_2)^s$		
		$z' \leftarrow z^s$		
		$x_1, x_2, e \in_\mathcal{R} \mathbb{Z}_q^*$		
		$B \leftarrow g_1^{x_1} g_2^{x_2} A_\mathcal{O}^{se} B_\mathcal{O}$		
		$u, v \in_\mathcal{R} \mathbb{Z}_q^*$		
		$a' \leftarrow a^u g^v$		
		$b' \leftarrow b^{su} A^v$		
		$c' \leftarrow \mathcal{H}(A, B, z', a', b')$		
		$c \leftarrow c'/u \bmod q$	$\xrightarrow{c}$	
		$g^r \stackrel{?}{=} h^c a$	$\xleftarrow{r}$	$r \leftarrow cx + w \bmod q$
		$(Ig_2)^r \stackrel{?}{=} z^c b$		
		$r' \leftarrow ru + v \bmod q$		

Figure 1: Withdrawal with Observer

2.4 Purchase

In a purchase, the customer sends the coin (A, B, a', b', z', r') to the vendor. The customer and the vendor perform a challenge-response-protocol: the vendor generates a challenge depending on the coin, the vendor's identification number and the time. The customer proves, with the help of the observer, that she knows a representation of A and B with respect to g_1 and g_2.

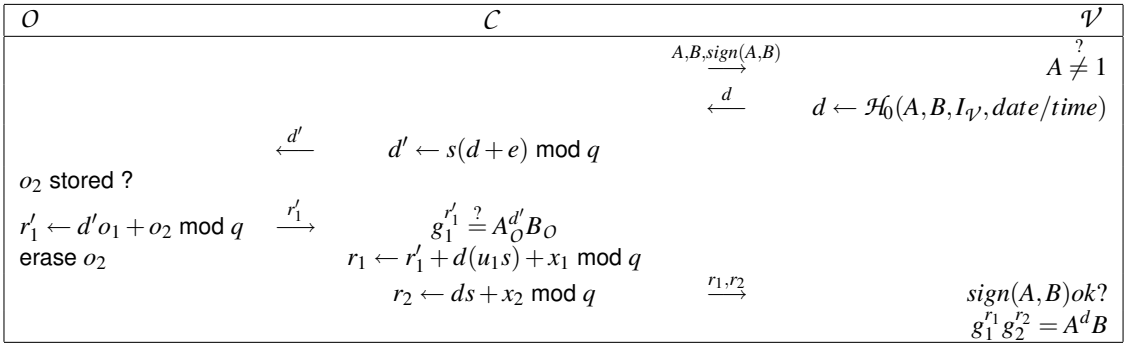

Figure 2: Purchase with Observer

It is easy to prove that the customer cannot compute o_1 and o_2 from the information sent by the observer, under the assumption that she cannot compute discrete logarithms in G. This implies, that the customer is not able to perform the proof of knowledge without the interaction with the observer. And since the observer erases o_2 after the purchase, the customer cannot double-spend the coin.

2.5 Deposit

To deposit a coin, the merchant sends the coin and the transcript of the challenge-response-protocol to the bank. The bank checks the correctness of the coin's signature, of the vendor's challenge d and of the customer's responses to the challenge. If everything is correct, the bank checks in its database whether the coin has already been deposited. If it has not, the bank stores the coin, the vendor's challenge, and the customer's responses.

If it is, the bank compares the values of the challenges and the responses. If the challenges are the same, the vendor has tried to cheat by depositing the same coin twice. Thus, the bank rejects the deposit. If the challenges differ, so do the customer's responses, which enables the bank to compute the customer's identification number. Let d and $\tilde{d}$ be the merchants' challenges and r_1, r_2 and $\tilde{r_1}$, $\tilde{r_2}$ the customer's responses:

$$\begin{aligned} g_1^{(r_1-\tilde{r_1})(r_2-\tilde{r_2})^{-1}} &= g_1^{(r'_1+dsu_1+y_1-\tilde{r'_1}-\tilde{d}su_1-y_1)(ds+y_2-\tilde{d}s-y_2)^{-1}} \\ &= g_1^{o_1+u_1} \\ &= I \end{aligned}$$

3 Fairness

Perfect anonymity protects the privacy of customers, in the sense that the bank is not able to decide whether or not the customer performed any given payment. Here, "perfect" refers to the fact that even a bank with unlimited computational power is not able to trace coins, i.e., the privacy protection is not based on any complexity assumptions.

To implement an anonymity-revocation mechanism one has to sacrifice the requirement of perfect anonymity. Frankel, Tsiounis, and Young showed in [8] that anonymity in a fair coin system can only be computational.

Furthermore, [8] shows that anonymity revocation has two different aspects:

- **Identity tracing**: after a payment, the identity of the payer is revealed.
- **Coin tracing**: after a withdrawal, the coins or some crucial parts of them are revealed.

Identity tracing is needed if the bank receives suspicious money and wishes to know who spent it. This can be necessary in case of money laundering. In case of blackmailing the bank has a protocol transcript of a suspicious withdrawal. The task is to find the withdrawn coins, in order to publish them to prevent payments with these coins. Therefore, in this case coin tracing is needed.

There are many proposals as to implement identity and coin tracing, [1] [8], [9], [13]. The most efficient solution for a fair coin system is presented in [9]. It is based on Brands' coin system, and guarantees fairness by using encryptions of the customer's identity and parts of the coin. The customer has to prove the correctness of her encryptions by zero-knowledge proofs of knowledge.

We will give a brief overview of the protocols: In principle, three additional non-interactive zero-knowledge proofs for the equality of discrete logarithms are performed additionally: two during the withdrawal to provide the bank with the encryption of a part of the withdrawn coin to ensure coin tracing and a third one during the payment to ensure identity tracing.

Notation: $EqLog[(A,a),G_1,(B,b),G_2]$ denotes the zero-knowledge proof presented in figure 3. The non-interactive variant is constructed by the standard technique of replacing the random challenge c by a hash value $c := H(A,B,A',B',a,b,G_1,G_2)$.

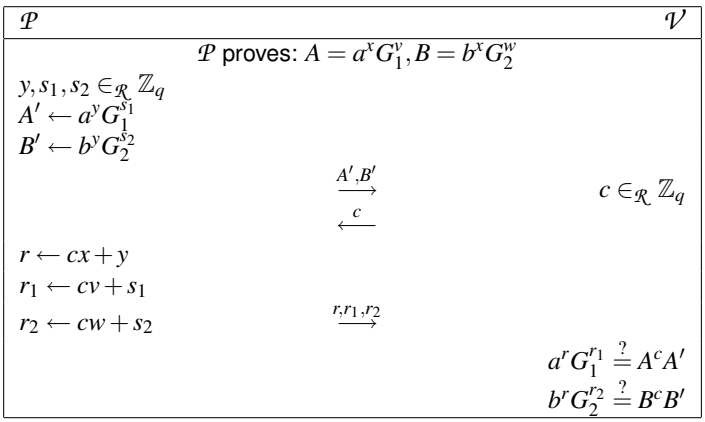

Figure 3: Proof of Equality of Logarithms

3.1 Initialization of the bank

As in the Brands' system the bank chooses a group G of prime order q. Generators g, $g_1,\ldots,g_4$ of G, q, and hash functions $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots$ are published.

The bank's secret key is x_B and the corresponding public keys are $h := g^{x_B}, h_1 := g_1^{x_B}, h_2 := g_2^{x_B}, h_3 := g_3^{x_B}$.

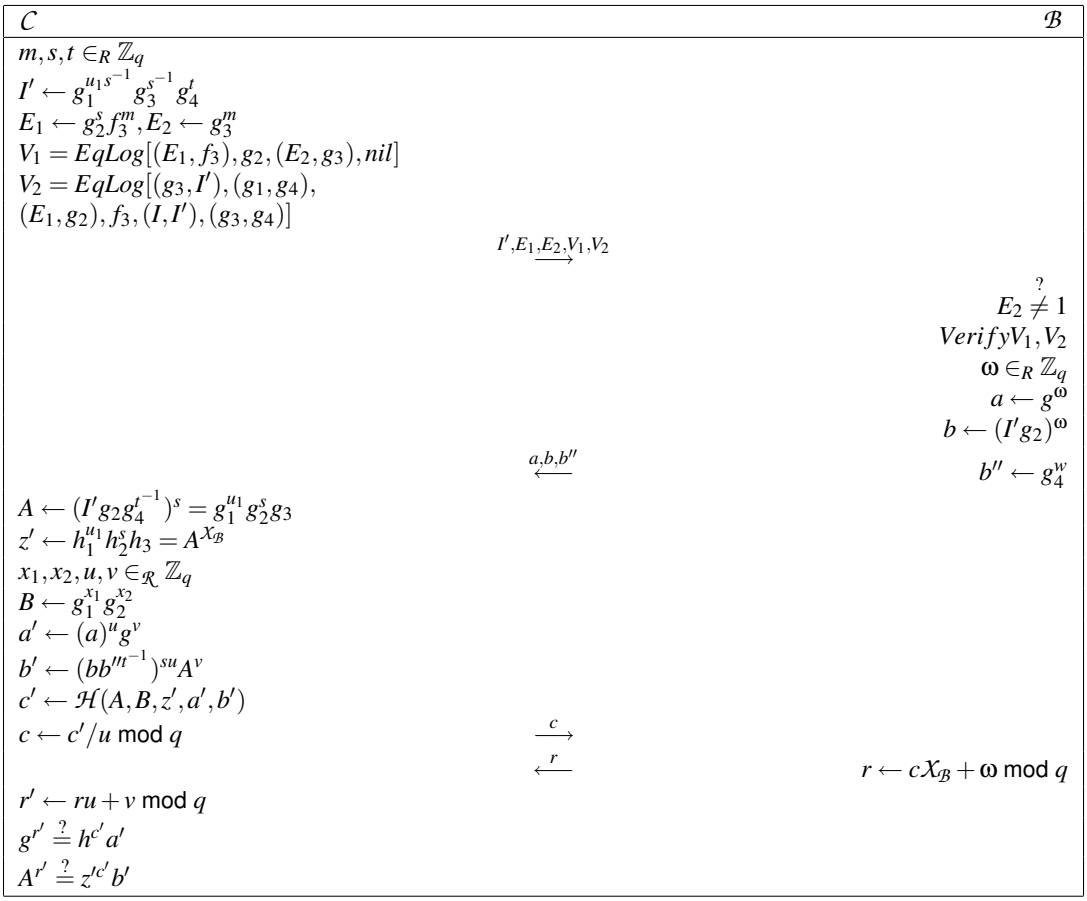

Figure 4: Fair Withdrawal

3.2 Initialization of the TTP

The TTP chooses a secret key x_T and publishes $f_2 := g_2^{x_T}, f_3 := g_3^{x_T}$.

3.3 Opening an account

There is no difference from Brands' system: the customer chooses $u_1 \in \mathbb{Z}_q$, $g_1^{u_1} g_2 \neq 1$, and computes $I := g_1^{u_1}$. She proves in zero-knowledge the knowledge of u_1.

3.4 Withdrawal

The new protocol, presented in figure 4, differs in two ways from the basic system. Firstly, the customer has to transmit an ElGamal encrypted part of her coin, i.e., g_2^s. Secondly, in order to force her to use this term in the following withdrawal, a new value I' is needed. I' encodes the customer's identity. In V_1 the customer proves the correctness of this encoding, in V_2 the correctness of the ElGamal encryption. The combination ensures that g_2^s can only be used by the participating customer.

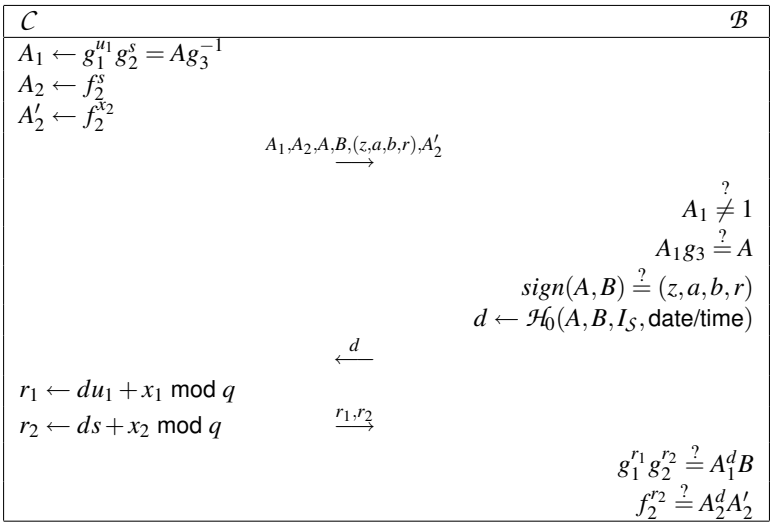

Figure 5: Fair Purchase

3.5 Purchase

The purchase protocol, presented in figure 5, contains an additional proof of equality of logarithms. The customer encrypts her identity under the public key of the TTP and proves that the encryption is a correct ElGamal encryption, and that the plaintext matches the identity number encoded in A of her coin.

3.6 Deposit

As in the basic system, the vendor deposits the coin by sending his protocol transcript of the payment to the bank.

Remark Note that the TTP does not register customers. Customers only need the TTP's public keys.

3.7 Analysis

We have to consider three aspects of security: (i) security for the bank, i.e., the unforgeability of coins, (ii) fairness, i.e., the TTP is able to trace identities and coins, (iii) security for the customers, i.e., the privacy protection. For a detailed discussion see [3] and [9].

(i) Unforgeability and unreusability are the same as in the basic system of Brands, [3].

(ii) The TTP can trace identities since it gets an ElGamal-Encryption A_1, A_2 of the customer's identity I. The correctness of the encryption is proven using a zero-knowledge protocol, hence the probability that the customer cheated without detection is negligible. The TTP can trace coins because it gets the ElGamal encryption E_1, E_2 of $g_2^s = A \cdot I^{-1}$.

(iii) The customers' anonymity is based on the Diffie-Hellman-Decision assumption. The proof is performed in the random oracle model by a simulation technique. For details see [9].

4 Fair Coins in Wallets with Observers

To combine both concepts we have to carefully implement an observer in the fair system. In the previous section we saw that fairness can be achieved by some zero-knowledge proofs which increase the computational effort of the protocols. We now show how to eliminate the zero-knowledge proofs by replacing them with a signature of the observer.

4.1 Initialization of the Bank and the TTP

In addition to the system setup presented in 3.1 another generator $g_5 \in G$ and two more hash functions $\mathcal{H}_2, \mathcal{H}_3$ are published. The hash function $\mathcal{H}_3$ and the generator g_5 are used for construction and verification of a Schnorr-signature generated by the observer.

4.2 Opening an Account

In addition to the account opening in 3.3 a signature key for the observer is created. A randomly chosen value x_O is stored in its memory. x_O which is unknown to the customer is the private key of the observer. The public key $g_5^{x_O}$ is stored by the bank and by the customer.

4.3 Withdrawal

The new withdrawal protocol presented in figure 6 uses a different method to achieve fairness than the one shown in 3.4. Instead of using zero-knowledge proofs to ensure tracing mechanisms, coin tracing is now guaranteed by the observer. Compared to the protocol in 3.4 the ElGamal encryption (E_1, E_2) and the value I' are computed by the observer and signed afterwards using the Schnorr-signature scheme.

The signature σ_O guarantees the correct construction of I' and the ElGamal encryption (E_1, E_2). To avoid the use of a subliminal channel between observer and bank as described in [12] all random values are chosen by the customer. Thus, the customer gets to know the observer's private key if observer and bank use the signature as a subliminal channel. The customer also verifies the signature as well as the correct construction of I' and the ElGamal encryption (E_1, E_2) before transmitting these values to the bank.

4.4 Purchase

The view of the vendor does not differ from that within the basic protocol. Since the customer does not know the discrete logarithm of her identity number I she has to perform the proof V_3, which ensures that owner tracing can be done by the trusted third party, with the help of the observer. The protocol is presented in figure 7.

We omit the deposit since it is the same as in the basic system.

4.5 Efficiency

When implemented using a subgroup of order q of $\mathbb{Z}_p^*$ we save 17 modular exponentiations compared to [9], which is about a third of the modular exponentiations made during withdrawal.

4.6 Discussion

Again, we have to consider three aspects in order to show the security of our construction: (i) The unforgeability and unreuseability are proven by a standard simulation technique. (ii) The tracing algorithms of our construction are the

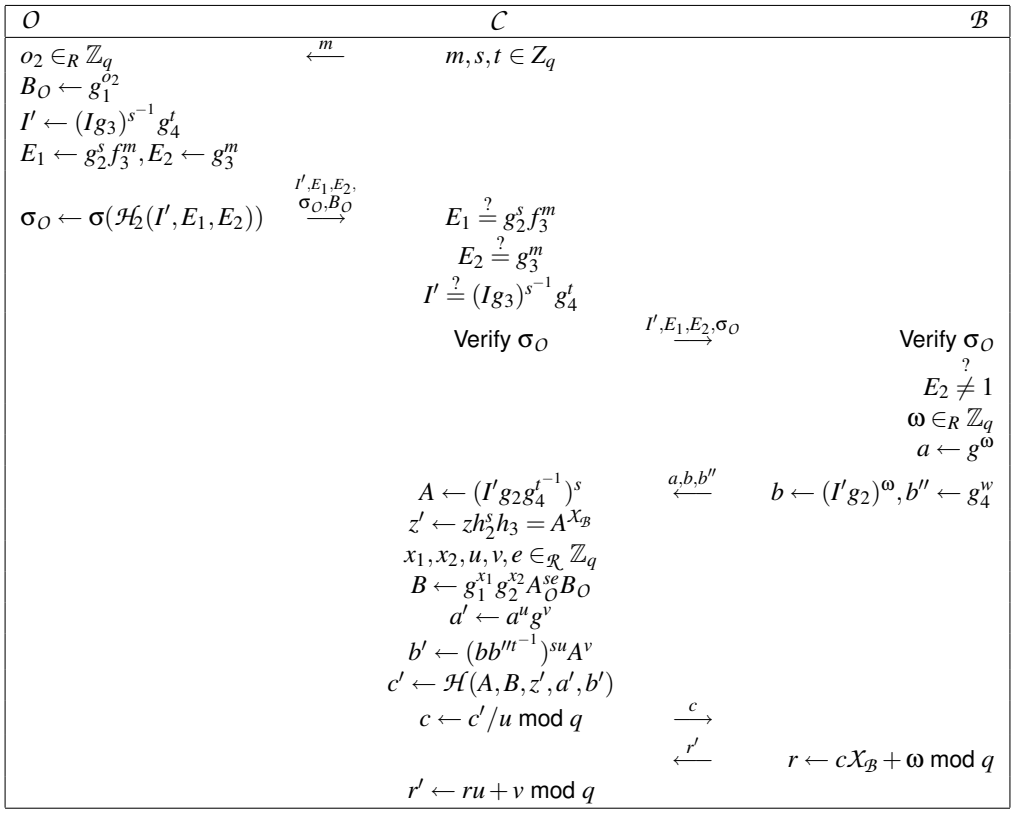

Figure 6: Fair Withdrawal with Observer

same as in [9] since the protocols views of the bank and the vendor remain the same. (iii) The most interesting part is the customers' privacy protection.

Theorem 1 *Under the Diffie-Hellman-Decision assumption and assuming that the observer is not returned to the bank the customers' privacy is computationally protected in the random oracle model.*

Proof
We show that a machine $\mathcal{M}$ capable of deciding which coin originated from which withdrawal can break the semantic security of the ElGamal encryption and hence breaks the Diffie-Hellman-Decision assumption. Let $i, j \in \{0,1\}$, $i = 1 - j$. $\mathcal{M}$ gets $\mu^{(i)} := g_2^{s^{(i)}}$, $\mu^{(j)} := g_2^{s^{(j)}}$, $s^{(i)}$, $s^{(j)}$ and the encryption $(E_1^{(0)}, E_2^{(1)})$ of $\mu^{(0)}$ and $(E_1^{(1)}, E_2^{(1)})$ of $\mu^{(1)}$ as an input. Choosing $s^{(i)}$ and $s^{(j)}$ uniformly on $\mathbb{Z}_q$ implies a uniform distribution of $\mu^{(i)}$ and $\mu^{(j)}$ on G. $\mathcal{M}$ has to find out whether i equals 0 or 1. $\mathcal{M}$ simulates two withdrawals and two payments:

- $\mathcal{M}$ chooses x_B and two random identity numbers $u_1^{(i)}, o_1^{(i)}$ and $u_1^{(j)}, o_1^{(j)}$.
- $\mathcal{M}$ knows $s^{(i)}$ and $s^{(j)}$. Hence $\mathcal{M}$ can compute:

$$\begin{aligned} A_1^{(i)} &:= g_1^{u_1^{(i)} + o_1^{(i)}} g_2^{s^{(i)}} & A_1^{(j)} &:= g_1^{u_1^{(j)} + o_1^{(j)}} g_2^{s^{(j)}} \\ A_2^{(i)} &:= f_2^{s^{(i)}} & A_2^{(j)} &:= f_2^{s^{(j)}} \\ A^{(i)} &:= A_1^{(i)} g_3 & A^{(j)} &:= A_1^{(j)} g_3 \end{aligned}$$

- $\mathcal{M}$ randomly chooses $x_1^{(i)}, x_2^{(i)}, d^{(i)}$ and $x_1^{(j)}, x_2^{(j)}, d^{(j)}$.

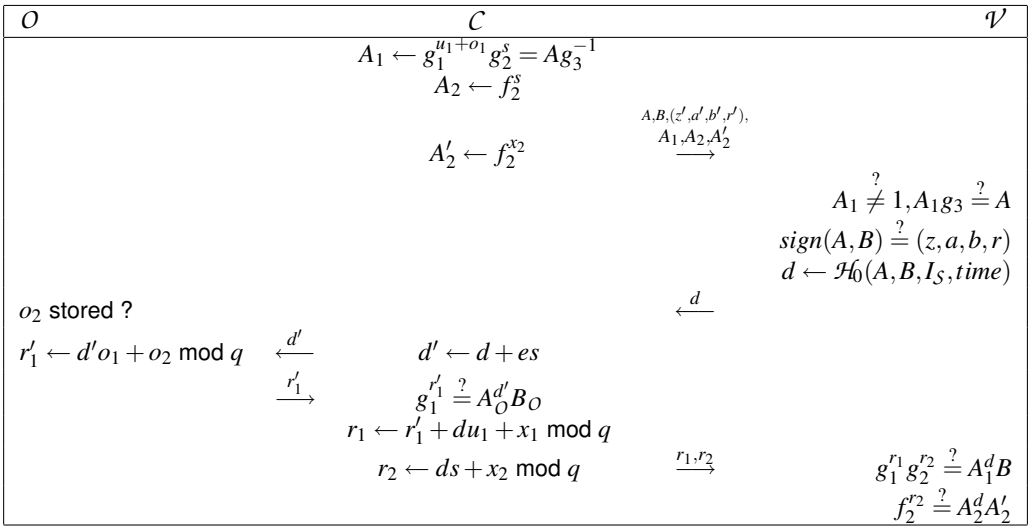

Figure 7: Fair Purchase with Observer

- $\mathcal{M}$ simulates the payments: $B^{(i)} := g_1^{(x_1^{(i)}} g_2^{x_2^{(i)}} A_1^{-d^{(i)}}$ and $B^{(j)} := g_1^{(x_1^{(j)}} g_2^{x_2^{(j)}} A_1^{-d^{(j)}}$. Since all computations are done in the random oracle model, $\mathcal{M}$ can set $\mathcal{H}(A^{(i)},B^{(i)},ID_H^{(i)}) =: d^{(i)}$ and $\mathcal{H}(A^{(j)},B^{(j)},ID_H^{(j)}) =: d^{(j)}$. $\mathcal{M}$ chooses $r_1^{(i)} := x_1^{(i)}, r_2^{(i)} := x_2^{(i)}$ and $r_1^{(j)} := x_1^{(j)}, r_2^{(j)} := x_2^{(J)}$ as the responses to these "challenges".
- $\mathcal{M}$ randomly chooses $o_2^{(i)}, d'^{(i)}, r_1'^{(i)}$ and $o_2^{(j)}, d'^{(j)}, r_1'^{(j)}$. Therefore, $\mathcal{M}$ gets two payment views:

$$A^{(i)}, A_1^{(i)}, A_2^{(i)}, B^{(i)}, ID_H^{(i)}, d^{(i)}, o_2^{(i)}, r'^{(i)}_1, r_1^{(i)}, r_2^{(i)}$$
$$A^{(j)}, A_1^{(j)}, A_2^{(j)}, B^{(j)}, ID_H^{(j)}, d^{(j)}, o_2^{(j)}, r'^{(j)}_1, r_1^{(j)}, r_2^{(j)}$$

- Analogously, $\mathcal{M}$ generates the withdrawal views. (Since $\mathcal{M}$ knows the "secret" key x_B it can compute valid signatures.)
- Since $\mathcal{M}$ can link payments and withdrawals (by assumption) it can determine the value of i and j - hence,$\mathcal{M}$ can distinguish ElGamal encryptions contradicting the Diffie-Hellman-Decision assumption.

This completes the proof and shows that the anonymity of the customers is computationally protected.

5 Generic Coin System Framework

In this section we present a framework for fair wallets with observer, where basically any blind signature scheme can be used as a underlying building block.

5.1 Setup

The bank $\mathcal{B}$ chooses an appropriate blind signature scheme σ_B, generates its secret key x_B and publishes the corresponding public key y_B as well as all system parameters.
The trusted third party chooses two signature schemes σ, σ',a probabilistic encryption scheme φ, generates the secret key x_T of the encryption scheme φ and publishes the corresponding public key y_T together with all system parameters, including the description of σ and σ'.

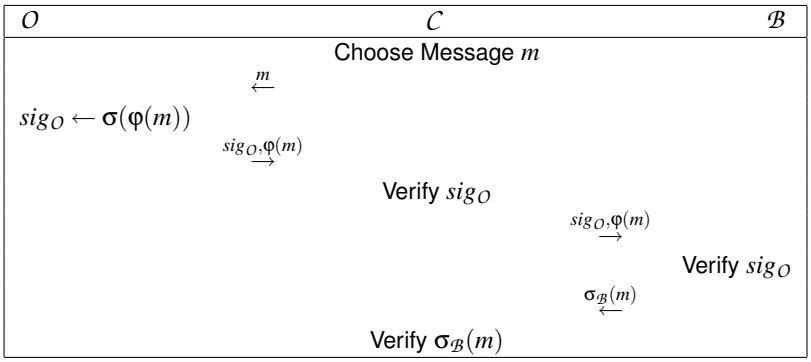

Figure 8: Withdrawal protocol in the generic model

5.2 Account Opening

To open an account, the TTP issues an observer $\mathcal{O}$ to the customer. In $\mathcal{O}$'s memory two secret keys x_O, x'_O for the signature schemes σ resp. σ′ are stored, which are unknown to the TTP. The corresponding public keys are published.

5.3 Withdrawal

To withdraw a coin from her account, customer $\mathcal{C}$ and the bank $\mathcal{B}$ perform the following protocol:

Step 1: $\mathcal{C}$ chooses a random message m and sends m to $\mathcal{O}$.

Step 2: $\mathcal{O}$ encrypts m under the public key y_T and signs the encrypted value with the signing key x_O and sends the signature $\sigma_O \leftarrow \sigma(\varphi(m))$ together with the encryption $\varphi(m)$ to $\mathcal{C}$.

Step 3: $\mathcal{C}$ verifies the signature sig_O and sends σ_O and $\varphi(m)$ to $\mathcal{B}$.

Step 4: $\mathcal{B}$ verifies sig_O and blindly signs m and sends the signature $\sigma_{\mathcal{B}}(m)$ to $\mathcal{C}$. If σ_O is a valid signature, $\mathcal{B}$ knows that the $\varphi(m)$ was correctly encrypted, which ensures coin tracing.

Step 5: $\mathcal{C}$ verifies $\sigma_{\mathcal{B}}(m)$ and obtains a signature on m, unknown to $\mathcal{B}$.

Remarks: Since $\mathcal{O}$ generated the encryption $\varphi(m)$ and signed it with σ_O the **coin tracing** mechanism is guaranteed. Note that $\mathcal{C}$ can still obtain blind signatures on coins m' different from m. But $\mathcal{C}$ cannot spend these coins, since they are not generated with the help of $\mathcal{O}$.

5.4 Payment

To pay with a valid coin $\mathcal{C}$ and $\mathcal{V}$ perform the following protocol:

Step 1: $\mathcal{C}$ sends a message to $\mathcal{O}$ that she wants to pay with coin m.

Step 2: $\mathcal{O}$ checks if m is still stored in its EEPROM. If so, $\mathcal{O}$ encrypts $\mathcal{C}$'s ID under the public key of the TTP and computes $sig'_O \leftarrow \sigma'(m, \varphi(ID))$, deletes m, and sends sig'_O to $\mathcal{C}$. If m is already erased $\mathcal{O}$ will abort the communication with $\mathcal{C}$.

Step 3: $\mathcal{C}$ verifies sig'_O and sends $sig'_O, \sigma_{\mathcal{B}}(m)$ to $\mathcal{V}$.

Step 4: $\mathcal{V}$ verifies both signatures sig'_O and $\sigma_{\mathcal{B}}(m)$.

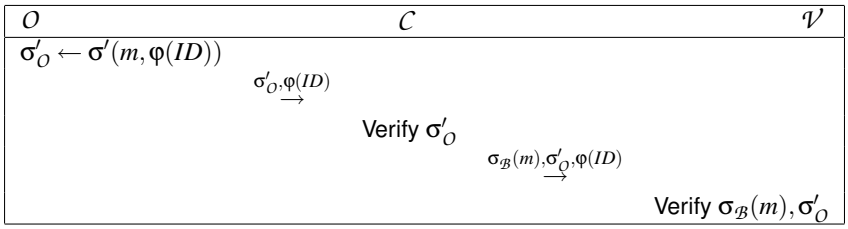

Figure 9: Payment protocol in the generic model

Remarks: The signature sig'_O on the encryption $\varphi(ID)$ generated by $\mathcal{O}$ guarantees the **owner tracing** mechanism. As mentioned above, $\mathcal{C}$ can obtain a blind signature on a different coin m' during the withdrawal protocol. The fact that the coin m is part of the message $(m, \varphi(ID))$ signed by $\mathcal{O}$, prevents $\mathcal{C}$ from successfully paying with m'.

5.5 Deposit

To deposit a coin at $\mathcal{B}$, $\mathcal{V}$ simply sends the coin m, its signature $\sigma_{\mathcal{B}}(m)$, the encrypted identity $\varphi(ID)$, and σ'_O to $\mathcal{B}$. The bank $\mathcal{B}$ will accept the coin if $\sigma_{\mathcal{B}}(m)$ is a valid signature on m and σ'_O is a valid signature on $(m, \varphi(ID))$.

5.6 On the choice of σ and σ'

Basically, the two signature schemes σ and σ' are used to guarantee coin tracing and owner tracing respectively, and to prevent $\mathcal{C}$ from spending coins that are not stored in the memory of $\mathcal{O}$.
During withdrawal one has to make sure that $\mathcal{B}$ is convinced that $\varphi(m)$ was generated by the observer $\mathcal{O}$, which was issued to $\mathcal{C}$ by TTP. Therefore it suffices to use an ordinary signature scheme for σ. Obviously, if one were to use the same key pair for σ', the anonymity of $\mathcal{C}$'s coins would be lost. So, the signature scheme σ' must preserve the anonymity of the coins. There are several ways to deal with this problem.

Pseudonyms: For each customer $\mathcal{C}$, the TTP creates a pseudonym, unknown to $\mathcal{B}$. For all pseudonyms a key pair is generated. The secret keys are stored in the customers' observers and the corresponding public keys are made public. Due to this fact, both $\mathcal{V}$ and $\mathcal{B}$ can verify whether or not sig'_O was generated by an observer, but $\mathcal{B}$ cannot link sig'_O to the customer who withdrew the coin. Yet, $\mathcal{B}$ can link all coins withdrawn by the same customer.

Group key: For σ', the TTP chooses one key pair, which belongs to the group of all customers. The secret key is stored in all issued observers and the corresponding public key is published. Since there is only one public key, $\mathcal{B}$ gets no information about the customer afterwards. But if only one customer extracts the signing key from his observer, the TTP has to issue new observers and all coins stored in the old observers will become invalid.

Group signature: Instead of using an ordinary signature scheme, a group signature scheme is used for σ'. So $\mathcal{V}$ and $\mathcal{B}$ do not learn any information about $\mathcal{O}$ during payment respectively deposit and both are convinced that

- the coin was signed by an observer, and
- that owner tracing can be performed at a later point of time.

6 Practical Implementations

Since the first digital coin systems were proposed in the early 1990s there have been many reasons why practical applications using digital coins have not been manageable:

- First of all, most of the mobile devices and especially smart cards were not able to perform the necessary mathematical operations in a reasonable amount of time.
- Digital coins systems make heavenly use of public key infrastructures that have not been established at that point in time.
- The security of digital coin systems as any other payment system implicitly relies on secure computation environments. But the work on developing trusted devices that pass independent security evaluation and certification has not started before the end of the 1990s.

The situation has dramatically changed in the recent years. First of all, the most recent digital coin systems allow for the usage of elliptic curve cryptography which is far more efficient than classical RSA cryptography. Especially, the coin system presented by S. Brands which is not only the most efficient coin system today but can also be implemented using elliptic curves which significantly increases the efficiency of the system.

But more important are the technical improvements of the last ten years. Mobile devices and smart cards can perform modular exponentiations in only a few milliseconds allowing for the efficient generation of signatures especially on elliptic curves.

Furthermore, the growing area of smart card supported security applications lead to an extensive analysis and better understanding of hardware and system security which resulted not only in the establishment of public key infrastructures but also in the development of evaluation and certification schemes for hardware and software like the Common Criteria. Since trust is most important in payment systems these efforts and changes are essential for implementing digital coins.

All these things considered, all technical prerequisites for digital coin systems are in place today.

7 Conclusion

We presented a digital coin system, which provides a physical defense as primary protection against double-spending by using a dedicated tamper-resistent device, supplemented by a well-known double-spending detection mechanism which can be invoked in the event of an observer being compromised by its owner.

Furthermore, our system offers anonymity revocation, i.e., identity and coin tracing mechanisms.

We proved that both concepts, carefully implemented, do not interfere with each other in regard to security, i.e., our construction is as secure as its building primitives.

Further we presented a generic model for fair wallets with observer, where basically any blind signature scheme can be used as an underlying building block. This generic approach offers the necessary flexibility for practical implementations since one can combine the most efficient cryptographic primitives to build the coin system.

We argued that all technical pre-conditions for the real world application of digital coins are in place.

Given our results for hardware protection and protection against criminal abuse of digital coin systems we conclude that all practical and theoretical methods to enable the implementation and usage of digital coin systems are available today.

References

[1] E. Brickell, P. Gemmell, D. Kravitz, “Trustee-based tracing extensions to anonymous cash and the making of anonymous change“, *Symposium on Distributes Algorithms (SODA),* 1995

[2] D. Boneh, R.A. DeMillo, R.J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults", *Proc. of Eurocrypt '97*, Lecture Notes in Computer Science 12¡33, Springer Verlag

[3] S. Brands, "Untraceable off-line cash in wallets with observers", *Proc. of Crypto '93*, Lecture Notes in Computer Science 773, Springer-Verlag

[4] D. Chaum, A. Fiat, M. Naor, "Untraceable Electronic Cash", *Proc. of Crypto '88,* Lecture Notes in Computer Science 403, Springer-Verlag

[5] D. Chaum, "Blind signatures for untraceable payments", *Proc. of Crypto '82*, Plenum Publishing, New York 1982

[6] D. Chaum, T.P. Pedersen, "Wallet Databases with Obsersers", *Proc. of Crypto '92,* Lecture Notes in Computer Science 740, Springer-Verlag

[7] N. Ferguson, "Extensions of Single-term Coins", *Proc. of Crypto '93*, Lecture Notes in Computer Science 773, Springer-Verlag

[8] Y. Frankel, Y. Tsiounis, M. Yung, "Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash", *Proc. of Asiacrypt '96*, Lecture Notes in Computer Science 1163, Springer-Verlag

[9] Y. Frankel, Y. Tsounis, M. Yung, "Fair Off-Line e-Cash made easy", *Proc. of Asiacrypt '98*, Lecture Notes in Computer Science, Springer-Verlag

[10] M. Franklin, M. Yung, "Secure and efficient off-line digital money ", *Proc. of Automata, Languages and Programming, ICAPL '93,* Lecture Notes in Computer Science 700, Springer-Verlag

[11] T. Okamoto, K. Ohta, "Universal Electronic Cash", *Proc. of Crypto '91,* Lecture Notes in Computer Science 576, Springer-Verlag

[12] G.J. Simmons, "The subliminal Channel in Digital Signatures", *Proc. of Eurocrypt '84*, Lecture Notes in Computer Science 209, Springer-Verlag

[13] M. Stadler, J.-M. Piveteau, J. Camenisch, "Fair Blind Signatures", *Proc. of Eurocrypt '95,* Lecture Notes in Computer Science 921, Springer-Verlag

[14] T. Sander, A. Ta-Shma, "Auditable, Anonymous Electronic Cash", *Proc. of Crypto '99,* Lecture Notes in Computer Science 1666, Springer-Verlag

[15] C. Schnorr, "Efficient signature generation by smart cards", *Proc. of Crypto '89, Lecture Notes in Computer Science,* Springer-Verlag

[16] B. von Solms, D. Naccache, "On blind signatures and perfect crimes", *Computers and Security,* 11(6), October 1992