ECOLE NORMALE SUPERIEURE

Efficient Scalable Fair Cash with Off-line Extortion Prevention

Holger PETERSEN
Guillaume POUPARD

LIENS - 97 - 7

Département de Mathématiques et Informatique

CNRS URA 1327

Efficient Scalable Fair Cash with Off-line Extortion Prevention

Holger PETERSEN
Guillaume POUPARD

LIENS - 97 - 7

May 1997

Laboratoire d'Informatique de l'Ecole Normale Supérieure
45 rue d'Ulm 75230 PARIS Cedex 05

Tel : (33)(1) 44 32 00 00

Adresse électronique : petersen, poupard@dmi.ens.fr

Efficient Scalable Fair Cash with Off-line Extortion Prevention

Holger Petersen[1] · Guillaume Poupard
Laboratoire d'informatique
Ecole Normale Supérieure
45, rue d'Ulm, F-75005 Paris
E-Mail: {petersen,poupard}@dmi.ens.fr

Abstract

Since the invention of blind signatures in 1982 by David Chaum, there have been many proposals to realize anonymous electronic cash using this mechanism. Although these systems offer high privacy to the users, they have the disadvantage that the anonymity might be misused by criminals in order to commit a perfect crime (without being physically present, and thus with the assurance of not being caught). The recent research focuses therefore on the realization of fair electronic cash systems where the anonymity of the coins is revocable by a trustee in the case of fraudulent users. In this paper, we describe the main characteristics of these systems and give a comparison of existing ones. The analysis allows us to propose a new efficient fair cash system which offers scalable security with respect to its efficiency. Our system is the first that prevents extortion attacks, like blackmailing or the use of blindfolding protocols under off-line payments and with the involvement of the trustee only at registration of the users. We give two applications, a highly secure one employing provable secure signature schemes for internet payments and a very efficient one for electronic purse realization.

Keywords: Electronic payment systems, anonymity revocation, electronic purse, internet payment

[1]This work has been supported by a postdoctoral fellowship of the NATO Science Committee disseminated by the DAAD.

1. Introduction

By the increasing number of participants in the worldwide computer networks, like the internet, the importance of electronic communication and electronic commerce grows rapidly. In future all kind of goods and multimedia information will be sold via networks (like teleshopping, stock information, newspapers, pictures, movies on demand, software etc.). Also financial transactions are executed through the internet (like home banking). Therefore we need secure electronic payment systems for *internet payment* to support these businesses. First approaches can be found e.g. in [BGHH95, SET 96, Waid96]. Outside the internet, *electronic purses* have been developed to replace the conventional purse [Chau87, BBCM94]. Both settings have different hardware and security requirements, which affects their design and implementation. Even if there will always be a need for non-anonymous payment systems for large amounts, for smaller payments it is required by the users that the payment system offers a certain amount of privacy (up to total anonymity), as otherwise the filing of user dossiers will become much easier as with conventional money [Chau85].

Further user requirements are *easy usable* systems, which are not harder to use than real cash, *cheap* transactions, *globally usable* payment systems which are *open* for everyone who wishes to participate, i.e. users, shops, banks and also hardware suppliers. These requirements are out of the scope of cryptography, but should also be mentioned, as the successful set-up of a payment system depends mainly on its acceptance by the users. Besides, the system should offer *flexibility* for the user at payment, i.e. it should avoid that the user has to contact the bank at each payment and also the system should be *robust* against any attack, i.e. attacks should only influence single entities but never the system as a whole. Specially, the working of the system could not be endangered by any attack, such that all users could loose their money already debited from their accounts.

For *non-anonymous* payments there have been several proposals to realize credit card models or low-value *micropayment* systems. For a survey of systems see [Hall95, AJSW96]. The concept of *anonymous* digital cash was invented 1982 by Chaum [Chau82]. The first systems based on his ideas have been designed for on-line payments [Chau87, BüPf89]. By the invention of the cut-and-choose methodology it was possible to design off-line systems, among them [ChFN88, OkOh91, FrYu92]. In 1993 the first systems not using the cut-and-choose paradigm were designed that offer computational anonymity for the users [Bra93b, Ferg93], i.e. withdrawn coins are untraceable to the users. Figure 1 gives a brief classification of electronic payment systems. By little amounts we mean very small amounts e.g. less than $1 payments, by small amounts we are thinking of amounts e.g. between $1 and $100 and by large amounts of all that are inferior to $100.

Unfortunately, this perfect anonymity might be misused by criminals to commit a *perfect crime* [SoNa92]. Blackmailing of coins, money laundry, extortion or theft of secret keys or the use of blindfolded protocols with the bank or trustees have been considered as possible attacks [SoNa92, BrGK95, JaYu96]. In order to prevent these threats the payment systems should provide *anonymity revocation* mechanisms, that allow the tracing of coins in any of the above scenarios by an authorized third party, the *trustee*, or a set of these parties. The first systems preventing blackmailing and money laundry have been proposed by [BrGK95, StPC95]. Since then there have been several proposals [CaPS95, CaPS96, M'Raï96, FuOk96, JaYu96] to prevent these attacks. All schemes require the participation of the trustee in the opening of an account or even in

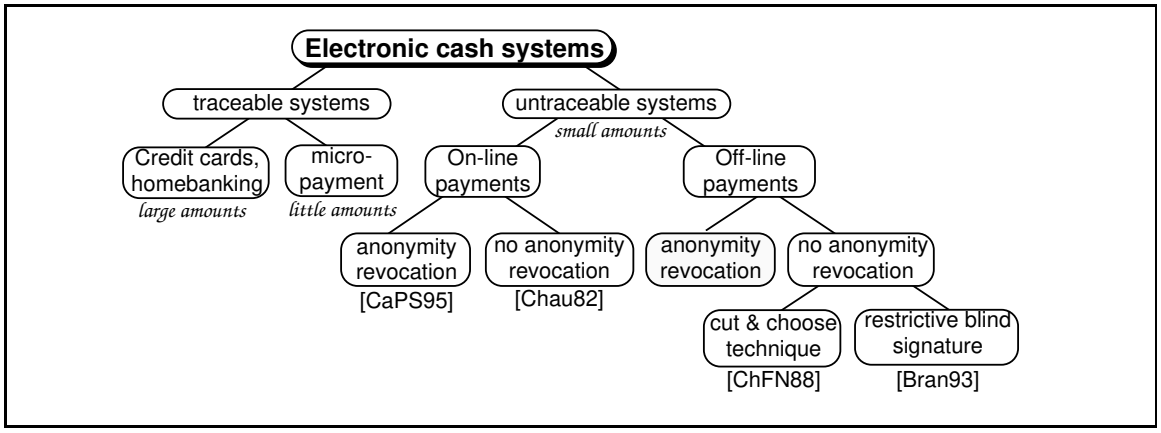

Figure 1: Classification of electronic payment systems

the withdrawal of coins. The only two systems that don't require trustee participation except for initialization of the system and for anonymity revocation have recently been proposed by [CaMS96, FrTY96]. However, they are unable to prevent extortion attacks and the use of blindfolding protocols. These attacks were only prevented in the systems of [JaYu96, FuOk96], which are therefore not as efficient as they require the trustee interaction in payment protocols. In case that one of these attacks is reported, they require an on-line payment protocol between the user, shop and trustee in order to prevent the spending of illegal coins. Figure 2 gives an overview over existing systems.

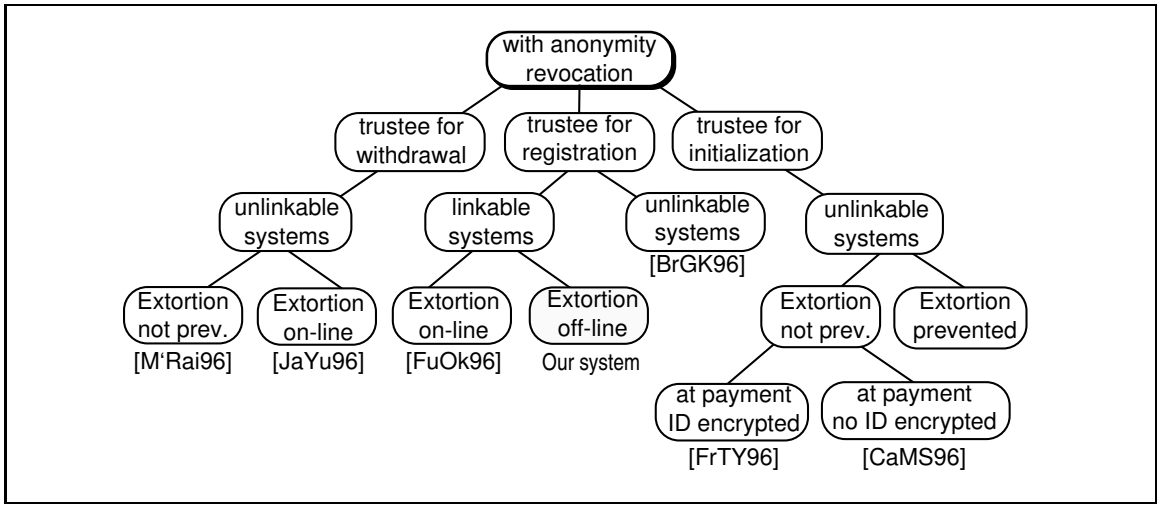

Figure 2: Classification of fair cash systems

Our results

We propose the first secure payment system that allows anonymity revocation by trustees in the case of any extortion attack under an *off-line* payment protocol. This is achieved by registering anonymous user keypairs at a trustee before withdrawal. After the registration, the trustee doesn't participate in the protocol, except for anonymity revocation after fraud. Thus our system is more efficient than the one of [JaYu96], which achieves the same extortion prevention at the cost of an interaction with the trustee at each withdrawal and under on-line payments. Although different payments under the same pseudonym are linkable, the *privacy* of payments is *scalable* by the user by increasing their number up to a maximum of one pseudonym per payment to

satisfy total unlinkability. Remarkable *benefits* of this approach are a modular, simple design that is easy to understand, to implement and to analyze with respect to security requirements, as different cryptographic tools interact as little as possible (in contrast to systems like e.g. [Bra93b, JaYu96, CaMS96]). The design results of a careful analysis of the attacks and requirements described in section two. Furthermore our system is versatile, as it allows the integration of multi-spendable and divisible coins and also supports the *challenge semantics* proposed in [JaYu96].

On the one hand, we make a clear distinction between internet and electronic purse payments, as they require different security, privacy and efficiency of the chosen protocols. On the other hand, we stress that their design can be embedded into the same framework. As our system, presented in section four, is scalable with respect to security, privacy and efficiency, it allows us to propose concrete protocols for both applications in sections seven and eight. They are quite efficient as all events, like withdrawal or payment, employ a single digital signature. We achieve the results in figure 3.

payment system	coin	pseudonym	100 coins/20 pseudo.	100 coins/100 pseudo.
internet payment	67	124	9 K	18,7 K
electronic purse	40	46	4,8 K	8,4 K

Figure 3: Size of the stored data (in byte) at the user's device

Survey of the paper

First we describe the properties of electronic payment systems and give a short classification of existing schemes in section three. In section four, we describe the ideas of our system and present the general scheme. We give a detailed security analysis of our scheme and show that it satisfies all security requirements presented in section two. After, we discuss the scalability of our system by proposing as well very efficient as secure choices for the employed cryptographic tools. Finally we focus on a secure variant of our protocol for internet payments and an efficient one for electronic purse payments.

2. Electronic Payment Systems

In this section we review the main properties of electronic payment schemes. Even if this has been well done in most previous papers (see e.g. [JaYu96]) we cannot drop this description as the design of our system follows from a precise analysis.

2.1 Events

We assume a simplified electronic cash system with just one bank, one user and one shop. Extensions to many of each are straightforward. In this setting seven main events are distinguishable:

1. *Initialisation:* Choice of system parameters and keypairs of all entities.
2. *Opening account:* The bank opens a user account and registers his personal data.
3. *Registration:* In the pseudonymous systems, the user registers at the trustee.

4. *Withdrawal:* The user withdraws digital coins from his account onto his device.
5. *Payment:* The user pays at the shop using the coins stored on his device.
6. *Deposit:* The shop deposits the digital coins at the bank and is credited accordingly.
7. *Revocation:* The trustee is able to compute either the shape of the coin from the withdrawal transcript or to compute the user's identity from the payment transcript in order to deter any perfect crime.

2.2 Models

In an electronic payment system there participate eight types of entities: These are users, banks, shops, trustees, judges, certification authorities, key directories and finally the attackers. All entities have different security requirements and also a different amount of trust in each other. We are going to describe the models we use.

Trust model

In the trust model, we describe the trust the participants have in each other. First, there is a natural confidence anyone can have to some entities because of their social position :

1. All participants trust the *judge* to act according to the law,
2. All participants trust the *certification authority* to perform its task correctly,
3. All participants trust the *trustee* to give accurate information in case of dishonest behavior of a user,
4. All participants trust the *tamper resistance* of devices and the accurate work of the hardware and installed software, e.g. that amounts are displayed on the screen are the same as credited.

More trust is usually needed but is not required in all systems :

5. The users trust the *bank* and the *trustee* not to cooperate against them if they behave honestly.
6. The users trust the *bank* to be honest, i.e. the bank manage the accounts according to the contracts signed with the users. For example, the bank does not try to steal money in the accounts of their clients.

We will see how to reduce the confidence of the users in the bank and the trustee using slightly more enhanced systems.

Attacker model

In the *traditional world* exist different kind of attackers: thefts, bank robbers, blackmailers, mafiosi, kidnappers or terrorists. Their interests and methods are different but they all have to be physically involved in their crime at some moment or even for a longer period (as for kidnapping). In the setting of *electronic cash* systems some new attacks have been considered, as the *perfect crimes* first mentioned in [SoNa92] and

the *ultimate crimes* introduced in [JaYu96]. These attacks are of short duration and without physical involvement of the attacker. After their appearance, the entity acts again by his own will. Observe, that this model *excludes* crimes like kidnapping or robbery of goods, as it is obvious that cryptography cannot help to prevent them.

Communication model

We assume for the *internet payment* scheme, that all communication is transmitted via a network and thus interceptable and vulnerable to eavesdropping attacks. For the *electronic purse* scheme, we assume that the communication between the bank and user and also between the user and shop is protected against these attacks and thus confidential, as the user is physically present for withdrawal at a point of sale (POS) terminal of the bank or for payment in a shop.

2.3 Technical requirements

For an electronic payment system a secure user device is needed in order to store the user's secret information. For example, in the setting of an *internet payment* system the device can be integrated in the user's computer as a black box. In the case of an *electronic purse* it might be a tamper resistant hardware[2] integrated on a smart card. In this context the use of a secure and trustworthy device under control of the user, the *electronic wallet*, is discussed [ChP92b].

2.4 Attacks

Attacks against electronic cash systems can be classified by who is *attacking*, who is *attacked*, their *strength* (e.g. measured in the effort to prevent them) or if they are applicable either against *all* kind of systems or only against *anonymous* ones. We give a classification by the attacking entity.

Fraudulent user

- *Overspending:* A user spends coins for a value exceeding their allowed value.

Fraudulent shop

- *Impersonation:* A shop responds or deposits a coin obtained from the user several times.
- *Money laundry:* A shop obtains digital coins by an illegal action. To conceal the origin of the money he issues a fictitious bill.

Fraudulent bank

- *Tracing a user:* The bank traces the relationship between a digital coin and a user.
- *Tracing a user with help of trustee:* False conviction of trustee that a coin has been overspent. As a result a honest user is incriminated after tracing his identity for this coin.

[2] The existence of tamper resistant devices is discussed controversyly in the literature. An example of successful tampering of "tamper-resistant" chips can be found in [AnKu96].

- *Framing a user:* False accuse of overspending a coin.
- *Framing a shop:* False accuse of double deposit of a valid coin.
- *Coin forgery after overspending:* The bank generates fictitious payment transcripts for an already overspent coin in order to be reimbursed immoderately.

Fraudulent trustee

- *Framing a user:* The trustee falsely identifies a honest user. Thereby the bank might incriminate this user without justification.

Fraudulent outsider

A fraudulent outsider is an entity which might be – but is not necessarily – registered to the trustee or have a bank account.

- *Coin forgery*
  For coin forgery there are three different attacks:
  Universal forgery: An entity, knowing the public parameters and maybe old payment transcripts, forges the bank's signature scheme in order to obtain valid coins.

  One-more forgery: An entity, that participates in n (parallel) withdrawal protocols obtains $n + 1$ valid coins.

  Overspending forgery: An entity knowing several payment transcripts of an overspent coin generates transcripts for fresh coins, which he is able to deposit.
- *Eavesdropping of coins or pseudonyms:* A passive attacker eavesdrops the communication at withdrawal, payment or deposit in order to obtain spendable coins. An active eavesdropper might also act as *man-in-the-middle* [RiSh84] and modify the protocol data. The same strategy is applicable at registration to obtain signed pseudonyms.
- *Theft or extortion of coins from the user:* An attacker either steals coin transcripts from the user's device or forces him to withdraw coins from his account and to transfer them to the attacker's device, such that he can spend them later.
- *Theft or extortion of coins from the shop:* An attacker either steals transcripts of not already deposited coins from the shop's device or forces the shop to reveal them.
- *Theft or extortion of secret keys:* The attacker either steals the secret keys of the bank (user/trustee), e.g. by hacking into its system or forces to reveal them. In the second case, the bank (user/trustee) is aware that the attack happened, which might not be the case after a theft. If this attack is used against a user, the attacker can also steal his electronic purse. Since the user's secret keys are only stored in the tamper proof hardware of his device, extortion might only be possible if the attacker also obtains the PIN of the user, e.g. by modifying the work of POS terminal or spying.
- *Blindfolding:* The attacker forces the bank (trustee) to engage in blindfolded protocols in order to obtain digital coins (certified pseudonyms), that he can spend successfully (use to withdraw untraceable coins).

These attacks might also be performed by a coalition of several entities, e.g. the bank and shop or the user and shop. A *fraudulent outsider* might attack several entities at once, e.g. bank and user or bank and trustee, in order to execute one of the attacks. Figure 4 summarizes the possible attacks.

honest / fraudulent	User	Shop	Bank	Trustee	Society
User	Impersonating forgery at overspending	Overspending	Overspending „one-more“ coin forgery	---	Combination of attacks
Shop	Impersonating forgery at overspending	Impersonating	Double deposit	---	Money laundry
Outsider	Coin extortion Extortion of secrets Eavesdropping forgery at overspending	Coin extortion Extortion of secrets Eavesdropping Impersonating	Blindfolding Extortion of secrets universal coin forgery	Blindfolding Extortion of secrets ---	Combination of attacks
Bank	Tracing Framing forgery at overspending	Framing	---	Tracing a honest user	Combination of attacks
Trustee	Tracing Framing	---	---	---	Combination of attacks
Coalition	Combination of attacks	Combination of attacks	Combination of attacks	Combination of attacks	

Figure 4: Summary of attacks

2.5 Security requirements

In order to resist the above attacks, untraceable electronic payment systems should fulfill many security requirements. We first describe those, that are imposed to be satisfied by all systems.

Unforgeability: Only authorized entities like banks are able to issue valid digital coins.

Untraceability: The relationship between a digital coin and a user is untraceable for the bank, except in the case of authorized revocation.

Unlinkability: Different coins spent by the same user are unlinkable.

Framing: No user or shop can be falsely incriminated by the bank or trustee.

In order to prevent passive and active eavesdropper attacks all communication should be authentic (integer) and confidential if necessary. Secondly, there are additional requirements in order to obtain fair electronic cash systems with revocable anonymity [CaMS96, FuOk96, FrTY96]:

Overspent-tracing: The bank can determine the identity of the user who overspends a coin. It is either realized by a separate mechanism or in the same way as user-tracing.

User-tracing: The bank and the trustee cooperate to match a spent coin to the user.

Coin-tracing: The bank and the trustee cooperate to compute information that allows the matching of a coin when it is spent or deposited. In some systems it might be possible, that the user himself announces the necessary information after an extortion of digital coins.

Extortion-tracing: The bank and trustee cooperate in order to compute information that allows the matching of a coin when it is spent or deposited.

Notice that overspent-tracing is achieved in all recent payment systems but extortion-tracing was only possible using an *on-line* payment protocol [JaYu96, FuOk96]. We are the first to present a solution that allows an *off-line* prevention of this attack.

2.6 Versatility

The basic concept of electronic coins might be extended in order to obtain versatile and efficient systems [OkOh91, PfWa96]. Additional properties are

- the *issue of receipts* to prove to a third party that a transaction took place,
- the *k-spendability* of a coin, such that the user's identity can be traced only if he spends the coin more than k times, or more generally
- the *divisibility* of the coin value v in fractions, such that the sum of all values of the fractions is equal to v.
- the support of the *challenge semantics*, introduced by [JaYu96], in order to extend the functionality by letting some bits of the challenge by the shop represent the meaning of the payment.

These properties can be efficiently achieved by our system. Other useful properties might be

- the *transferability* of coins between users, without interacting with the bank [ChP92a],
- the *loss tolerance* of unspent coins in order to credit their value after some time period if the device is lost [WaPf89, PfWa95].

3. Classification of fair payment systems

Fair payment systems might be classified by various aspects affecting either their efficiency or security and privacy. A first characteristic is, if the payment is done *on-line* or *off-line*. Off-line systems are more efficient, but don't allow the cryptographic prevention of double spending. Thus this must be prevented either by physical measures or detected afterwards by cryptographic mechanisms. Another aspect with regard to efficiency is the trustee involvement in the system:

1. The trustee is involved only during the *initialization* of the system and for *revocation* [CaMS96, FrTY96, DFTY97]. The user proves to the bank (and the shop) that the coin transcript contains information that allows the trustee to revoke the anonymity.
2. The trustee is involved only during the *registration* of the user [CaPS96, FuOk96, BrGK95, RaGV97].
3. The trustee is involved in *every withdrawal* [CaPS95, JaYu96, M'Raï96, JaYu97]. He performs or keeps trace of the blinding of a coin. Thus he is able to revoke its anonymity.
4. In case of any extortion attack the trustee is involved by shop during each *payment* of coins that were signed under suspicious keypairs of the bank [JaYu96, FuOk96, JaYu97].

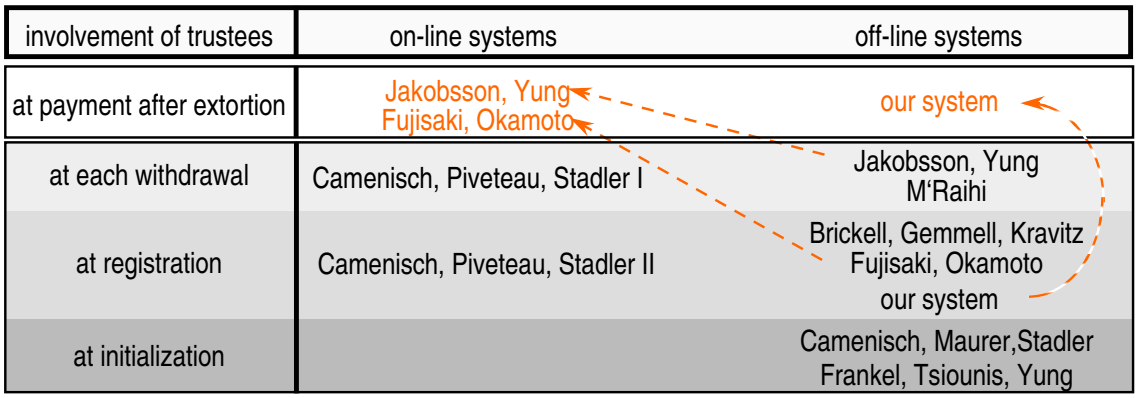

Figure 5: First classification of fair payment systems

A first classification considering these aspects is given in figure 5. The dashed arrows indicate the transformation of the systems after an extortion has been reported.
Another aspect with regard to security is the supported anonymity *revocation* (see section 2.5). For efficiency it can be considered, if the basic cash system already supports overspent-tracing without the help of the trustee. Furthermore it can be classified, if the underlying blind signature scheme is based on the RSA signature scheme [Chau82, RiSA78] or the Schnorr signature scheme [Schn89, Okam92, ChP92b]. A last aspect is, if different payments of the same user are *linkable* by anyone, by the trustee or *unlinkable* which leads to a more detailed distinction:

1. *Linkable payments:*
   The system in [FuOk96] is based on the payment system of Chaum [Chau82, Chau87]. The user-tracing enables at the same time the overspent-tracing. The systems in [CaPS95, CaPS96] are based on an on-line payment system of [CaPS94], which prevents the overspending of coins from the beginning. The systems in [CaPS96, FuOk96] need the trustee at registration to issue the user certified, pseudonymous data, which he uses at payment. In [CaPS96] the relation between an anonymous and a personal account is notified by the trustee and the anonymous account number is certified by him. In [FuOk96] random public parameters are certified and their relation to the user's identity is stored by the trustee. All coins, that are withdrawn using the same pseudonym are thus linkable by anyone, as the pseudonym is shown at each payment. To obtain *unlinkable* payments, the user has to choose a new certified pseudonym for each withdrawal. Then these systems differ from the unlinkable systems only in their storage requirements but benefit from a more efficient withdrawal protocol (the well known dualism between storage and efficiency of protocols).

2. *Trustee linkable payments:*
   The system in [M'Raï96] is also based on the payment system of Chaum. The user obtains several pseudonyms from the bank at the opening of an account, which he uses during withdrawal at the trustee. The trustee is engaged in the blind signature protocol with the bank and obtains the coin for the user. This allows him trivially to trace all coins to the user's pseudonym. Therefore, he has to store one tuple for each withdrawn coin. The pseudonym is not part of the coin, thus no-one except the trustee is able to link different payments. To obtain *unlinkable* payments even for the trustee, the user has to choose a new pseudonym for each communication with the trustee and is allowed to withdraw only one coin each

time, as otherwise the trustee would be aware of the relation among different pseudonyms.

3. *Unlinkable payments:*
   These systems are subdivided by the basic payment system.

   a) *Simple systems:* The systems in [JaYu96, JaYu97] are based on the payment system of Chaum [Chau87]. The trustee obtains a new pseudonym of the user at each withdrawal. He is involved in each blind signature protocol with the bank and stores the protocol data. The bank is able to obtain the user's identity from the pseudonym shown at the trustee. Thus the bank together with the trustee are able to trace coins. Furthermore, the system supports extortion-tracing employing an on-line payment protocol with the bank and trustee, to verify if a spent coin was legally withdrawn by the user.

   b) *Complex systems:* The systems [BrGK95, CaMS96, FrTY96, DFTY97] are based (except the second system in [BrGK95]) on the payment system of Brands [Bra93a, Bra93b]. In all systems, different payments are unlinkable and the overspent-tracing and user-tracing are realized by independent mechanisms, such that the need of the trustee is not always requested. The protocols need high computational effort at withdrawal of the coins, as the user proves interactively, that the coins might be traced by the trustee later, without involvement of the trustee in this proof.

Summarizing all aspects we obtain the comparison in figure 6.

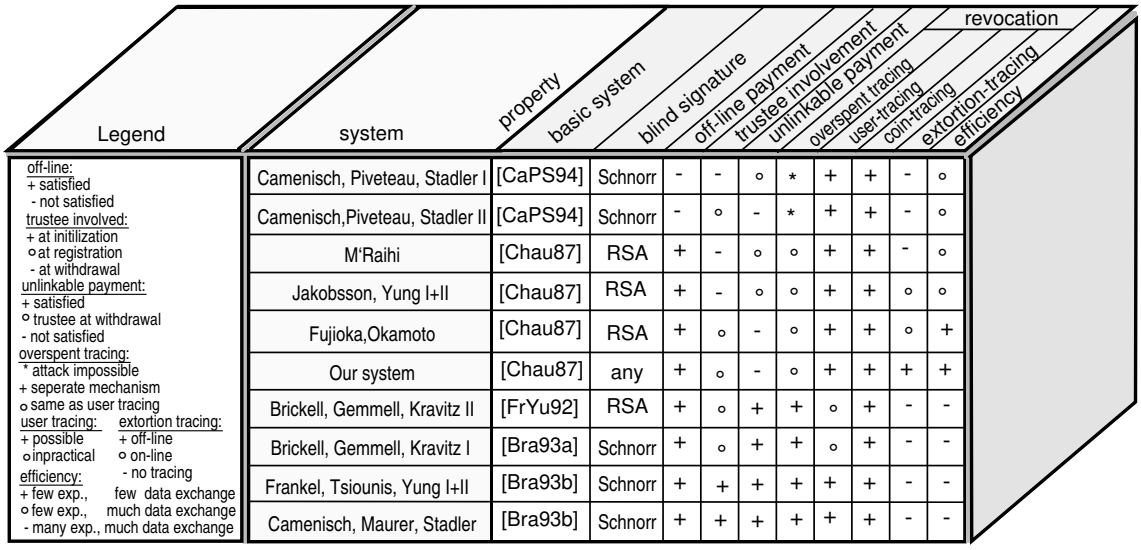

Figure 6: Comparision of basic properties

4. An efficient fair cash system

We focus the description on the payment of coins for a *unique* coin value and a *single* time of validity. Generally, a combination of these two aspects is achieved either by a corresponding keypair of the bank or by using advanced techniques (e.g. [Bra93b, AbFu96]).

4.1 Basic design

The design of most payment systems is based on the approach of Chaum [Chau82, Chau87]. We also take use of it but apply a few modifications as it is shown in figure 7. First, the user opens an account at the bank but he also has to register at the trustee and consequently obtains a certified pseudonymous keypair (PS_x, PS_y) in our system. In fact, the presence of this trustee is the main difference between the two systems but it allows to achieve an enhanced security. Then he uses a blind signature scheme in order to withdraw anonymous digital coins from the bank. In the Chaum's system, the coin is just a number of a special form. In ours, the public part of the user's pseudonym PS_y is embedded into the coin and PS_x is used during payment to generate a digital signature σ_C to prevent various attacks. Then the coin is always spent using a "challenge-response" protocol between the user and shop in order to sign a random challenge under the user's secret key C_x (resp. secret pseudonym PS_x), which is verifiable by the corresponding public key C_y (resp. PS_y). The parallel between the basic system [Chau87] and our proposal can easily be seen in figure 7.

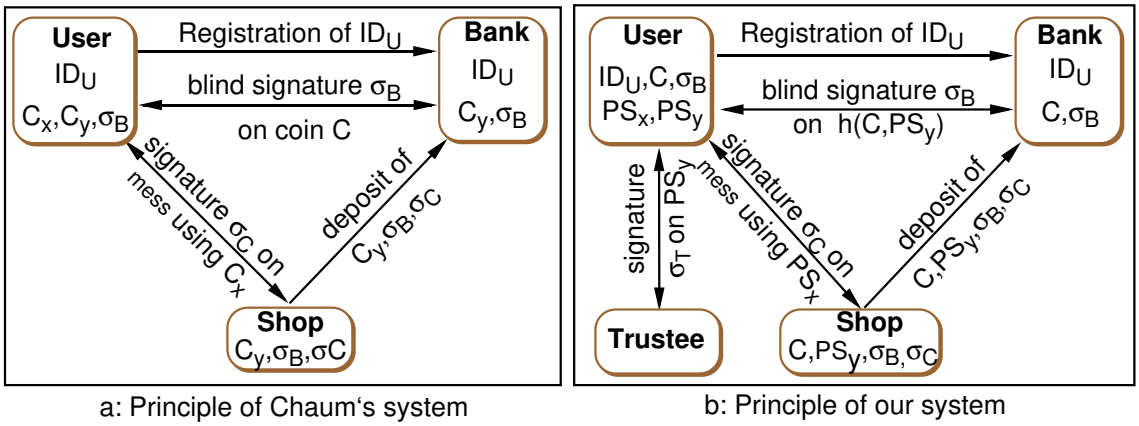

Figure 7: Basic designs of payment schemes

4.2 Cryptographic tools

We need six cryptographic tools as building blocks to design our system. These are

1. a collision resistant cryptographic hash function h,
2. three (different) signature schemes $(\mathcal{G}_\mathcal{T}, \mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$, $(\mathcal{G}_\mathcal{U}, \mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$, $(\mathcal{G}_\mathcal{C}, \mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$ with corresponding security parameters, message spaces, key generation algorithms $\mathcal{G}_\mathcal{T}, \mathcal{G}_\mathcal{U}, \mathcal{G}_\mathcal{C}$, signature algorithms $\mathcal{S}_\mathcal{T}, \mathcal{S}_\mathcal{U}, \mathcal{S}_\mathcal{C}$ and verification algorithms $\mathcal{V}_\mathcal{T}, \mathcal{V}_\mathcal{U}, \mathcal{V}_\mathcal{C}$ used by the trustee, the user and for signing payments respectively (for exact definition see [GoMR88]),
3. a blind signature scheme $(\mathcal{G}_\mathcal{B}, \mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$ used by the bank to sign withdrawn coins,
4. an interactive authentic key exchange protocol $\mathcal{K}$ with mutual user authentication that generates a fresh authentic session key K for the authentic communication between A and B. We will denote it as $K_{A,B} := \mathcal{K}(ID_A, ID_B)$ (for a survey of these schemes see [RuOo94]),
5. a probabilistic encryption scheme $(\mathcal{E}_\mathcal{Z}, \mathcal{D}_\mathcal{Z})$ used with the trustee and shops. The encryption of a message m for $\mathcal{Z}$, $\mathcal{Z} \in \{T, S\}$ is denoted as $e := \mathcal{E}_\mathcal{Z}(m)$ and decryption as $m := \mathcal{D}_Z(e)$,

6. a symmetric cryptosystem $(\mathcal{E}_K, \mathcal{D}_K)$, where K is the session key.

These tools are defined as usually (see e.g. [MeOV97]). The three signature schemes might be chosen independently by the entities. $(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$ must be an existentially unforgeable signature scheme in order to prevent transparently blindfolding attacks [JaYu96]. The key generation algorithms $\mathcal{G}_\mathcal{Z}$ with $\mathcal{Z} \in \{B, T, U, C\}$ return keypairs $(x_\mathcal{Z}, y_\mathcal{Z})$ for the bank, trustee, user and signature generation for the coins respectively. The generation of a signature by entity $\mathcal{Z}$ for message m is described as $\sigma_\mathcal{Z} := \mathcal{S}_\mathcal{Z}(x_\mathcal{Z}, m)$ and the verification of this signature by $\mathcal{V}_\mathcal{Z}(y_\mathcal{Z}, m, \sigma_\mathcal{Z}) \in \{true, false\}$. We attach ID_U to user U with account acc_U and ID_S to shop S with account acc_S. $Ind(x)$ is a pointer to the stored secret key x.

4.3 Databases

The scheme employs different databases for the participants:

- The *bank* needs three different databases, that should be protected against modification and erasing:
  1. a user database (U-DB) to store the user's identities and account numbers,
  2. a database of withdrawal transcripts (W-DB) and
  3. a database of deposit transcripts (D-DB).
- The *user* needs two databases:
  1. a coin database (C-DB) to store withdrawn but not already spent coins,
  2. a pseudonym database (PS_x-DB) to store the secret pseudonyms confidentially.
- The *shop* needs a payment database (P-DB) to store coins before their deposit.
- The *trustee* needs a pseudonym database (PS_y-DB) to store the pseudonyms and identities of the users.

An overview about necessary databases is given in figure 8. Figure 8b shows the influ-

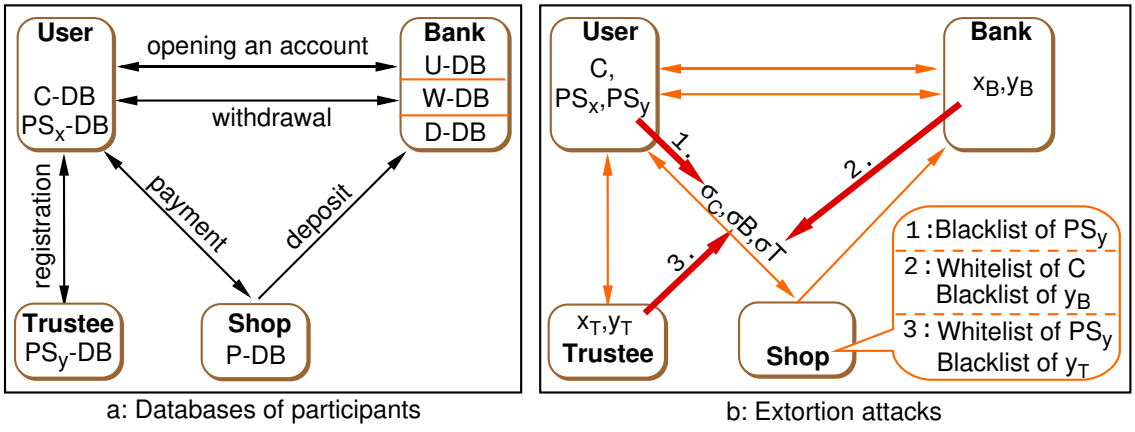

Figure 8: Overview of databases and blacklists

ence of an extortion of the different secret parameters PS_x, x_B, x_T on the forgerability of the digital signatures $\sigma_C, \sigma_B, \sigma_T$ involved (indicated by the three arrows 1.-3.). After extortion attacks, temporarily different lists are distributed among the shops:

- a *blacklist* of blackmailed pseudonyms of users,
- a *blacklist* of blackmailed secret keys of the trustee together with a *whitelist* of still acceptable pseudonyms after this attack and
- a *blacklist* of blackmailed secret keys of the bank together with a *whitelist* of still acceptable coins after this attack.

To shorten the lists, it is recommended to store only a hash value of the compromised keys (e.g. 10 byte). In all cases, the secret parameters of the attacked entities are replaced immediately, such that all lists expire after the maximal lifetime of the coins. The lists are only an *emergency measure*, which should deter fraudulent entities from committing crimes, as they might be detected anyway. Thus the efficiency of these countermeasures is not highly important, as they are normally not used. Anyway they should be efficient enough to be practical to prevent denial of service attacks. The content of the databases (DB), blacklists (BL) and whitelists (WL) are resumed in figure 9.

owner	abbre-viation	access at/after	stored information	authenticity guaranteed by	confi-dential
databases					
Trustee	PS_y-DB	registration	$ID_U, PS_y, \sigma_U, Ind(x_T)$	σ_U	yes
Bank	U-DB	opening account	ID_U, acc_U	–	no
	W-DB	withdrawal	ID_U,C,$Ind(x_B)$, $\mathcal{E}_{\mathcal{T}}(h(PS_y, C))$[3]	$\tilde{\sigma}_B$	no
	D-DB	deposit	$C, PS_y, ID_s, \sigma_T, \sigma_C$	σ_T, σ_C	no
User	C-DB	withdrawal and payment	$C, \sigma_B, Ind(PS_x)$	σ_B	no
	PS_x-DB		$PS_x, (PS_y), \sigma_T$	σ_T	yes
Shop	P-DB	payment	$C, PS_y, \sigma_B, \sigma_T, \sigma_C$	σ_B, σ_C	no
revocation lists					
Shop	U-BL	user extortion	voided PS_y	$\mathcal{S}_{\mathcal{T}}(x_T, PS_y)$	no
	C-WL	bank extortion	valid $h(PS_y, C)$'s	$\mathcal{S}_{\mathcal{T}}(x_T, h(PS_y, C))$	no
	B-BL	bank extortion	voided y_B	$\mathcal{S}_{\mathcal{T}}(x_T, y_B)$	no
	PS-WL	trustee extortion	valid PS_y	$\mathcal{S}_{\mathcal{T}}(\tilde{x}_T, PS_y)$[4]	no
	T-BL	trustee extortion	voided y_T	$\mathcal{S}_{\mathcal{CA}}(x_{CA}, y_T)$	no

Figure 9: Databases and revocation lists

4.4 General protocol

An overview about the main transactions is given in figure 10.

Opening an account

1. The user U with identity ID_U identifies himself to the bank and obtains an account number acc_U.
2. The bank stores (ID_U, acc_U) in its U-DB.

[3] In case of electronic purse, this is the symmetric encrypted value $\mathcal{E}_{PS_x}(C)$.
[4] The signature is generated using a fresh trustee's key $\tilde{x}_T$ generated after the attack.

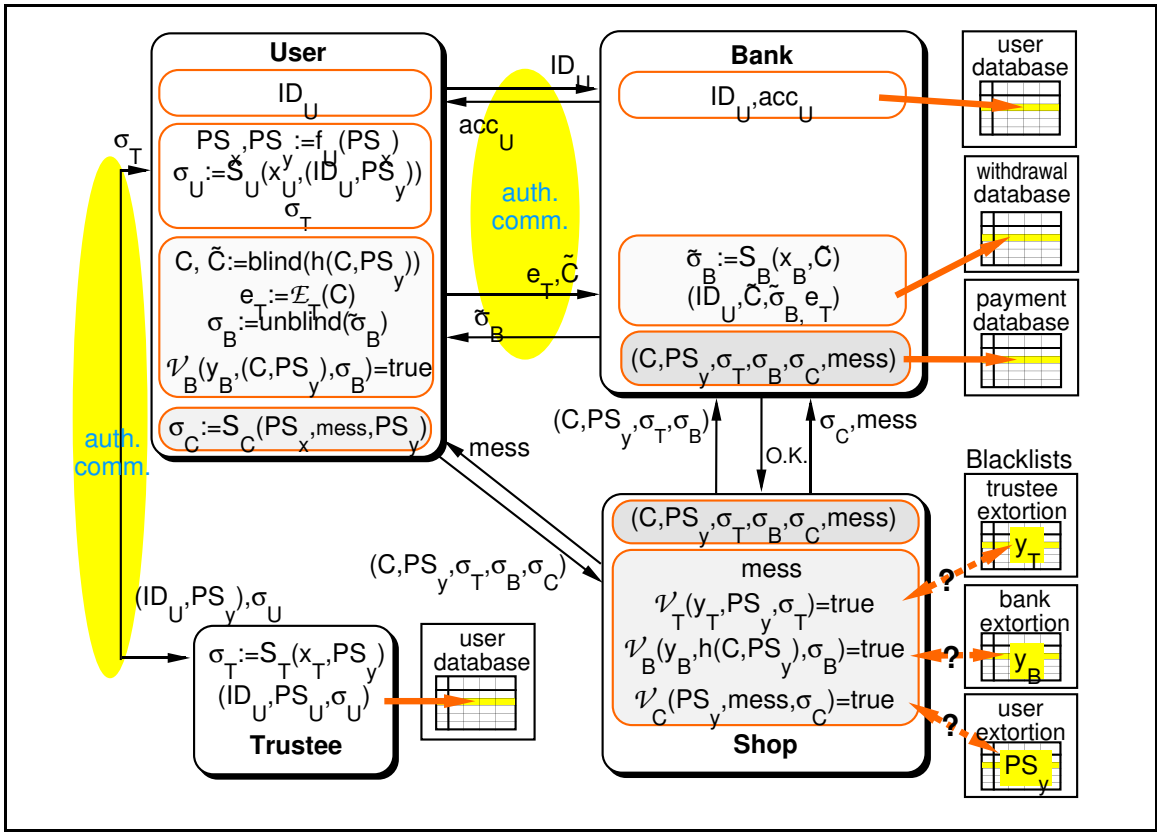

Figure 10: Overview of the main transactions

Registration at trustee

1. To identify himself to the trustee and to obtain an authentic session key $K_{U,T}$, the user and trustee participate in the authentic key exchange protocol $\mathcal{K}$ and obtain $K_{U,T} := \mathcal{K}(ID_U, ID_T)$. To obtain a confidential and authenticated communication $K_{U,T}$ is used for encryption of all communication in steps 2. and 3.
2. The user generates a pseudonymous keypair $(PS_x, PS_y) := \mathcal{G}_\mathcal{C}$. He computes $\sigma_U := \mathcal{S}_\mathcal{U}(x_U, (ID_U, PS_y))$ and sends $\mathcal{E}_{K_{U,T}}(ID_U, PS_y, \sigma_U)$ to the trustee.
3. The trustee verifies $\mathcal{V}_\mathcal{U}(y_U, (ID_U, PS_y), \sigma_U) \stackrel{?}{=} true$, calculates $\sigma_T := \mathcal{S}_\mathcal{T}(x_T, PS_y)$ and transmits $\mathcal{E}_{K_{U,T}}(\sigma_T)$ to the user U. He stores $(ID_U, PS_y, \sigma_U, Ind(x_T))$ in his PS_y-DB.
4. The user verifies $\mathcal{V}_\mathcal{T}(y_T, PS_y, \sigma_T) \stackrel{?}{=} true$ and stores all values. He secretly stores PS_x.

These steps might be processed several times to obtain several pseudonymous keypairs (PS_x, PS_y). For an efficient *electronic purse* protocol, it is possible to make the modification that the trustee generates $(PS_x, PS_y) := \mathcal{G}_\mathcal{C}$ himself, signs PS_y, transmits all values confidentially to the user and keeps PS_x secret for a later use in the symmetric cryptosystem $(\mathcal{E}, \mathcal{D})$.

Withdrawal protocol

1. In order to identify himself to the bank and to obtain an authentic session key $K_{U,B}$ for the communication, the user and bank participate in the authentic

key exchange protocol $\mathcal{K}$ and obtain $K_{U,B} := \mathcal{K}(ID_U, ID_B)$. $K_{U,B}$ is used to authenticate the communication in steps 2. and 3. as in the registration protocol.

2. The user U generates random coin C (possibly containing some redundancy). He computes $\tilde{C} := blind(h(C, PS_y))$, $e_T := \mathcal{E}_{\mathcal{T}}(h(PS_y, C))$[5] and transmits these values to the bank. In case of an extortion attack against the bank, it sends all stored values e_T to the trustee (as described below).
3. The bank computes $\tilde{\sigma}_B := \mathcal{S}_{\mathcal{B}}(x_B, \tilde{C})$ and sends it to the user U. It subtracts the value of the coin from the user's account acc_U and stores $(ID_U, Ind(x_B), \tilde{C}, e_T)$ in his W-DB.
4. The user computes $\sigma_B := unblind(\tilde{\sigma}_B)$ and verifies $\mathcal{V}_{\mathcal{B}}(y_B, h(C, PS_y), \sigma_B) \stackrel{?}{=} true$. If the verification fails, the user asks the bank to resent the signature for the blinded coin $\tilde{C}$. As we assume in the security model that the bank doesn't steal money from the user, they repeat this until the verification of σ_B is correct. The user keeps (C, σ_B) as his coin and notices the relation to the tuple (PS_x, σ_T).

Payment protocol

1. The shop sends the user a uniquely generated message $mess$.
2. The user generates $\sigma_C := \mathcal{S}_{\mathcal{C}}(PS_x, (C, ID_S, mess))$ and sends the payment transcript $(C, PS_y, \sigma_B, \sigma_T, \sigma_C)$ to the shop. In the case of *internet payments*, the value σ_C is encrypted as $e_S := \mathcal{E}_{\mathcal{S}}(\sigma_C)$ in order to prevent its eavesdropping.

3a. If *no extortion* attack was reported, the shop verifies $\mathcal{V}_{\mathcal{T}}(y_T, PS_y, \sigma_T) \stackrel{?}{=} true$, $\mathcal{V}_{\mathcal{B}}(y_B, h(C, PS_y), \sigma_B) \stackrel{?}{=} true$ and also $\mathcal{V}_{\mathcal{C}}(PS_y, (C, ID_s, mess), \sigma_C) \stackrel{?}{=} true$. He stores the payment transcript together with $mess$ in his P-DB.

3b. After a *user extortion* attack the shop receives an actualized U-BL from the trustee. If $PS_y \in$ U-BL, he rejects the coin. Otherwise he accepts it as in step 3a.

3c. After a *bank extortion* attack the shop receives an actualized B-BL and C-WL from the trustee. If ($y_B \in$ B-BL and $C \notin$ C-WL) he rejects the coin. Otherwise he accepts it as in step 3a.

3d. After a *trustee extortion* attack, the shop receives an actualized T-BL and PS-WL from the trustee. If σ_T was generated under $y_T \in$ T-BL and $PS_y \notin$ PS-WL, he rejects the coin. Otherwise he accepts it as in step 3a.

Step 1 is omitted if the user computes a fresh, unique value $mess$, e.g. as a function of $time, contract$ etc. In the case of divisible coins $mess$ must contain the fraction C_f of coin C that is spent (see section 5). Then C can be spent multiple times upto its whole value, determined at withdrawal, has been spent.

Deposit protocol

1. The shop sends the tuple $(C, PS_y, \sigma_B, \sigma_T)$ to the bank.
2. The bank verifies $\mathcal{V}_{\mathcal{T}}(y_T, PS_y, \sigma_T) \stackrel{?}{=} true$, $\mathcal{V}_{\mathcal{B}}(y_B, h(C, PS_y), \sigma_B) \stackrel{?}{=} true$ and checks, whether the coin was already deposited under the same pseudonym PS_y. In this case, it finds a tuple (C, σ'_C) in D-DB and sends σ'_C to the shop as a proof. If $\sigma_C = \sigma'_C$ the shop is accused of double deposit and cannot deposit the coin. If $\sigma_C \neq \sigma'_C$ or if the coin C was not already deposited in D-DB, the bank sends the shop a signed acknowledgment about this fact.

[5] or $e_T := \mathcal{E}_{PS_x}(h(PS_y, C))$ in the case of electronic purse payments

3. The shop sends the tuple $(ID_S, acc_S, mess, \sigma_C)$ to the bank.
4. The bank verifies $\mathcal{V}_{\mathcal{C}}(PS_y, (C, ID_s, mess), \sigma_C) \stackrel{?}{=} true$ and checks if C has been overspent under PS_y. In this case it initiates the *user-tracing* protocol described below with the trustee to identify the cheating user. As a proof, it sends the trustee the payment transcripts $(C, PS_y, mess_1, \sigma_B, \sigma_T, \sigma_{C,1}), \ldots, (C, PS_y, mess_k, \sigma_B, \sigma_T, \sigma_{C,k})$.
5. If every thing is okay the bank stores the payment transcript in its P-DB and credits the shop's account acc_S by the value of C.

Overspent-tracing and user-tracing

1. In order to prove that the coin C was overspent the bank sends several payment transcripts $(C, PS_y, mess_1, \sigma_B, \sigma_T, \sigma_{C,1}), \ldots, (C, PS_y, mess_k, \sigma_B, \sigma_T, \sigma_{C,k})$ to the trustee.
2. The trustee verifies all transcripts, as the shop did in step 3 of the payment protocol.
3. If the verification is correct, the trustee looks for the tuple (ID_U, PS_y, σ_U) in his PS_y-DB and extracts (ID_U, σ_U) which he sends to the bank.

Coin-tracing

Usually coin-tracing is considered in unlinkable payment systems in order to prevent extortion or theft of coins. In our system this attack is impossible, as the knowledge of a coin C without the corresponding secret key PS_x doesn't help for successful payment of the coin. Thus this attack should be considered only if the pseudonym keypair (PS_x, PS_y) is revealed at the same time. In this case, the *extortion-tracing* described below applies as countermeasure.

Extortion-tracing

We describe the actions that are taken by the attacked party in order to guarantee that the extorted values are put on a revocation list. The treatment of extortion attacks by the shop have been described in steps 3b-d. of the payment protocol above.

1. *Extortion of user's secret key PS_x:*
   The user reports the attack to the trustee, who looks in his list PS_y-DB for (PS_y, ID_U). If ID_U doesn't correspond to the user he refuses to accept this attack, in order to prevent false extortion reports by malicious users. Otherwise he puts PS_y on U-BL and distributes this list immediately among the shops. He issues $\mathcal{S}_{\mathcal{T}}(x_T, (ID_U, PS_y))$ to the user. This allows the user to deposit or exchange unspent coins withdrawn under PS_y at the bank, after he has identified as their legal owner.
2. *Extortion of bank's secret key x_B:*
   To avoid the forgery of electronic coins in case of an extortion of x_B, the bank sends the encrypted values $E_T(h_2(C_i, PS_y))$ of all legally withdrawn coins C_i in his W-DB to the trustee. The trustee decrypts the $h_2(C_i, PS_y)$'s and puts them on C-WL. Additionally, the public key y_B corresponding to x_B is put on B-BL. Both lists are immediately distributed among the shops. The bank generates a fresh keypair $(\tilde{x}_B, \tilde{y}_B) := \mathcal{G}_{\mathcal{B}}$ for the issue of new coins.

3. *Extortion of trustee's secret key x_T:*
   After an extortion of x_T the trustee puts all pseudonyms PS_y in his PS_y-DB legally signed under this key on PS-WL and at the same time the public key y_T that corresponds to x_T on T-BL. Both lists are distributed immediately among the shops. The trustee generates a fresh keypair $(\tilde{x}_T, \tilde{y}_T) := \mathcal{G}_{\mathcal{T}}$ for the certification of user pseudonyms.
4. *Blindfolding coins under bank's secret key x_B:*
   Transparent blindfolding of coins is not possible, because of the authentic key exchange protocol $\mathcal{K}$ with user authentication between the user and the bank. If the user enforces the blindfolded withdrawal of a coin without proper identification, the bank is aware that the attack happens and takes the same countermeasures as in case of extortion of x_B.
5. *Blindfolding pseudonyms PS_x, PS_y under trustee's secret key x_T:*
   As transparent blindfolding is impossible due to the use of an existentially unforgeable signature scheme [JaYu96] the trustee will be aware of the attack and takes the same countermeasures as under point 3. in the case of an extortion attack of x_T.

4.5 Reducing trust in trustees

In the above protocol, the user has to trust the trustee that

- he doesn't reveal his pseudonym to anyone, such that his payments will become traceable (*anonymity w.r.t. to trustee* as requested in [JaYu96] is not yet satisfied).
- he interacts correctly for decryption of $e_i = E_T(C_i)$ in case of an extortion of the bank's secret key x_B as otherwise the user is unable to spend valid coins C_i signed under x_B afterwards.

It depends on the *trust model* if the users accept these conditions. Otherwise it is possible to reduce trust in the trustee by sharing PS_y and e_i among some trustees [FuOk96]. Then the signature scheme $(\mathcal{G}_{\mathcal{T}}, \mathcal{S}_{\mathcal{T}}, \mathcal{V}_{\mathcal{T}})$ will be substituted by a multisignature scheme and the probabilistic encryption scheme $(\mathcal{E}_{\mathcal{T}}, \mathcal{D}_{\mathcal{T}})$ by an encryption scheme with either threshold- or multi-decryption (for a survey of these schemes see [Desm94]).

In order to prevent *only* the linkability of payments w.r.t. the trustee, the following modifications of the *opening of an account* and *registration* protocols are sufficient, if the communication between the user and the trustee can be anonymous. In fact, this can be achieved using trusted mixes on the Internet, by changing the communications structure between the user and trustee e.g. via the bank as in [JaYu96] or using public POS-terminals in the case of electronic purse applications.

1. At opening of an account, the bank signs $h(acc_U)$, i.e. $\sigma_P := \mathcal{S}_{\mathcal{B}}(x_B, acc_U)$ and transfers this signature confidentially together with acc_U to the user.
2. The user registers at the trustee by sending the tuple $(h(acc_U), \sigma_P, PS_y)$ to the trustee where $h(acc_U), \sigma_P$ replaces (ID_U, σ_U).
3. In the case of *user-tracing*, the trustee finds $(h(acc_U), \sigma_P, PS_y)$ in his PS_y-DB which he sends to the bank. This allows the bank to match PS_y to acc_U and ID_U respectively.

This modification is even more efficient for the user, as it avoids the generation of σ_U for him. Similar mechanisms to solve this problem have been proposed in [JaYu96, M'Raï96].

4.6 Security analysis

For the security analysis we benefit from the modular design of our system using well known cryptographic primitives. Although all attacks mentioned in section 2.4. will be proved to be prevented, the security analysis is clearly structured as we avoided interaction between the mechanisms as much as possible. Theorems 4.1 - 4.6 analyse the influence of the cryptographic tools described in section 4.2 for security and theorems 4.7 - 4.9 with the importance of the different revocation lists, introduced in section 4.3.

Definition 4.1 *A signature scheme $(\mathcal{S}, \mathcal{V})$ is called* provably computational secure*, if existential forgery of a signature with respect to an adaptively chosen message attack is proved to be equivalent to a known computational hard problem (e.g. factorization or discrete log).*

Theorem 4.1 *Assuming $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$ is a provably computational secure* blind *signature scheme (in the sense of [PoS96b]) and the function h is a collision intractable hash function, the system achieves unforgeability and therefore is secure against* coin forgery*.*

Proof 4.1: Since $\mathcal{S}_\mathcal{B}$ is secure against existential forgery, all coin signatures σ_B on the message $h(C, PS_y)$ must have been (blindly) generated by the bank. As the hash function h is collision intractable by the user, it is impossible for him to find a coin value $C' \neq C$ (or $PS'_y \neq PS_y$) with $h(C', PS_y) = h(C, PS_y)$ ($h(C, PS'_y) = h(C, PS_y)$ respectively). Therefore, the system achieves unforgeability of coins. □

Theorem 4.2 *Assuming $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$ is a provably computational secure* blind *signature scheme (in the sense of [PoS96b]) and $(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})$ is a strong probabilistic encryption scheme, the system achieves untraceability and therefore is secure against* tracing a user by the bank*.*

Proof 4.2: As $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$ is a perfect blind signature scheme, the signed coin (C, σ_B) is untraceable to the view $(\tilde{C}, \tilde{\sigma}_B)$. Furthermore, the ciphertext e_T delivered with the blind coin, doesn't allow the bank to match it to a given coin later-on, as $(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})$ is a probabilistic encryption scheme. □

Theorem 4.3 *Assuming $(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$ is a provably computational secure signature scheme, the system is secure against* (1) impersonation, (2) framing a user by the bank, (3) tracing a user with help of trustee *and* (4) coin forgery after overspending*.*

Proof 4.3: Since anybody who does not know the secret key PS_x cannot produce a signature σ_C of a tuple $(C, ID_S, mess)$ such that $\mathcal{V}_\mathcal{C}(PS_y, \sigma_C, (C, ID_S, mess)) = true$, impersonation by the bank is impossible. If the bank wants to frame an honest user who did not overspent a coin, she must show at least two tuples $(C, PS_y, mess_1, \sigma_B, \sigma_T, \sigma_{C,1})$ and $(C, PS_y, mess_2, \sigma_B, \sigma_T, \sigma_{C,2})$. But since $(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$ is a provably computational secure signature scheme, the bank cannot produce more signature $\sigma_{C,i}$ than she received. So the attack (2) is prevented. The two last ones are also avoided for the same reason, the knowledge of some signatures $\sigma_{C,1}, \ldots \sigma_{C,k}$ on the same coin by the bank does not help to generate new signatures of the user on this coin. □

Theorem 4.4 *Assuming* $(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$ *is a provably computational secure signature scheme, the system is secure against* framing a user by the trustee.

Proof 4.4: Since the trustee needs to generate a valid signature $\mathcal{S}_\mathcal{U}(x_U, (ID_U, PS_y))$ for a given PS_y and any known ID_U in order to frame user U, this attack is not possible assuming $(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$ is a provably computational secure signature scheme. □

Theorem 4.5 *Assuming* $(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$ *is a provably computational secure signature scheme, the system achieves* overspent- *and* user-tracing *and is therefore secure against* overspending, money laundry *and* transparently blindfolding of the trustee.

Proof 4.5: Since $\mathcal{S}_\mathcal{T}$ is unforgeable, a signature σ_T is a proof that the trustee knows a relation between a real user identity and a signed pseudonyms PS_y. In the mechanism of user tracing, the bank sends a tuple $(C, PS_y, mess, \sigma_B, \sigma_T, \sigma_C)$ and the trustee verifies all signatures. Since it was already proved, that they were necessarily generated by the correct entities, σ_B authenticates a coin $h(C, PS_y)$, then σ_C proves that C was in fact spent under pseudonym PS_y and finally σ_T proves that the trustee knows a pair (PS_y, ID_U) to relate C to ID_U. So *overspending* and *money laundry* are prevented. Moreover, engaging the trustee in a transparently blindfolding protocol for σ_T is impossible, as assuming the contrary the user would know two valid signatures after only one interaction with signer (the blinded one and the unblinded one), which contradicts the existential unforgeability of $\mathcal{S}_\mathcal{T}$. □

Theorem 4.6 *Assuming, the key exchange protocol* $\mathcal{K}$ *is secure against any active and passive attacks and the symmetric cryptosystem* $(\mathcal{E}_\mathcal{K}, \mathcal{D}_\mathcal{K})$ *is not breakable without knowing the proper key* K*, the system is secure against* eavesdropping of coins and pseudonyms *and* framing of a shop by a bank.

Proof 4.6: The user communication at registration and withdrawal is protected under an authentic session key obtained from the authentic key exchange protocol $\mathcal{K}$. As it is resistant to eavesdropping and man-in-the-middle attacks, the communication protocols inherit this property. Since transmission of σ_C between the user and the shop is encrypted under y_S, the bank does not know σ_C and thus can't frame the shop. □

We are now going to study the security of our system against extortion of coins or secret keys. The case of their theft is similar if it is immediately discovered (e.g. by frequent audit). If the theft is not remarked, there are no cryptographic ways to protect the system.

Theorem 4.7 *Assuming that the user blacklist (U-BL) is properly used*[5]*, the system achieves all kinds of extortion-tracing and is thus secure against* extortion of coins.

Proof 4.7: The *extortion of coins from the user* in order to spend or deposit them is impossible, as the relation between C and PS_y is embedded in the structure of the coin and protected by σ_B. Thus C has to be signed with the corresponding PS_x at payment, which appears in the U-BL, if it was stolen together with C. The *extortion of coins from the shop* in order to deposit them at the bank on the attackers account is impossible, as the shop's identity is included in the signed message as payment, which prevents to credit them to another account. □

Theorem 4.8 *Assuming that all blacklists described in section 4.3 (U-BL, B-BL and T-BL) are properly used* [5]*, the system achieves all kinds of extortion-tracing and is thus secure against* extortion of secret keys.

[5] e.g. it is immediately updated and distributed after fraud has been reported

Proof 4.8:

- In case of *extortion of the user's secret key* PS_x, PS_y is blacklisted immediately and therefore any coin withdrawn under this pseudonym is rejected by any shop.
- In case of an *extortion* of the bank's secret key x_B, it is put immediately on B-BL, which prevents his further use. The hash values of all legally obtained coins signed under x_B are listed in an authentified whitelist, which prevents the use of others, that have been signed hereafter using x_B.
- In case of *extortion* of the trustee's key, the mechanism is the same as before, replacing x_B by x_T and coins by pseudonyms. □

Theorem 4.9 *Assuming, that the coin and pseudonym whitelists (C-WL and PS-WL) are properly used, no money that hasn't been already spent, is lost for the honest user.*

Proof 4.9:

- If a pseudonym PS_x was extorted and therefore blacklisted, the legal user of this pseudonym gets a signature $\mathcal{S}_{\mathcal{T}}(x_T, (PS_y, ID_U))$ from the trustee. This allows him to exchange withdrawn coins under this pseudonym at the bank for fresh coins, after he has authentified himself as their legal owner.
- If the bank's secret key x_B was extorted and therefore blacklisted, the users who possess legally withdrawn coins C under this secret key can still spend them, as their hashes $h(PS_y, C)$ appear on the whitelist C-WL. Nobody else can forge these coins, as he doesn't know C and the pre-image of PS_y, which are both necessary to spend it.
- If the trustee's secret key x_T was extorted and therefore blacklisted, the users who possess legally obtained pseudonyms PS_y certified under this secret key, can still use them, as they appear on the whitelist PS-WL. Nobody else can reuse them, as he is not able to obtain the corresponding secret key PS_x. □

5. Versatility of system

The system can be easily modified in order to achieve more security properties, to reduce the trust needed in each other or to make it more efficient. We are going to describe a few extensions :

- The issue of *receipts* can be added to all transactions as described in [Chau87] in order to minimize the trust between the entities. Using such a mechanism, the user no longer has to trust his bank to be honest because the receipts can prove who is guilty in case of disagreement.
- The use of *challenge semantics* permits to add information to payment transcripts and consequently all the applications already described by Jakobsson and Yung.
- The *k-spendability* or more general *divisibility* of coins is obtained as in the schemes [JaYu96, M'Raï96] by including the spent fraction C_f of the coin C into the message at payment. This needs one signature for each fraction that is spent. A more efficient way to achieve divisibility is the micro-payment approach described in the following.

- An alternative solution to obtain *divisible* coins are micro-payments, e.g. described by Rivest and Shamir, Anderson, Pedersen or for the μ-iKP scheme [RiSh96, AnMS96, Pede96, HaSW96]. Their extension is adaptable to our scheme, by using hash chains like in payword [RiSh96]. The user chooses a random value w_n and generates $w_i = h(w_{i+1})$ for $i = [n-1:0]$. Then he withdraws the coin $C = w_0$ with value v, that is divisible into n parts. For example, we consider a \$1-coin C with $n = 100$. To perform micro-payment, the user first *registers* at the shop by forwarding the coin C like in a "normal" payment. Then he might spend \$1 but he can also divide it into many micro-payments, that do not need the use of strong cryptographic tools like digital signature for being spent. For example, if he wants to spend 5 cents and an hour later 7 cents, he first reveals w_5 to the shop who can verify $h^{(5)}(C) = w_5$. Then he reveals $w_{5+7} = w_{12}$ and the shop checks $h^{(7)}(w_{12}) = w_5$. When the user has spent all parts (micro-coins) of the original coin C, the shop deposits C at the bank as usually, by adding w_n as a proof that the whole coin C has been spent. This extension is very useful for internet applications, e.g. in order to reduce the communication and computational effort. A small disadvantage is, that the coin must be wholly spent at the same shop, but possibly during different payments. Reimbursement of unspent parts of a coin C violates the untraceability of the coin.

- The *transferability* of coins is possible as described in [ChP92a] but not very efficient, as the size of a transferred coin increases in size.

6. Scalability of cryptographic tools

The choice of concrete cryptographic tools in our general scheme allows to obtain scaleability w.r.t. to security and efficiency. We discuss possible choices of those tools, that influence the efficient implementation on the user's device.

- *Choice of blind signature scheme* $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$*:* Tradeoff between efficient blinding & storage on the user's device and provable security. The most *efficient* scheme w.r.t. *blinding* is the blind RSA signature scheme with small public exponent. It allows message-recovery and therefore signed values don't have to be stored. The most *efficient* scheme w.r.t. signature *storage* is the blind Schnorr signature [ChP92b, Okam92]. Also other variants of the family of ElGamal signature schemes that allow message recovery might be considered [NyRu94, HMP94b], which reduces the parameter size further.

  A *provably computational secure* choice is the blind Okamoto signature [PoS96b]. Also other schemes, like restrictive blind signatures [Bra93b] that allow to prevent overspending without the help of the trustee, might be considered, although there is no advantage to use them as the trustee is needed anyhow.

- *Choice of user and coin signature schemes* $(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$ *and* $(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$*:* Tradeoff between efficient generation and provable security. PKP [Sham89] or CLE [Ster94] are *original* choices well suited for electronic purse. The generated signature is of larger size in these cases, which doesn't matter for the user, as they have to be stored only in the bank's database. An *efficient* and at the same time *provably computational secure* choice is the Schnorr signature [Schn89, PoS96a]. Besides these, any signature scheme from the ElGamal-family

(e.g. [ElGa85, NIST91, HMP94a]) or RSA [RiSA78] might be used, which also offers message recovery and thus reduces the storage for the bank's database.

- *Interaction between signature scheme* $(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$ *and* $(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$*:* The verification of the signature σ_C is done using the public key PS_y, which itself is certified by the digital signature σ_T. Therefore the shop has to verify *two* signatures subsequently. If the key PS_y is issued as a self-certified key by the trustee, both verifications can be performed in a single step [Pete96, PeHo97]. We will show an example for this: The trustee issues $PS_x := x_T \cdot h(r_T, data) + k_T \pmod q$, such that the corresponding public part is computed as $PS_y := y_T^{h(r_T, data)} \cdot r_T \pmod p$, where *data* contains e.g. some information about the validity of the pseudonym. This expression might be used to compute PS_y in the verification equation of the signature σ_C. If e.g. σ_C is chosen as a Schnorr signature, it is generated as $s_C := PS_x \cdot h(r_C, C, mess) + k_C \pmod q$. This leads to the verification equation

$$\alpha^{s_C} \equiv PS_y^{h(r_C, C, mess)} \cdot r_C \equiv (y_T^{h(r_T, data)} \cdot r_T)^{h(r_C, C, mess)} \cdot r_C \pmod p,$$

  which can be verified by 1.16 times the effort of a single signature verification [YeLL94], thus the savings is about 42% of computational effort.

- *Probabilistic encryption scheme* $(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})$*:* As probabilistic public-key cryptosystem we might choose the ElGamal encryption scheme [ElGa85], where the encryption key $K := y_T^k \pmod p$ is chosen in a multiplicative subgroup of order q of $\mathbf{Z}_p^*$. We might also consider the randomized RSA encryption scheme, i.e. $\mathcal{E}_T(m) := (c_1, c_2) = ((m+k)^e \pmod n, k^e \pmod n)$, which is more efficient for the sender, if used with small public exponent e (e.g. $e = 17$), but less efficient for the recipient of the ciphertext, who has to decrypt it as $c_1^d - c_2^d \pmod n$ and needs two (full) exponentiations to perform this step.

- *Authentic key exchange protocol* $\mathcal{K}$*:* The authentic key exchange protocol should generate a fresh, unique session key and at the same time offer implicit or better explicit user authentication. As an efficient scheme with implicit user authentication, we might use one of the schemes proposed in [MaTI86], which have two rounds and need two exponentiations for each party in order to obtain a fresh key. The key confirmation is performed during the use of the keys in the following.

  If the parties want to authenticate themselves explicitly, they could use one of the three protocols described in [LiLe95], which offer explicit key authentication and mutual key confirmation, but need three rounds.

  A more efficient two round protocol that uses digital signatures for explicit user authentication during key-exchange has been proposed in [Pete96]. It also offers unilateral key confirmation and needs one additional exponentiation for signature verification compared to the first protocol with implicit key authentication.

- *Pseudo-random number generator* $\mathcal{G}$*:* There are many ways to generate pseudo-random numbers from a given seed. We can first use a combination of linear feedback shift registers (LFSR). This is very efficient but there are no proofs of security. In fact, many of them have already been broken so that they cannot be used in secure applications. Nevertheless, they are interesting for implementation on cheap smart cards in electronic purse applications.

  A more secure approach consists in using complexity theory to prove the security of generators based on the same hard problems as in public-key cryptography. The

simplest and most efficient one is to our knowledge the Blum, Blum and Shub generator [BlBS86], also called quadratic residue generator. Another one, not based on a number theoretical problem, can be found in [FiSt96]. It is probably the best choice if a low cost smart card is used as electronic purse.

- *Coin C:* Tradeoff between *efficiency*, like a multi-spendable or divisible coin and *privacy*, i.e. the linkability of such payments. The linkability of coins might be worse, if parts of different coins are spent within the same payment at once. This might allow to link many more payments of a person than originally intended.

- *Size of coin:* The coins are chosen randomly and are related to the pseudonym PS_y. Therefore it suffices to choose a length that avoids the double choice of the same random value as long as PS_y is used, when applying the birthday paradox. In the case of micro-payments, the coin is the certified root of a hash chain and must therefore be of the same size as the hash value of the chosen hash function.

- *Payment:* For efficient payment, the non-interactive variant of the payment protocol can be used as long the signed message $mess$ and thus the generated signature is unique in order to prevent falsely framing of the shop.

To demonstrate the power of the general concept, we will scale our scheme for the two basic applications of *internet payment* and *electronic purse*, where the security and efficiency requirements are quite different.

7. Secure protocol for internet payment

If the user communicates to the shop via the internet using his connected personal computer, he is able to use *provably secure* and efficient primitives. This allows the use of the system for reasonable payments e.g. up to US$ 1000. To obtain this, we select different variants of the Schnorr signature scheme [Schn89]. The size of the signature parameters for the signature schemes $(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$ and $(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$ should be taken very large, as they need to be reliable for a long time. But the parameter size of $(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$ can be optimized according to the fastest known algorithms for computation of discrete logarithms, as the shop will deposit a coin after a short storage (e.g. once a day/week) at the bank. So it is always possible to increase the size of those parameters if new attacks are proposed.

As blind signature scheme $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$, we choose the Okamoto blind signature [Okam92], which has been proven to be computational secure [PoS96b] and which is also based on the basic Schnorr scheme. To obtain a hash value of accurate length, we choose the standard SHA-1 [NIST93, NIST95] for which no attacks have been reported. Finally, for $(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})$ we choose the ElGamal encryption scheme, which is more efficient for the trustee to decrypt in the case of fraud. As coin size, we choose a 48 bit number, which gives a collision probability of $\sim 2^{-24}$ applying the birthday paradox. At last, the quadratic residue generator proposed by Blum, Blum and Shub seems to be the best choice for the pseudo-random generator.

All those choices for the cryptographic tools needed in the system leads to an efficient and secure protocol for internet payment. We only use well known primitives with proved theoretical security. Figure 11 shows that a coin can be stored with 67 bytes and that each pseudonym needs 124 bytes. We give a detailed description of this specific protocol in the appendix.

- *signature schemes* $(\mathcal{G}_\mathcal{T}, \mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$, $(\mathcal{G}_\mathcal{U}, \mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$: Schnorr signature with large parameter size,
- *signature schemes* $(\mathcal{G}_\mathcal{C}, \mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$: Schnorr signature with smaller parameters,
- *blind signature scheme* $(\mathcal{G}_\mathcal{B}, \mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$: Okamoto blind signature scheme,
- *probabilistic encryption scheme* $(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})$: ElGamal encryption scheme,
- *authentic key exchange protocol* $\mathcal{K}$: key exchange with explicit user authentication [Pete96],
- *symmetric cryptosystem* $(\mathcal{E}_K, \mathcal{D}_K)$: Data Encryption Standard (DES),
- *hash function* h: Secure Hash Algorithm (SHA),
- *pseudo-random number generator* $\mathcal{G}$: quadratic residue generator [BlBS86].

Table 1: Choice of cryptographic tools for internet payment

8. Efficient protocol for electronic purse

The constraint for electronic purses is to allow payments at a shop using only a smart card. If the card is only used to compute σ_C at payment, this can be achieved by using a simple card that supports integer arithmetics. need a more powerful device, e.g. a wallet, under control of the user to verify σ_B, σ_T and generate σ_U. The second solution consists in using the electronic purse for registration and withdrawal e.g. at a bank terminal (POS). We focus on an efficient realization of this here.

The size of Schnorr signatures is quite small and they are efficient to generate, therefore we choose them as signature schemes $(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$, $(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$ and $(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$ (in the last scheme the exponentiation to obtain r-values can also be pre-computed [Schn89] or delegated using *use & throw coupons* [NRVR94] if the card has enough memory). To obtain hash values of suitable length, we use MD-5 [Rive92]. As signature scheme for the coin C we use CLE [Ster94], which can be efficiently implemented on a cheap smart card and which optimises the size of the pseudonyms even if the signature produced by this scheme are quite long, but have to be stored only in the bank's database, which is considered to have enough storage. After the registration, the card obtains a 160 bit keypair (PS_x, PS_y) for use with the zero-knowledge identification scheme CLE and the 288 bit Schnorr signature σ_T. So each pseudonym needs only 46 bytes to be stored in the card. As symmetric cryptosystem $(\mathcal{E}, \mathcal{D})$ we use DES with PS_x as session key for the user and trustee. The size of the coin is chosen as a 24 bit number, which allows to spend up to 2^{24} coins under the same pseudonym.

Using this system a card with only 5 KB of memory can store 58 pseudonyms and the same number of coins. If we allow multi-spendable coins and assume, that each coin is related to a unique pseudonym, the card can be used at 58 different shops with *no linkability* between the shops. So we obtain an efficient protocol for electronic purse which can protect at the same time the privacy of the honest users and all the entities against dishonest ones.

Figure 11 sums up the storage (in byte) needed for a coin and a pseudonym, for a prime modulus p of 64 byte and q of 20 byte as proposed e.g. in [NIST91].

- *signature schemes* $(\mathcal{G}_\mathcal{T}, \mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$, $(\mathcal{G}_\mathcal{U}, \mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$: Schnorr signature scheme,
- *signature scheme* $(\mathcal{G}_\mathcal{C}, \mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$: CLE,
- *blind signature scheme* $(\mathcal{G}_\mathcal{B}, \mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$: blind Schnorr signature scheme,
- *authentic key exchange protocol* $\mathcal{K}$: key exchange with implicit user authentication [MaTI86],
- *symmetric cryptosystem* $(\mathcal{E}_K, \mathcal{D}_K)$: DES,
- *hash function* h: MD-5,
- *pseudo-random number generator* $\mathcal{G}$: an LFSR based generator.

Table 2: Choice of cryptographic tools for electronic purse payment

	Internet payment		Electronic purse	
cryptographic tool	scheme	storage at user	scheme	storage at user
	temporary values			
$(\mathcal{S}_\mathcal{U}, \mathcal{V}_\mathcal{U})$	Schnorr	$p:64, q:20, x_U:20$	Schnorr	$p:64, q:20, x_U:20$
$(\mathcal{E}_\mathcal{T}, \mathcal{D}_\mathcal{T})/(\mathcal{E}_K, \mathcal{D}_K)$	RSA encr.	$N:64, y_T:2, e_T:64$	DES	$PS_x:7(10), e:8$
hash function h	SHA	20	MD-5	16
$(\mathcal{S}_\mathcal{B}, \mathcal{V}_\mathcal{B})$	Okamoto	$\sigma_B:60, C:6$	Schnorr	$\sigma_B:36, C:3$
$Ind(PS_y)$		1		1
Coin		67		40
$(\mathcal{S}_\mathcal{C}, \mathcal{V}_\mathcal{C})$	Schnorr	$PS_x:20, PS_y:64$	CLE	$PS_x:10$
$(\mathcal{S}_\mathcal{T}, \mathcal{V}_\mathcal{T})$	Schnorr	$\sigma_T:40$	Schnorr	$\sigma_T:36$
Pseudonym		124		46

Figure 11: Storage requirements of internet and electronic purse scheme w.r.t user

9. Conclusion

We presented an efficient payment system with anonymity revocation. It is the first scheme, that achieves off-line prevention of all kind of extortion attacks. Thereby we assumed that the attacks were of short duration and without physical involvement of the attacker, as otherwise no cryptographical protection is possible. Due to its scalable security w.r.t. efficiency, we were able to demonstrate secure realizations for an internet payment scheme as well as a highly efficient payment scheme for electronic purse applications.

References

[AbFu96] M.Abe, E.Fujisaki, "How to Date Blind Signatures", Advances in Cryptology: Proc. Asiacrypt'96, Lecture Notes in Computer Science 1163, Springer Verlag, (1996), pp. 244 – 251.

[AnKu96] R.Anderson, M.Kuhn, "Tamper Resistance - A Cautionary Note", Usenix Electronic Commerce Workshop, (1996), 11 pages.

[AnMS96] R.Anderson, C.Manifavas, C.Sutherland, "NetCard – A Practical Electronic Cash System", Lecture Notes in Computer Science 1189, Proc. Security Protocols Workshop, Springer Verlag, (1997), pp. 49 – 57.

[AJSW96] N.Asokan, P.Janson, M.Steiner, M.Waidner, "Electronic Payment Systems"', IBM, (1996), 16 pages.

[BGHH95] M.Bellare, J.Garay, R.Hauser, A.Herzberg, H.Krawczyk, M.Steiner, G.Tsudik, M.Waidner, 'iKP - A family of Secure Electronic Payment Protocols", Proc. 1st Usenix Workshop on Electronic Commerce, (1995).

[BlBS86] L.Blum, M.Blum, M.Shub, "A Simple Unpredictable Pseudo-Random Number Generator", SIAM Journal on Computing, Vol. 15, 2, (1986), pp. 364 – 383.

[BBCM94] J.Boly, A.Bosselaers, R.Cramer, R.Michelsen, S.Mjølsnes, F.Muller, T.Pedersen, B.Pfitzmann, P.de Rooij, B.Schoenmakers, M.Schunter, L.Vallée, M.Waidner, "The ESPRIT Project CAFE - High security Digital Payment Systems", Proc. Esorics '94, Lecture Notes in Computer Science 875, Springer Verlag, (1994), pp. 217 – 230.

[BüPf89] H.Bürk, A.Pfitzmann, "Payment systems enabling security and unobservability", Computers & Security, Vol. 8, (1989), pp. 399 – 416 .

[Bra93a] S.Brands, "An efficient off-line electronic cash system based on the representation problem", CWI Technical Report CS-R9323, Amsterdam, (1993), 77 pages.

[Bra93b] S.Brands, "Untraceable Off-Line Cash in Wallets with Observers", Lecture Notes in Computer Science 773, Advances in Cryptology: Proc. Crypto '93, Springer Verlag, (1994), pp. 302 – 318.

[Bran95] S.Brands, "Off-Line Electronic Cash Based on Secret-Key Certificates", Proc. of 2nd Int. Symposium of Latin American Theoretical Informatics (LATIN '95), (1995), 48 pages.

[BrGK95] E.Brickell, P.Gemmell, D.Kravitz, "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Exchange", Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), (1995), pp. 457–466.

[CaMS96] J.Camenisch, U.Maurer, M.Stadler, "Digital Payment Systems with Passive Anonymity-Revoking Trustees", Lecture Notes in Computer Science 1146, Proc. ESORICS'96, Springer Verlag, (1996), pp. 31 – 43.

[CaPS94] J.Camenisch, J.-M.Piveteau, M.Stadler, "An Efficient Payment System Protecting Privacy", Lecture Notes in Computer Science 875, Computer Security - ESORICS'94, Springer Verlag, (1995), pp. 201 – 215.

[CaPS95] J.Camenisch, J.-M.Piveteau, M.Stadler, "Faire anonyme Zahlungssysteme", Proc. GISI'95, Informatik aktuell, Springer Verlag, (1995), pp. 254–265.

[CaPS96] J.Camenisch, J.-M.Piveteau, M.Stadler, "An efficient Fair Payment System", Proc. 3rd ACM Conference on Computer and Communications Security, ACM Press, (1996), pp. 88–94.

[Chau81] D.Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, Vol. 24, No. 2, February, (1981), pp. 84 – 88.

[Chau82] D.Chaum, "Blind signatures for untraceable payments", Advances in Cryptology: Proc. Crypto '82, Plenum Press, (1983), pp. 199 – 203.

[Chau85] D.Chaum, "Security without identification: Transaction systems to make big brother obsolete", Communications of the ACM, Vol. 28, October, (1985), pp. 1030–1044.

[Chau87] D.Chaum, "Privacy protected payments: Unconditional Payer and/or Payee Anonymity", Smart Card 2000: The future of IC Cards, North-Holland, (1989), pp. 69–92.

[ChFN88] D.Chaum, A.Fiat, M.Naor, "Untractable electronic cash", Lecture Notes in Computer Science 403, Advances in Cryptology: Proc. Crypto '88, Springer Verlag, (1988) pp. 319 – 327 .

[ChP92a] D.Chaum, T.P.Pedersen, "Transferred cash grows in size", Lecture Notes in Computer Science 658, ACP Eurocrypt '92, Springer Verlag, (1993), pp. 357 – 367.

[ChP92b] D.Chaum, T.P.Pedersen, "Wallet databases with observers", Lecture Notes in Computer Science 740, Advances in Cryptology: Proc. Crypto '92, Springer Verlag, (1993), pp. 89–105.

[DFTY97] G.Davida, Y.Frankel, Y.Tsiounis, M.Yung, "Anonymity Control in E-Cash Systems", Proc. Financial Cryptography Workshop, February, (1997), 15 pages.

[Desm94] Y.Desmedt, "Threshold Cryptography", European Trans. on Telecommunications, Vol. 5, No. 4, (1994), pp. 35 – 43.

[ElGa85] T.ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Vol. IT-30, No. 4, July, (1985), pp. 469 – 472.

[Ferg93] N.Ferguson, "Single Term Off-Line Coins", Lecture Notes in Computer Science 765, Advances in Cryptology: Proc. Eurocrypt '93, Springer Verlag, (1993), pp. 318 – 328.

[FiSt96] J.B.Fischer, J.Stern, "An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding", Lecture Notes in Computer Science 1070, Advances in Cryptology: Proc. Eurocrypt'96, Springer Verlag, (1996), pp. 245 – 255.

[FrTY96] Y.Frankel, Y.Tsiounis, M.Yung, " "Indirect discourse Proofs": Achieving Efficient Fair Off-Line E-Cash", Lecture Notes in Computer Science 1163, Advances in Cryptology: Proc. Asiacrypt'96, Springer Verlag, (1996), pp. 286- 300.

[FrYu92] M.Franklin, M.Yung, "Towards Provably Secure Efficient Electronic Cash", Columbia University, Dept. of Computer Science, TR CUCS-018-92, April, (1992).

[FrYu93] M.Franklin, M.Yung, "Secure and efficient off-line electronic digital money", Lecture Notes in Computer Science 700, 20th Int. Colloquium on Automata, Languages and Programming (ICALP), Springer Verlag, (1993), pp. 265 – 276.

[FuOk96] E.Fujisaki, T.Okamoto, "Practical Escrow Cash System", Lecture Notes in Computer Science 1189, Proc. 1996 Cambridge Workshop on Security Protocols, June, Springer Verlag, (1997), pp. 33 – 48.

[GiSt94] M.Girault, J.Stern, "On the length of cryptographic hash-values used in identification schemes", Lecture Notes in Computer Science 839, Advances in Cryptology: Proc. Crypto '94, Springer Verlag, (1994), pp. 202 – 215.

[GoMR88] S.Goldwasser, S.Micali, R.Rivest, "A secure digital signature scheme", SIAM Journal on Computing, Vol. 17, 2, (1988), pp. 281 – 308.

[Hall95] P.Hallman-Baker, "W3C payment resources", internet draft, (1995), available at `http://www.w3.org/hypertext/WWW/Payments/overview.html`

[HaSW96] R. Hauser, M.Steiner, M.Waidner, "Micro-payments based on iKP", Proc. SECURICOM 96, 14th Worldwide Congress on Computer and Communications Security and Protection, (1996), pp. 67 – 82.

[HMP94a] P.Horster, M.Michels, H.Petersen, "Meta-ElGamal signature schemes", Proc. 2. ACM conference on Computer and Communications security, ACM Press, November, (1994), pp. 96 – 107.

[HMP94b] P.Horster, M.Michels, H.Petersen, "Meta-Message recovery and Meta-blind signature schemes based on the discrete logarithm problem and their applications", Lecture Notes in Computer Science 917, Advances in Cryptology: Proc. Asiacrypt '94, Springer Verlag, (1995), pp. 224 – 237.

[JaYu96] M.Jakobsson, M.Yung, "Revokable and Versatile Electronic Money", Proc. 3rd ACM Conference on Computer and Communications Security, ACM Press, (1996), pp. 76–87.

[JaYu97] M.Jakobsson, M.Yung, "Applying Anti-Trust Policies to Increase Trust in a Versatile E-Money System", Proc. Financial Cryptography Workshop, (1997), 21 pages.

[LiLe95] C.H.Lim, P.J.Lee, "'Several practical protocols for authentication and key exchange"', Information Processing Letters, Vol. 53, (1995), pp. 91 – 96.

[MaTI86] T.Matsumoto, Y.Takashima, H.Imai, "'On seeking smart public-key distribution systems"', Trans. IECE Japan, Vol. 69, No. 2, February, (1986), pp. 99 – 106.

[MeOV97] A.Menezes, P.C.van Oorschot, S.Vanstone, "Handbook of Applied Cryptography", CRC Press, (1997).

[M'Raï96] D.M'Raïhi, "Cost Effective Payment Schemes with Privacy Regulations", Lecture Notes in Computer Science 1163, Advances in Cryptology: Proc. Asiacrypt '96, Springer Verlag, (1996), pp. 266 – 275.

[MRPo97] D.M'Raihi, D.Pointcheval, "Off-Line Trustees and Revokability: a Framework for Efficient Electronic Money", manuscript, (1997), 15 pages.

[NRVR94] D.Naccache, D.M'Raihi, S.Vaudenay, D.Raphaeli, "Can D.S.A. be improved ? – Complexity Trade-Offs with the Digital Signature Standard", Lecture Notes in Computer Science 950, Advances in Cryptology: Proc. Eurocrypt '94, Springer Verlag, (1995), pp. 77 – 85.

[NBS 77] NBS FIBS PUB 46, "Data Encryption Standard", National Bureau of Standards, U.S. Department of Commerce, Jan. 1977.

[NIST91] National Institute of Standards and Technology, Federal Information Process. Standard, FIPS Pub XX: Digital Signature Standard (DSS), (1991).

[NIST93] National Institute of Standards and Technology, Federal Information Processing Standards Publication, FIPS Pub 180: Secure Hash Standard (SHA), 11. May, (1993), 12 pages.

[NIST95] National Institute of Standards and Technology, Federal Information Processing Standards Publication, FIPS Pub 180-1: Secure Hash Standard (SHA-1), 17.April, (1995), 14 pages.

[NyRu94] K.Nyberg, R.Rueppel, "'Message recovery for signature schemes based on the discrete logarithm problem"', Lecture Notes in Computer Science 950, Advances in Cryptology: Proc. Eurocrypt '94, Springer Verlag, (1994), pp. 182 – 193.

[Okam92] T.Okamoto, "Provable secure and practical identification schemes and corresponding signature schemes", Lecture Notes in Computer Science 740, Advances in Cryptology: Proc. Crypto'92, Springer Verlag, (1993), pp. 31–53.

[Okam95] T.Okamoto, "An Efficient Divisible Electronic Cash Scheme", Lecture Notes in Computer Science 963, Advances in Cryptology: Proc. Crypto'95, Springer Verlag, (1995), pp. 438 – 451.

[OkOh91] T.Okamoto, K.Ohta, "Universal electronic cash", Lecture Notes in Computer Science 576, Advances in Cryptology: Proc. Crypto '91, Springer Verlag, (1992), pp. 324–337.

[Pede96] T.Pedersen, "Electronic Payments of Small Amounts", Lecture Notes in Computer Science 1189, Proc. Security Protocols Workshop, Springer Verlag, (1997), pp. 59 – 68.

[Pete96] H.Petersen, "'Digital signature schemes based on the discrete logarithm problem and their applications"', (in German), Dissertation, University of Technlogy Chemnitz-Zwickau, Shaker Verlag, (1996), 277 pages.

[PeHo97] H.Petersen, P.Horster, "'Self-certified keys – Concepts and Applications"', to appear in Proc. 3rd Int. Conference on Communications and Multimedia Security, Chapman & Hall, September, (1997), 15 pages.

[PfWa95] B.Pfitzmann, M.Waidner, "Strong Loss Tolerance for Untraceable Electronic Coin Systems", Hildesheimer Informatik-Berichte 15/95, (1995), 44 pages.

[PfWa96] B.Pfitzmann, M.Waidner, "Properties of Payment Systems: General Definition Sketch and Classification", IBM Research Report RZ 2823, IBM Research Division, (1996), 29 pages.

[PoS96a] D.Pointcheval, J.Stern, "Security Proofs for Signatures", Lecture Notes in Computer Science 1070, Advances in Cryptology: Proc. Eurocrypt'96, Springer Verlag, (1996), pp. 387 – 398.

[PoS96b] D.Pointcheval, J.Stern, "Provably Secure Blind Signature scheme", Lecture Notes in Computer Science 1163, Advances in Cryptology: Proc. Asiacrypt'96, Springer Verlag, (1996), pp. 252 – 265.

[PoSt97] D.Pointcheval, J.Stern, "New Blind Signatures Equivalent to Factorization", Proc. 4th ACM Conference on Computers and Communications Security, ACM Press, (1997), 8 pages.

[RaGV97] C.Radu, R.Govaerts, J.Vandewalle, "Efficient Electronic Cash with Restricted Privacy", Proc. Financial Cryptography Workshop, (1997), 8 pages.

[Rive92] R.Rivest, "'The MD5 Message Digest Algorithm"', RFC 1321, April, (1992).

[RiSA78] R.L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Comm. of the ACM, Vol. 21, (1978), pp. 120–126.

[RiSh84] R.L.Rivest, A.Shamir, "How to expose an Eavesdropper", Comm. of the ACM, Vol. 27, (1984), pp. 393–395.

[RiSh96] R.Rivest, A.Shamir, "PayWord and MicroMint: Two simple micropayment schemes", Lecture Notes in Computer Science 1189, Proc. Security Protocols Workshop, Cambridge, Springer Verlag, (1997), pp. 69 – 87.

[RuOo94] R.Rueppel, P.C.can Oorschot, "Modern key agreement techniques", Computer Communications, Vol. 17, Vol. 7, (1994), pp. 458 – 465.

[Schn89] C.P.Schnorr, "Efficient identification and signatures for smart cards", Lecture Notes in Computer Science 435, Advances in Cryptology: Proc. Crypto '89, Springer Verlag, (1990), pp. 239–251.

[Scho95] B.Schoenmakers, "An efficient electronic payment system withstanding parallel attacks", CWI Technical Report CS-R9522, (1995), 14 pages.

[SET 96] Mastercard, VISA, "Secure Electronic Transactions", Draft, June, (1996).

[Sham89] A.Shamir, "An efficient Identification Scheme Based on Permuted Kernels", Lecture Notes in Computer Science 435, Advances in Cryptology: Proc. Crypto '89, Springer Verlag, (1990), pp. 606 – 609.

[SoNa92] S.von Solms, D.Naccache, "Blind signatures and perfect crimes", Computers & Security, Vol. 11 , (1992), pp. 581 – 583.

[Sta96b] M.Stadler, "Cryptographic Protocols for Revocable Privacy", Dissertation ETH No. 11651, ETH Zürich, (1996), 111 pages.

[StPC95] M.Stadler, J.-M.Piveteau, J.Camenisch, "Fair-Blind Signatures", Lecture Notes in Computer Science 921, Advances in Cryptology: Proc. Eurocrypt '95, Springer Verlag, (1995), pp. 209 – 219.

[Ster94] J.Stern, "Designing identification schemes with keys of short size", Lecture Notes in Computer Science 839, Advances in Cryptology: Proc. Crypto, Springer Verlag, (1995), pp. 164 – 173.

[WaPf89] M. Waidner, B.Pfitzmann, "Loss-tolerant Electronic Wallet", Proc. Smart Card 2000, North-Holland, (1991), pp. 127 – 150.

[Waid96] M.Waidner, "Development of a Secure Electronic Marketplace for Europe", Proc. ESORICS '96, Lecture Notes in Computer Science 1146, Springer Verlag, (1996), pp. 1–14.

[Wayn96] P.Wayne, "Digital Cash: Commerce on the Net", Academic Press, (1996).

[YeLL94] S.-M.Yen, C.-S.Laih, A.K.Lenstra, "'Multi-exponentiation"', IEE Proc.-Comput. Digit. Tech., Vol. 141, No. 6, November, (1994), pp. 325 – 326.

A Example of payment scheme

User	**Opening an account**	Bank
ID_U	$\xrightarrow{\text{Auth}}$	ID_U
acc_U	$\xleftarrow{acc_U}$	store (ID_U, acc_U) in U-DB

User	**Registration**	Trustee
$k_1, PS_x, k_U \in_R \mathbf{Z}_q$		$k_2, k_T \in_R \mathbf{Z}_q$
$r_1 := \alpha^{k_U} \pmod p$	$\xrightarrow{r_1}$	r_1
	$\xleftarrow{r_2}$	$r_2 := \alpha^{k_2} \pmod p$
r_2		
$K_{U,T} := h\left(r_2^{x_1} \cdot y_2^{k_1} \pmod p\right)$		$K_{U,T} := h\left(r_1^{x_2} \cdot y_1^{k_2} \pmod p\right)$
$PS_y := \alpha^{PS_x} \pmod p$		
$r_U = \alpha^{k_\sigma} \pmod p$		
$s_U := x_U \cdot h(r_U, ID_U, PS_y)$ $+k_U \pmod q$		
	$\xrightarrow{ID_U, \mathcal{E}_{K_{U,T}}(PS_y, \sigma_U)}$	
$\sigma_U := (r_U, s_U)$		ID_U, PS_y, σ_U
		signature accepted, if
		$\alpha^{s_U} \stackrel{?}{\equiv} y_U^{h(r_U, ID_U, PS_y)} \cdot r_U \pmod p$
		$r_T = \alpha^{k_T} \pmod p$
		$s_T := x_T \cdot h(r_{PS}, PS_y) + k_{PS} \pmod q$
	$\xleftarrow{E(\sigma_T)}$	
σ_T		$\sigma_T := (r_T, s_T)$
signature accepted, if		
$\alpha^{s_T} \stackrel{?}{\equiv} y_T^{h(r_T, PS_y)} \cdot r_T \pmod p$		
store PS_x, PS_y, σ_T in PS$_x$-DB		store $(ID_U, PS_y, \sigma_U, Ind(x_T))$ in PS$_y$-DB

User	**Withdrawal**	Bank
$k_3 \in_R \mathbf{Z}_q$		$k_4 \in_R \mathbf{Z}_q$
$r_3 := \alpha^{k_U} \pmod p$	$\xrightarrow{ID_U, r_3}$	ID_U, r_3
	$\xleftarrow{r_4}$	$r_4 := \alpha^{k_4} \pmod p$
r_4		
$K_{U,B} := h\left(r_4^{x_3} \cdot y_4^{k_3} \pmod p\right)$		$K_{U,B} := h\left(r_3^{x_4} \cdot y_3^{k_4} \pmod p\right)$
$C \in_R [0 : 2^{24}]$		$k_B \in_R \mathbf{Z}_q$
$e_T := (h(PS_y, C) + k_5)^e \pmod{n_T}$	$\xrightarrow{\mathcal{E}_{K_{U,B}}(e_T, k_5^e)}$	e_T, k_5^e
	$\xleftarrow{\mathcal{E}_{K_{U,B}}(\tilde{r}_B)}$	$\tilde{r}_B := \alpha^{k_B} \pmod p$
$\tilde{r}_B$		
$u, v \in_R \mathbf{Z}_q^*$		
$r_B := \tilde{r}_B^u \cdot \alpha^v \pmod p$		
$e := H(r_B, C, PS_y)$		
$\tilde{e} := e/u \pmod q$	$\xrightarrow{\mathcal{E}_{K_{U,B}}(\tilde{e})}$	$\tilde{e}$
	$\xleftarrow{\mathcal{E}_{K_{U,B}}(\tilde{s}_B)}$	$\tilde{s}_B := x_B \cdot \tilde{e} + \tilde{k}_B \pmod q$
$\tilde{s}_B$		
$s_B := \tilde{s}_B \cdot u + v \pmod q$		
$\sigma_B := (r_B, s_B)$		$(ID_U, Ind(x_B), \tilde{C}, e_T)$
		stored in W-DB
signature accepted, if		
$\alpha^{s_B} \stackrel{?}{\equiv} y_B^{h(r_B, C, PS_y)} \cdot r_B \pmod p$		debit of acc_U
store $(C, \sigma_B, Ind(PS_y))$ in C-DB		

Concrete choices of the protocols with Schnorr-signature scheme

User	**Payment**	Shop
$mess$	$\xleftarrow{mess}$	$mess := f(ID_S, time, \ldots)$
$k_C \in_R \mathbf{Z}_q^*$ $r_C = \alpha^{k_C} \pmod p$ $s_C := PS_x \cdot h(r_C, C, ID_S, mess, PS_y) + k_C \pmod q$		
$\sigma_C := (r_C, s_C)$	$\xrightarrow{C, PS_y, \sigma_B, \sigma_T, \sigma_C}$	$C, PS_y, \sigma_B, \sigma_T, \sigma_C$ $\alpha^{s_T} \stackrel{?}{\equiv} y_T^{h(r_T, PS_y)} \cdot r_T \pmod p$ $\alpha^{s_B} \stackrel{?}{\equiv} y_B^{h(r_B, C, PS_y)} \cdot r_B \pmod p$ $\alpha^{s_C} \stackrel{?}{\equiv} PS_y^{h(r_C, C, ID_s, mess, PS_y)} \cdot r_C(p)$ store $(C, PS_y, mess, \sigma_B, \sigma_T, \sigma_C)$
		<u>after user extortion:</u> $PS_y \in$ U-BL $\Rightarrow$ reject <u>after bank extortion:</u> $y_B \in$ B-BL and $C \notin$ C-WL $\Rightarrow$ reject <u>after trustee extortion:</u> $y_T \in$ T-BL and $PS_y \notin$ PS-WL $\Rightarrow$ reject

Shop	**Deposit**	Bank
$C, PS_y, \sigma_B, \sigma_T$	$\xrightarrow{C, PS_y, \sigma_B, \sigma_T}$	$C, PS_y, \sigma_B, \sigma_T$ $\alpha^{s_T} \stackrel{?}{\equiv} y_T^{h(r_T, PS_y)} \cdot r_T \pmod p$ $\alpha^{s_B} \stackrel{?}{\equiv} y_B^{h(r_B, C, PS_y)} \cdot r_B \pmod p$ C already deposited under PS_y ? search $(C, \sigma'_C) \in$ D-DB
σ'_C	$\xleftarrow{\sigma'_C}$	Yes: send σ'_C, No: acknowledge
$\sigma_C \neq \sigma'_C$?	$\xrightarrow{ID_S, acc_S, mess, \sigma_B}$	$ID_S, acc_S, mess, \sigma_C$ $\alpha^{s_C} \stackrel{?}{\equiv} PS_y^{h(r_C, C, ID_s, mess, PS_y)} \cdot r_C(p)$ C overspent under PS_y ? Yes: *user tracing* protocol No: credit of acc_S by value of C store $(C, PS_y, ID_S, \sigma_T, \sigma_C)$ in D-DB

Concrete choices of the protocols with Schnorr-signature scheme

The following table summarizes the number of exponentiation for the different parties, which are all computed in multiplicative subgroup of order q, i.e. can be implemented with an average of $1.5|q|$ multiplications (respectively $1.75|q|$ multiplications for the multiexponentiations) [YeLL94].

Phase	User	Trustee	Bank	Shop
Opening an account	–	–	–	–
Registration	5	3	–	–
Withdrawal	5	–	4	–
Payment	1	–	–	3
Deposit	–	–	3	–

Table 3: Efficiency of the example payment protocol