Distributed Trustees and Revocability: a Framework for Internet Payment

[Published in R. Hirschfeld, Ed., *Financial Cryptography*, vol. 1465 of *Lecture Notes in Computer Science*, pp. 28–41, Springer-Verlag, 1998.]

David M'Raïhi[1] and David Pointcheval[2]

[1] GEMPLUS, Cryptography Department, 92447 Issy-les-Moulineaux, France
E-mail: David.Mraihi@gemplus.com
[2] GREYC, Université de Caen, 14032 Caen Cedex, France
and LIENS, École Normale Supérieure, 75230 Paris Cedex 05, France
E-mail: David.Pointcheval@info.unicaen.fr

Abstract. From von Solms and Naccache's standpoint, constructing a practical and secure e-money system implies a proper regulation of its privacy level. Furthermore, when the system benefits from a widely connected communication network, tuning precisely this control for achieving efficiency without endangering security is a hard task. In order to solve this specific problem, we propose an e-cash scheme based on the usage of provably secure primitives, where trustee quora are in charge of privacy control. Moreover, Trustees remain off-line throughout the e-coin's life to reduce the communication flow and improve the resulting scheme performance.

1 Introduction

Reaching the end of the twentieth century, our society is deeply engaged in a vast technologic revolution. The huge growth of digital communications and mobile technologies reflects this transformation, a paper-based society changing to an electronic media world. The introduction of public-key cryptography [9] opened the door for capital additions to this construction of a technological-oriented society. Digital signature [22, 12, 24] is obviously one of the most significant examples of this major contribution.

Electronic cash, originally based on a variation of the digital signature paradigm, blind signatures [7], is a practical aspect of the continuous mutation we are living now. The core idea is to mimic metal coins, by delivering electronic coins that users could also spend anonymously. Another important goal consists in avoiding calling the bank at payment time to prevent double-spending. In Chaum's original proposal, the bank had to check every deposited coin against the list of spent coins. Shop's guarantee that a coin received is a valid coin therefore required the bank to support a real-time payment architecture, at a huge investment cost in terms of computation and communication capacities.

Chaum, Fiat and Naor [8] proposed to add detection mechanisms to solve this particular issue. They introduce the first off-line electronic cash scheme, based on zero-knowledge proofs and cut-and-choose techniques. Hence, the first *practical* electronic cash system [8] would provide privacy and security, but at a huge computational cost.

Nevertheless, digital age is not the perfect age and as digital technologies were growing on, Evil found its path through a new mutation: digital crime. Thus, anonymity granted by blind signatures could lead to various criminal activities [26, 2, 4]. Considering potential attacks from large-scale criminal organizations, introducing the concept of revocability is a natural approach. Basically, the idea is to give the control of all privacy issues to a trusted entity, being any combination of different parties such as judges, users' associations or governmental representatives.

Related work: Escrowed cash introduced in [2] as schemes [5] based on the fair blind signature primitive [4] give a good flavor of the concept but required the presence of Trustee during withdrawals thereby decreasing drastically overall performance of the scheme. A new model [13], solving the bank robbery attack by implementing a secret channel between the bank and the trustee and introducing the concept of challenge semantic, leads to reconsider security, scalability and flexibility topics of revocable e-cash schemes. A very interesting technique based on a modification of DSS [14] proposed a distributed architecture for the trustees; this paper mainly concentrate on the protocol aspect as a basic block for scheme construction. The attack model and the impact on security are analyzed in [15]. Recent works introduce the first revocable off-line (with respect to the Trustees) e-cash schemes, based on proofs of knowledge and equality of discrete logarithms [3] or on indirect discourse proofs [11]. In this setting, the idea is to reduce the communication burden while preventing most of the possible attacks. Trustees never participate in protocols related to the normal usage of coins: they are only involved in tracing operations.

Our solution is smart-card oriented, taking into account the main advances in this field, namely the possibility to achieve public-key operations efficiently. We also wanted to emphasize the impact of the network structure in terms of communication and security; using only provably secure primitives is eventually a new contribution to the promotion of prudently designed e-cash schemes.

Achievements: In this paper, we extend [17] introducing revocability over a distributed communication network, *i.e.* where trustees are distributed over a network as Internet. Such a structure provides both high resistance to attacks and faults as well as various trade-offs in terms of computation, communication cost and memory requirements. Our main concern is to focus on the user's side and limit its technical requirements, specifically computational and communication requirements. The second objective was to reduce the level of trust implied by [17] while respecting our primary purpose. We concentrate therefore on the network topology and security aspects, investigating several solutions. In partic-

ular, the primitives [23, 24, 21] considered in our scheme are provably secure in order to enhance security analysis.

The main principles are:

1. Usage of Pseudonyms [6]: users are able to communicate anonymously with Trustees and Payees. Using [17] techniques, Pseudonyms (in short Ps) are derived from public user IDs I only a Trustee (in short TTP) subset knows the link between I and Ps. Furthermore, those pseudonyms offer the possibility to exchange coins (received from other users) and have the bank refresh coins (when validity date expires) without revealing any user ID-related. Obviously, by revealing a couple (I, Ps), Trustees enable payment tracing.
2. Combined certification of Ps (by TTP) and e-coins (by the bank): such a double certification enables TTPs to remain off-line during all coins-life related actions: withdrawal, payment, deposit, transfer and refreshment. TTPs interact with users only at the account opening stage[1].
3. Distribution of the Trustees: a collaboration of k trustees is required for any operation related to Ps certification. Privacy control is ruled by a quorum of trustees and as long as k trustees remain honest, user and transaction tracings are possible. Finally, the presence of any subset of k trustees is required to prove that a coin is related to a transaction, giving honest users an additive protection against a malevolent trustee.

2 Communication Models

Different constructions are possible (see figure 1):

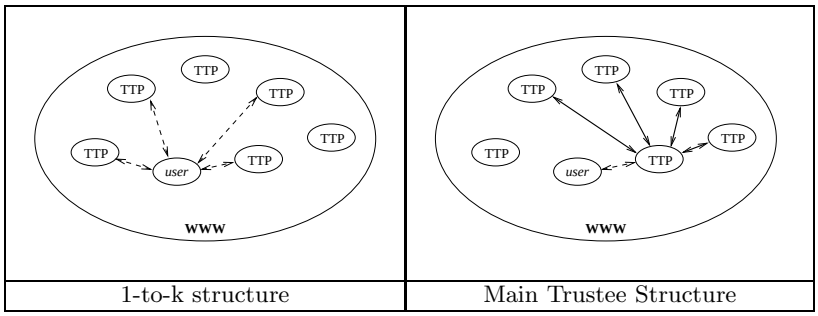


Fig. 1. Communication Model

– Basic 1-to-k Structure: a user contacts all trustees and engages protocols with them; a clear bottleneck of this solution is the transmission rate between the user and TTPs, since we can assume that the user's communication routines

[1] which is not the case in [3] and [11] schemes where Trustees collaboration is only required in case of disputes or overspending detection

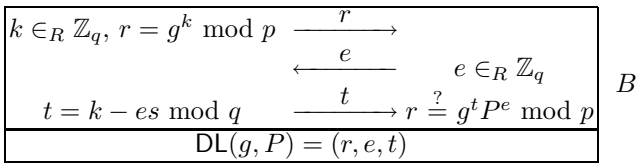
run on low-cost devices. On the other hand, user's control is simplified since he initiates all communications.
– Main Trustee Structure: a user initiates communication with a trustee T_i (chosen at random among the k trustees) and delegates all the other tasks to T_i. In this setting, T_i is used as a gate to the global network of trustees; communication speed is therefore improved due to the usage of T_i fast transmission facilities.

3 The Basic Scheme

3.1 Primitives

The proposed scheme uses various primitives for authentication of users, issuing and verification of credentials, signature of transaction transcripts and encryption of privacy related information. These basic blocks are (where p and q are large prime integers such that $q \mid p-1$, and g is an element of $\mathbb{Z}_p$ of order q):

1. User's Identification (protocol initiation): Schnorr identification scheme [23] whose security has been proven to be equivalent to the discrete logarithm problem (even against active attacks [25]);



$\mathsf{DL}(g, P)$ proves to B that A knows s such that $P = g^s \bmod p$.

2. Signature (transaction and I certification) – Signature: the signature [24] derived from Schnorr's identification protocol. An existential forgery under an adaptative attack is equivalent to solving the underlying discrete logarithm problem [20];
3. Verifiable Secret Sharing (distributed trustees): let s be a secret key and $P = g^s \bmod p$ the associated public key. We want to distribute s among n participants in such a way that only a collusion of k of them might retreive, or at least use, s. Let Q be a random polynomial of degree $k-1$ over $\mathbb{Z}_q$ such that $Q(0) = s$: $Q(x) = a_{k-1}x^{k-1} + \ldots, a_1 x + s$. The secret s can then be shared among the n participants secretly distributing $s_i = Q(i)$ to the i^{th} one and broadcasting $P_i = g^{s_i} \bmod p$ for $i = 1, \ldots, n$. For any subset E of $\{1, \ldots, n\}$, and any $j \in E$, let us denote by $L_{E,j}$ the Lagrange's polynomial:

$$L_{E,j}(x) = \prod_{i \in E \setminus \{j\}} \frac{i-x}{i-j} \text{ so that } L_{E,j}(i) = 0 \ (\forall i \in E \setminus \{j\}) \text{ and } L_{E,j}(j) = 1.$$

Therefore, for any subset E of $\{1, \ldots, n\}$ with at least k elements,

$$Q(x) = \sum_{j \in E} s_j L_{E,j}(x), \text{ and so } s = \sum_{j \in E} \alpha(E, j) s_j \text{ where } \alpha(E, j) = L_{E,j}(0).$$

Furthermore, each participant can verify that his share is correctly related to $P_j = g^{s_j} \bmod p$, and that the secret can be properly rebuilt from the shares:

$$P = g^s = \prod_{j \in E} (g^{s_j})^{\alpha(E,j)} = \prod_{j \in E} P_j^{\alpha(E,j)} \bmod p.$$

Then $\mathsf{VSS}(s) = (s_1, \ldots, s_n)$.

4. Distributed Computation (Ps computation) – Dist-Comp: let us denote by X and s_j, for $j = 1, \ldots, n$, respectively, the secret key and shares $\mathsf{VSS}(X)$. Let E be a subset of k trustees who want to secretly compute $J = I^X \bmod p$. By broadcasting their shares $J_j = I^{s_j} \bmod p$ they get:

$$J = \prod J_j^{\alpha(E,j)} \bmod p = \mathsf{Dist\text{-}Comp}(X, I).$$

5. Shared Signatures (Ps certification): let us denote by $X, Y = g^X \bmod p$ and s_j, for $j = 1, \ldots, n$, respectively, the secret key, the public key and shares $\mathsf{VSS}(X)$. Let E be a subset of k trustees. They each choose a random element $k_j \in_R \mathbb{Z}_q$ and broadcast $r_j = g^{k_j} \bmod p$ and all compute:

$$r = \prod r_j^{\alpha(E,j)} = g^{\sum k_j \alpha(E,j)} = g^k \bmod p, \text{ where } k = \sum k_j \alpha(E,j) \bmod q.$$

They can compute the challenge $e = H(m, r)$, where m is the message to be signed. Then they compute their part of the signature: $t_j = k_j - e s_j \bmod q$. The signature of m is the triple (r, e, t) where $t = \sum \alpha(E,j) t_j \bmod q$:

$$g^t Y^e = \prod g^{\alpha(E,j) t_j} g^{e\alpha(E,j) s_j} = \prod g^{\alpha(E,j)(t_j + e s_j)} = \prod g^{\alpha(E,j) k_j} = r \bmod p.$$

This protocol $\mathsf{Sh\text{-}Sig}(X, m)$ provides a Schnorr signature (r, e, t) in such a way that no participant learns anything about others participants' secrets.

6. Blind Signature (coin certification) – Bl-Sig: the Okamoto-Schnorr blind scheme [18, 21]. If the secret key is denoted by $S = (x, z)$ and the public one by $P = y = g_1^x g_2^z \bmod p$, the signature the user gets is a tuple $(\varepsilon, \rho, \sigma)$ such that $\varepsilon = H(g_1^\rho g_2^\sigma y^\varepsilon, m)$, where m is the message to be blindly signed. This protocol offers a nice property for e-cash scheme construction: one-more forgery (*i.e.* generating one more signature) is infeasible;
7. Encryption (private channel between bank and users) – Cipher: El Gamal encryption scheme [12] which security is equivalent to the Diffie-Hellman problem, proven [9] equivalent in almost all cases to the discrete logarithm problem [16].
8. Shared Proof of Equality of Discrete Logarithms (privacy revocation) – $\mathsf{SEqDL}(I, J, g, Y)$: The n trustees share a secret X into s_j (for $j = 1, \ldots, n$) as described above and $Y = g^X \bmod p$. The user possesses a secret key s and $I = g^s \bmod p$. Furthermore, $J = Y^s = I^X \bmod p$. The trustees want to prove that $\log_I J = \log_g Y$. In order to achieve this goal, they randomly choose a secret $k_j \in \mathbb{Z}_q$, broadcast $u_j = I^{k_j} \bmod p$ and $v_j = g^{k_j} \bmod p$. They can compute $u = \prod u_j^{\alpha(E,j)}$ and $v = \prod v_j^{\alpha(E,j)}$ as well as the challenge

$e = H(u, v)$. Then, they broadcast $t_j = k_j - es_j \bmod q$. Finally, if we compute $t = \sum t_j \alpha(E, j) \bmod q$, it satisfies

$$u = I^t J^e \bmod p \text{ and } v = g^t Y^e \bmod p.$$

The triple $\mathsf{SEqDL}(I, J, g, Y) = (u, v, t)$ provides a proof of equality of the discrete logarithms $\log_I J = \log_g Y$, without revealing anything.

3.2 Protocols

In this section, the different protocols involved in the scheme are presented: registration (where a user obtains a set of Pseudonyms for protecting his privacy) and the different actions related to a financial transaction, namely withdrawal of coins, payment of purchases and deposit of transaction transcripts.

Registration Opening an account consists in two distinct phases (see figure 5) where a user interacts with the bank and a subset of k trustees:

1. Bank: the user proves his identity by exhibiting a "physical" proof such as a passport or any official document. He generates and sends his public identity $I = g^s$, where I represents an El Gamal-like public key. The bank stores I and the user's real identity ID and sends back the related certificate $Sig = \mathsf{Signature}_B(I)$.
2. TTPs: the user interacts with k TTPs to get his pseudonyms. The TTPs share the knowledge of:
   - a master secret key X splitted into n sub-keys $(s_1, \ldots, s_n) = \mathsf{VSS}(X)$.
   - π keys X_j, each distributed in n sub-keys $(s_{j,1}, \ldots, s_{j,n}) = \mathsf{VSS}(X_j)$ for $j = 1, \ldots, \pi$.

   We denote by E the subset of the k TTPs contacted by the user.
   The user first proves his knowledge of the secret information s related to I. Every TTP delivering his share of the pseudonyms $J_{j,i} = I^{s_{j,i}} \bmod p$, they can compute the pseudonyms $J_j = \prod_i J_{j,i}^{\alpha(E,i)} = I^{X_j} = Y_j^s \bmod p$, and produce a shared signature of J_j. The triple (J_j, e_j, t_j) corresponds to a certified pseudonym Ps_j and satisfies $e_j = H(g^{t_j} Y^{e_j}, J_j)$. Eventually, TTPs store in the registration log file the set $(I, \{Ps_j\}_{j \le \pi})$ to be able to revoke privacy when required.

Observation: Any TTP, merely reading the registration log file, could link I to a specific transaction since the user must "pseudo-signs" (using Ps_j) to spend coins. However, only a quorum of any k trustees can prove this link, as we will see below.

Withdrawal In order to withdraw coins (see figure 2) the user first sends I and proves his knowledge of s such that $I = g^s \bmod p$ [23]. Then, the bank blindly signs a coin in which the user embeds the public part of one of his pseudonyms J_j.

Obviously, such a coin is not traceable by the bank but the different spendings related to this coin are linkable. The value of each coin is represented by a counter which must be controlled before any payment (to avoid overspending).

Observation: Data required to rebuild the coin signature are encrypted by the bank, using I as an El Gamal public key. This additional protection certifies that only a user knowing s can recover J_j signature and improves the protocol's robustness.

Payment During payment (see figure 3), the payer sends Ps_j and a coin C, after verification of the associated counter, to the payee who checks Ps_j certificate and C validity. The payer generates the transaction signature, proving that he knows s such that $J_j = g^{X_j s} = {Y_j}^s$, with a challenge depending on the amount, the pseudonym, the coin and the "name" of the payee.

Observation: The payee can decide whether he prefers to deposit the coin or transfer it: if he declares $Name$ to be his public identity I, he must deposit the coin at the bank on his non-anonymous account; if he defines it to be one of his Pseudonyms, he can transfer transactions, with the help of the bank, into a new anonymous coin.

Deposit A user must deposit a transaction which field $Name = I$ (see figure 4). Basically, the user sends a transaction τ then the bank checks data validity, performs overspending verification and credits the corresponding account.

Transfer If a payee has associated one of his Pseudonyms to several transactions, he must transfer them with the help of the bank to obtain a coin corresponding to the total transaction amount:

1. the user sends the transactions associated to his Pseudonym J'_j
2. the bank checks their validity
3. the bank checks that the user knows the associated secret key
4. the bank generates a new coin C' linked to the transfered transactions The link with the transactions is necessary to enable the bank to prove a possible overspending.

Observation: This protocol is a straightforward concatenation of the deposit and withdrawal protocols in order to minimize computations.

Refreshment A refreshment protocol is possible in order to enable a user to exchange coins whose validity date is near expiration. The new coin cumulates the corresponding amounts and is associated to the previous ones by using the same random value, in order to guarantee correct overspending verification. As above, this protocol is simply the combination of a deposit and a withdrawal.

Privacy revocation

- *Payment-based Tracing:* Upon overspending detection, the bank issues the list of transactions and sends them to the TTPs center. After verifying the bank's claim, the TTPs return the identity I associated to the J_j included in the transactions together with the proof that $\log_I J_j = \log_g Y_j \ (= X_j)$.
- *Withdrawal-based Tracing:* In this situation, the user requiring protection against abuse (such as a criminal forcing him to withdraw anonymously e-coins and reveal the Ps related to the secret key s) will give the Ps corresponding to the withdrawal session and prove that he knows s (in order to avoid false accusations by a malevolent user knowing a certain Ps). The pseudonym is blacklisted to identify related-coins on-the-fly and block the transaction.

4 Security Analysis

In this section, we sketch the different security rationale of our scheme.

4.1 Forgery

A money forgery attack consists in a coalition of payers, payees and trustees making extra-money from the original pool of electronic coins certified by the bank or modifying coin values. Consider two possibilities:

1. transform a bank signature on a coin with value a into a signature on a coin with value a' where $a' > a$: we assume that the bank's secret keys have been properly generated (*i.e.* randomly) to avoid any correlation between keys. Any other manipulation is equivalent to possibility 2;
2. build a new certified coin from the public view of the protocols: this is equivalent to generate more coins than what allowed. The usage of a blind signature based on the witness indistinguishability guarantees the bank against such a forgery [21].

4.2 Bank robbery

We consider that the attack can either consist in simply forcing the bank to deliver blindly certified coins or even stealing the bank's keys by any mean (a physical attack of the bank system or kidnapping the bank manager). In order to prevent such an attack, two kinds of techniques can be applied:

- the bank stores any withdrawal [19] she properly completes, until their expiration date, and TTPS to periodically blind-certify the list. If a robbery occurs, the bank replaces its keys, and asks everybody to refresh their coins. The refreshment is performed with the help of TTPs who control, in the previous list, whether the coins have been fairly withdrawn. The logical consequence is that the thief cannot spend his coins, otherwise he will be discovered.

– after any withdrawal, the user asks TTPs to perform a shared signature of his new coin. Next, this certificate will be required for any transaction. In case of bank robbery, TTPs stop certifying coins which contain stolen keys of the Bank.

4.3 Privacy

Obviously, privacy protection provided by our scheme is only conditional, since users' untraceability is revocable and relies on the difficulty of the Diffie-Hellman problem [9]. Nevertheless, this problem has been proven to be equivalent in almost all instances to the discrete logarithm problem [16]. One may also observe that privacy is restricted by the number of Ps since transactions related to a certain J_j are linkable; but the bank cannot link these transactions to I anyway.

Private channel The very nature of I and Ps enables the bank to communicate securely with a user by El Gamal encryption (with $I = g^s$ during withdrawal and $Y_j = J_j^s$ during transfer) to prevent other users (I' or Y_j') from eavesdropping and mounting a very basic active attack: ask the question e' instead of the e from I or Y_j. A similar mechanism at registration time protects the user from the bank trying to discover a link between user's identity and his pseudonyms when the user sends I to the TTPs: since I is probabilistically encrypted, the bank cannot correlate $\{Ps_j\}$ to any known I.

Privacy revocation

Theorem 1. *The scheme achieves overspending robustness.*

Proof. A user has to sign a transaction in order to spend a coin; given that the user's signature is existentially unforgeable, it is infeasible for an attacker to generate a different signature for a given transaction.

Theorem 2. *The scheme achieves revocable privacy: only the bank and at least k TTPs can prove that a transaction was issued by a user whose identity is ID.*

Proof. First, observe that at withdrawal time, the user sends his public identity I but obtains a blind signature on J_j, that will be associated with further transactions performed by the user. Therefore, any coin related to J_j is spent anonymously. The bank can neither link any transaction to a specific I (since a transaction is linked to J_j that the bank blindly signs during withdrawal) nor trace a coin. On the contrary, the bank and any TTP can easily link a transaction to user's identity:

– assuming that the bank detects overspending of coin C, she presents the related transaction τ to any TTP who extracts J_j and looks-up the corresponding I in the database. This TTP reveals I to the bank who can identify the user responsible of the fraud. However, only k TTPs can prove together

the link between I and J_j with $\log_I J_j = \log_g Y_j$, since this link is protected by the Diffie-Hellman decisional problem[2];
- assuming that the user's secret key have been stolen; the user asks the TTPs to reveal the set $\{J_j\}_{j \leq \pi}$ corresponding to I; TTPs add them to the coin blacklist.

4.4 Impersonation

Theorem 3. *The scheme achieves framing freeness:*

1. *neither the bank nor TTP can falsely prove that a user performed a transaction,*
2. *neither the bank nor TTP can spend a coin withdrawn by a user.*

Proof. Again, consider the two possible attacks:

1. Assuming that the bank wants to prove that a user overspent coin C; the bank has to deliver to TTPs the corresponding set of transactions τ_i and the signatures corresponding to (J_j, Y_j) public key which is equivalent to knowing the secret key s since user's signature scheme is existentially unforgeable. Assuming now that TTPs want to hide the identity I of a malevolent user and reveal I'; TTPs must prove that: $\log_{I'} J_j = \log_g Y_j = X_j = \log_I J_j$. Therefore, $J_j = I^{X_j} = I'^{X_j}$ which is equivalent to $I' = I$, implying that TTPs must send $I' = I$.
2. User's signature implies that spending a coin required to know user's secret key s due to the existentially unforgeability property; therefore neither the bank nor TTP can spend users' coins.

4.5 Usage of keys

A careful analysis of the scheme leads to the following observation: secret keys s and X are used for El Gamal encryption and Schnorr signature. This feature, introduced for the sake of efficiency (see next section), could open the door to some attacks in case information related to the keys leaks during protocols exchange.

Assuming that signing with a key k reveals enough information for breaking El Gamal: it could be possible to break the related Diffie-Hellman problem with the same key k. Since this problem has been proven to be equivalent in almost all instances to the discrete logarithm problem [16], it means that one could eventually break Schnorr protocol using these information. Thus, the usage of k in our setting does not give any extra advantage over a direct attack on the signature scheme (which means breaking discrete logarithm).

[2] given Y_j, $g = Y_j^{1/X_j}$ and $J_j = Y_j^s$, for any T, it is computationally impossible to decide whether $T \stackrel{?}{=} Y_j^{s/X_j} = g^s = I \bmod p$

5 Efficiency

The scheme presented is generic in the sense that implementations could rely on different cryptographic primitives. Nevertheless, choosing DLP-based primitives is well-suited to our construction since discrete logarithm provides several provably secure schemes [23, 24, 20, 21]. Furthermore, Ps security and efficiency rely on the exponentiation properties (in the sense of Diffie-Hellman's key-exchange protocol [9]). These properties are therefore closely related to our scheme's overall performance. Finally, the scheme is computationally efficient and offers protection (conditional privacy for users, revocable privacy for the bank) to all participants.

Computations: From the user's standpoint, the maximal number of cumbersome computations, *i.e.* exponentiations in a finite field, is four at registration and six at withdrawal time whereas the user has to perform only one exponentiation to spend a coin. Now, we must consider that the registration at the TTPs will be performed once for all, *i.e.* once the user stored the $\{Ps_j\}$ list, he will only withdraw coins from time to time. The extra cost of three exponentiations for obtaining a $\{Ps_j\}$ set is therefore merely marginal. Furthermore, a user may decide to store coins received from other users in order to transfer them to the bank and obtain a coin which value is equivalent to the total value of collected coins. This may result in reducing computations for subsequent payments since the user may exchange a lot of coins at the cost of about one exponentiation only.

The average time for computing an exponentiation on a station or Pentium-like PC is around 40 ms, mainly depending on the exponent size and techniques used for reduction; this is roughly equivalent to performance obtained with a portable device (e.g a smart card) with a cryptographic accelerator. Such timings clearly clamp the total time for a transaction under 1 s, even assuming a low-rate communication link between users and TTPs.

Communications: An important property of this scheme is to allow the trustees to be off-line during payment and withdrawal. A structure where communication between users and trustees is minimized increases overall performance (e.g Main Trustee Structure presented in section 2). The transfer protocol may also reduce global communication in the system by enabling users to avoid several payment interactions (for spending many coins).

Memory Requirements: Considering usual size of parameters, Ps_j is 104-byte long since this is a Schnorr's public key J_j (64 bytes) with its certificate (40 bytes). A coin requires only 64 bytes to be stored with:

Public Information: Ps reference j (1 byte),
date (1 byte) and amount (2 bytes);
Coin Signature: digest (20 bytes) and blind signature (40 bytes).

Observe that this is significantly less than most previously proposed schemes [1, 10, 13] even considering that coins grow in size after transfer operations. Actually,

a coin corresponding to n coins is $(64+8n)$-byte long since the list of coin-related randoms is appended to the new coin. Nevertheless, since the user transfered n coins, he saves $n \times (64 - 8)$ bytes of storage.

The other advantage granted by pseudonym usage is that the cost for Ps storage is divided by the total number of related coins. Since payments performed with a Ps are linkable, this amortization of the memory requirement is clearly associated to the privacy level a user wants to achieve.

Overall Performance: The scheme efficiency thereby compares favorably with recently proposed schemes [3, 11] at a double cost:

- scalable restriction of payment anonymity at the exact appreciation of users,
- presence of Trustees at the account opening, which is not a serious drawback considering that registration is performed only once.

6 Conclusion

We exhibited an efficient electronic cash scheme providing a high level of performance and security. The structure of the public-key architecture combined with Diffie-Hellman's paradigm leads to an efficient construction, resistant to various attacks [13]. The scheme offers also coin semi-transferability and refreshment in order to achieve an user-friendly electronic money system. Finally, the distribution of trustees through a communication network allows several implementation choices, in order to precisely balance security and efficiency.

References

1. S. A. Brands. Untraceable Off-line Cash in Wallets with Observers. In *Crypto '93*, LNCS 773, pages 302–318. Springer-Verlag, 1994.
2. E. Brickell, P. Gemmell, and D. Kravitz. Trustee-based Tracing Extensions to Anonymous Cash and Making of Anonymous Change. In *SODA '95*, pages 457–466, 1995.
3. J. Camenisch, U. Maurer, and M. Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trustees. In *ESORICS '96*, LNCS 1146. Springer-Verlag, 1996.
4. J. Camenisch, J.-M. Piveteau, and M. Stadler. Fair Blind Signatures. In *Eurocrypt '95*, LNCS 921, pages 209–219. Springer-Verlag, 1995.
5. J. Camenisch, J.-M. Piveteau, and M. Stadler. An Efficient Fair Payment System. In *Proc. of the 3rd CCCS*, pages 88–94. ACM press, 1996.
6. D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.
7. D. Chaum. Blind Signatures for Untraceable Payments. In *Crypto '82*, pages 199–203. Plenum, NY, 1983.
8. D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Crypto '88*, LNCS 403, pages 319–327. Springer-Verlag, 1989.
9. W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT–22, no. 6, pages 644–654, november 1976.

10. N. Ferguson. Extensions of Single Term Coins. In *Crypto '93*, LNCS 773, pages 292–301. Springer-Verlag, 1994.
11. Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect Disclosure Proof": Achieving Efficient Fair Off-Line E-Cash. In *Asiacrypt '96*, LNCS 1163, pages 286–300. Springer-Verlag, 1996.
12. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory*, volume IT–31, no. 4, pages 469–472, july 1985.
13. M. Jakobsson and M. Yung. Revokable and Versatile Electronic Money. In *Proc. of the 3rd CCCS*, pages 76–87. ACM press, 1996.
14. M. Jakobsson and M. Yung. Distributed "Magic Ink" Signatures. In *Eurocrypt '97*, LNCS 1233, pages 450–464. Springer-Verlag, 1997.
15. M. Jakobsson and M. Yung. Applying Anti-Trust Policies to Increase Trust in a Versatile e-money System. In *Financial Cryptography '97*, LNCS 1318. Springer-Verlag, 1998.
16. U. M. Maurer. Diffie Hellman Oracles. In *Crypto '96*, LNCS 1109, pages 268–282. Springer-Verlag, 1996.
17. D. M'Raïhi. Cost-Effective Payment Schemes with Privacy Regulation. In *Asiacrypt '96*, LNCS 1163, pages 266–275. Springer-Verlag, 1996.
18. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Crypto '92*, LNCS 740, pages 31–53. Springer-Verlag, 1992.
19. H. Petersen and G. Poupard. Efficient Scalable Fair Cash with Off-line Extortion Prevention. In *Proc. of ICICS'97*, LNCS 1334, pages 463–477. Springer-Verlag, 1997.
20. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, 1996.
21. D. Pointcheval and J. Stern. Provably Secure Blind Signature Schemes. In *Asiacrypt '96*, LNCS 1163, pages 252–265. Springer-Verlag, 1996.
22. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, february 1978.
23. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, pages 235–251. Springer-Verlag, 1990.
24. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
25. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, pages 256–266. Springer-Verlag, 1997.
26. S. von Solms and D. Naccache. On Blind Signatures and Perfect Crimes. *Computers & Security*, 11:581–583, 1992.

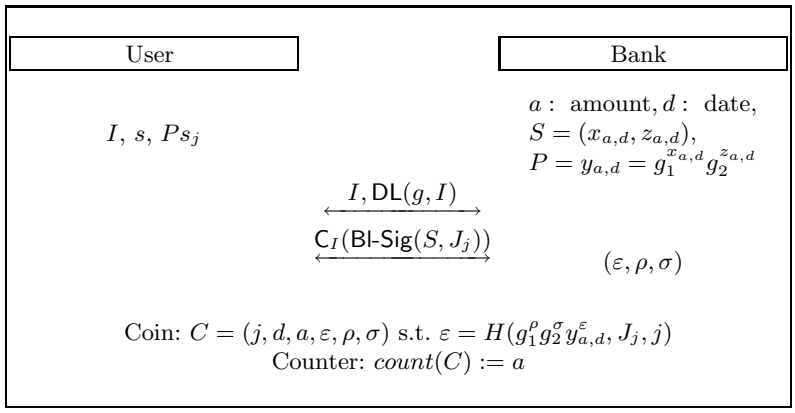


Fig. 2. Withdrawal

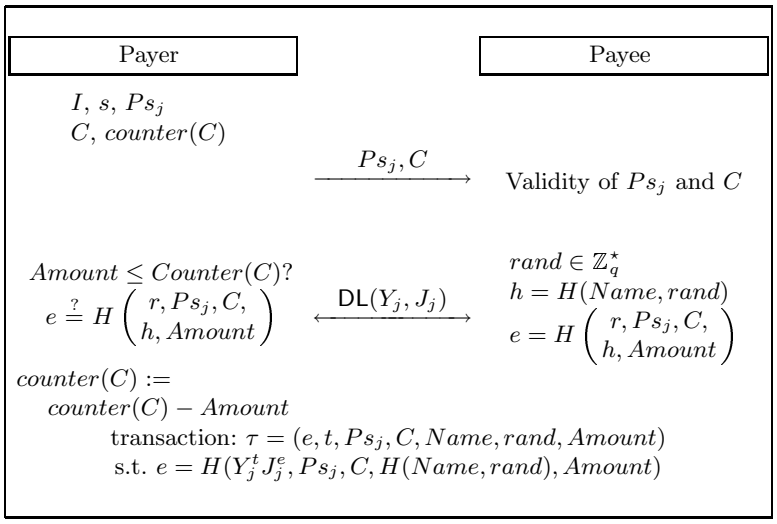


Fig. 3. Payment

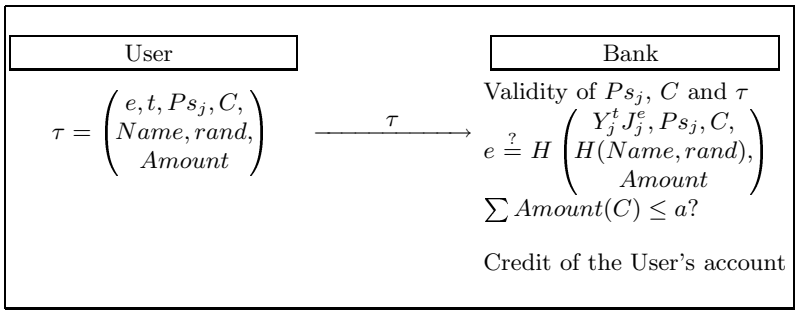


Fig. 4. Deposit ($Name = I$)

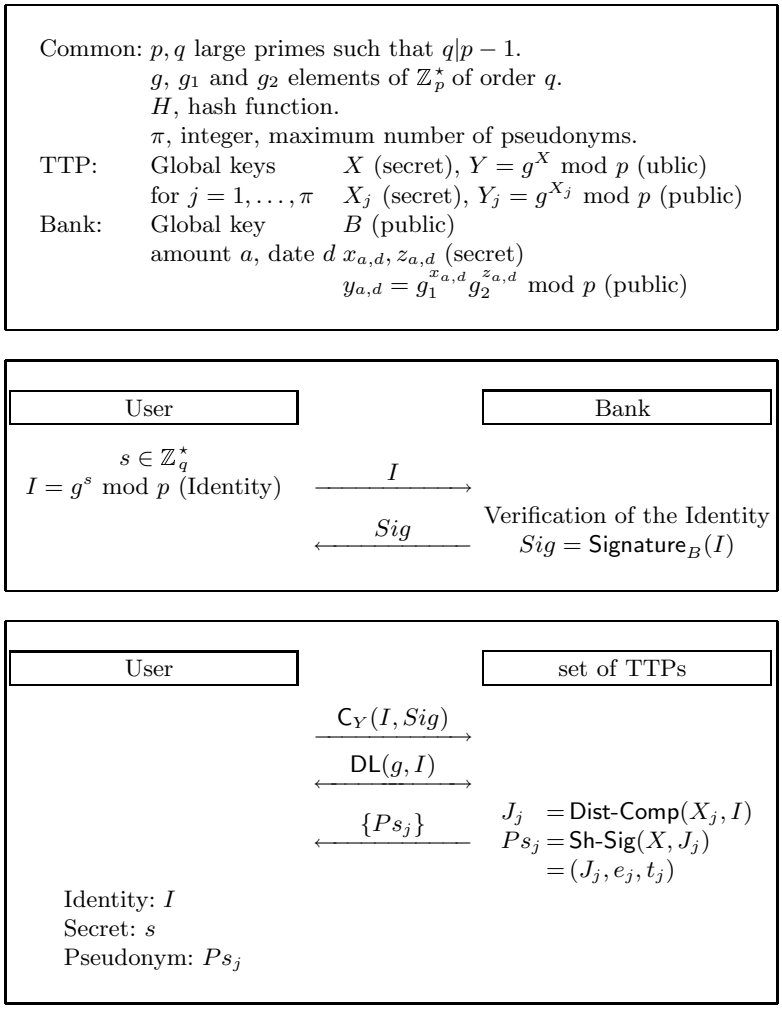


Fig. 5. Opening account