Transferable Constant-Size Fair E-Cash

Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud

École normale supérieure, LIENS - CNRS - INRIA, Paris, France
http://www.di.ens.fr/{~fuchsbau,~pointche,~vergnaud}

Abstract. We propose a new blind certification protocol that provides interesting properties while remaining efficient. It falls in the Groth-Sahai framework for witness-indistinguishable proofs, thus extended to a certified signature it immediately yields non-frameable group signatures. We then use it to build an efficient (offline) e-cash system that guarantees user anonymity and transferability of coins without increasing their size. As required for fair e-cash, in case of fraud, anonymity can be revoked by an authority, which is also crucial to deter from double spending.

1 Introduction

1.1 Motivation

The issue of anonymity in electronic transactions was introduced for e-cash and e-mail in the early 1980's by Chaum, with the famous primitive of blind signatures [Cha83, Cha84]: a signer accepts to sign a message, without knowing the message itself, and without being able to later link a message-signature pair to the transaction it originated from. In e-cash systems, the message is a serial number to make a coin unique. The main security property is resistance to "one-more forgeries" [PS00], which guarantees the signer that after t transactions a user cannot have more than t valid signatures.

Blind signatures have thereafter been widely used for many variants of e-cash systems; in particular *fair* blind signatures [SPC95], which allow to provide revocable anonymity. They deter from abuse since in such a case the signer can ask an authority to reveal the identity of the defrauder. In order to allow the signer to control some part of the message to be signed, *partially* blind signatures [AO00] have been proposed.

Another primitive providing anonymity are group signatures [Cv91], enabling a user to sign as a member of a group without leaking any more information about his identity. The strong security model in [BSZ05] considers dynamic groups in which the group manager is not fully trusted: one thus requires that the latter cannot frame honest users.

For e-cash systems, the classical scenario is between a bank, a user and a merchant/shop: the user withdraws money from the bank and can then spend it in a shop. The latter deposits it at the bank to get its account credited. Literature tries to improve the withdrawal and the spending processes, e.g. with divisible e-cash [EO94, CG07]. However, for many applications, such as e-tickets

J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009, LNCS 5888, pp. 226–247, 2009.

or coupons [NHS99], transferability [OO90, OO92, CG08] is a more desirable property. It is known that the size of coins grows linearly in the number of transfers [CP92]—a drawback we will avoid in our construction by modifying the model (cf. Sect. 1.3).

Classical e-cash requires that as long as a user does not spend a coin twice (double spending), she remains anonymous. Von Solms and Naccache [vSN92] pointed out that perfect anonymity enables perfect crimes, and thus suggested *fair* e-cash, where an authority can trace coins that were acquired illegally. Necessity to fight money laundering also encourages the design of fair e-cash systems enabling a trusted party to revoke the anonymity of users, whenever needed.

1.2 Contributions

Our first result is the definition and efficient pairing-based instantiation of a new primitive, which we call *partially-blind certification*. A protocol allows an issuer to interactively issue a *certificate* to a user, of which parts are then only known to the user and cannot be associated to a particular protocol execution by the issuer. The certificates are unforgeable in that from q runs of the protocol with the issuer cannot be derived more than q valid certificates. We then give two applications of the primitive:

- In order to achieve anonymity and unlinkability in group signatures, a common approach is the following: Using a signing key provided by the group manager, a user produces a signature, encrypts it and adds a proof of its validity. For this method to work efficiently in the standard model, these signing keys have to be constructed carefully. In [BW07] for example, it is the group manager that constructs the entire signing key—which means that he can impersonate (*frame*) users.
  Groth [Gro07] achieves *non-frameability* by using *certified signatures* (defined in [BFPW07]): The user chooses a verification key which is signed by the issuer. A signature produced with the corresponding signing key together with the verification key and the issuer's signature on it can then be verified under the issuer's key. Security of Groth's instantiation however relies on an unnatural assumption.
  We avoid this by observing the following: it is not necessary that the user choose the verification key, as long as she can be sure that the private key contains enough entropy. Since the blind component of our instantiation of our primitive can serve as signing key, our construction applies immediately to build non-frameable group signatures (see Sect. 4).
- Second, in e-cash, the serial number of a coin needs to contain enough entropy to avoid collisions, but again the user need not control it entirely. Partially-blind certificates are applicable here too.

1.3 Transferable Anonymous Constant-Size Fair E-Cash

The instantiation we give of our new primitive allows it to be combined with the results of Groth and Sahai [GS08], which is crucial to our main contribution:

an efficient standard-model anonymous fair e-cash system in the classical three-party scenario with the following novel features:

First, coins are transferable while remaining constant in size. We circumvent the impossibility results by introducing a new method to trace double spenders: the users keep *receipts* when receiving coins instead of storing all information about transfers inside the coin. The amount of data a user has to deal with is thus proportional to the number of coins he received, rather than the path a coin took until reaching him.

Second, partial blindness of our certificates provides the strongest possible notion of anonymity: a user remains anonymous even w.r.t. an entity issuing coins and able to *detect* double spendings.[1] Moreover, coins are unlinkable to anyone except the authority and the double-spending detector. We give an overview of our model before getting back to its security properties.

- The participants of the system are the following: the system manager (that registers users within the system), the bank (issuing coins), users (that withdraw, transfer or spend coins), merchants to which coins are spent, the double-spending detector, and a trusted authority, called tracer, that can trace coins, revoke anonymity and identify double-spenders.
- In order to get a coin, a user runs a withdrawal protocol with the bank, after which he holds a coin and a receipt to be kept even after transferring or spending the coin (to defend himself against wrongful accusation of double-spending).[2]
- Another protocol enables users to transfer coins to other users who, besides the coin, also get a receipt, which they keep too.
- To spend the coin, the user interacts with a merchant. The latter will deposit the coin at the bank who invokes the double-spending detector to check if it has already been spent. If it is the case, the tracer is invoked to reveal the double spender. He does so by tracing back the two instances of the coin by asking the receipts from the users that transferred the coins until identifying the double spender.

Note that the tracing authority identifies *innocent* users that merely transferred a coin that has been used fraudulently before. However, this does not weaken anonymity, which does not hold against the tracer anyway and since identities are not revealed to anyone else. Moreover, this can be proved to be unavoidable in order to achieve constant-size transferable coins. An inevitable shortcoming of our model is that a user who *loses* a receipt can be accused of double spending, since he cannot prove legal acquisition of the coin if he transferred it. The system satisfies the following security notions:

[1] In fair e-cash, there exists an authority that can trace users (user-tracing) and coins (coin-tracing) under a judge decision, in case of fraud suspicion (not necessarily double spending). We separate the notions of *detection* of double spendings, which is done on a regular basis when a coin is deposited, from that of *tracing*, which is performed by a trusted authority only when a fraud was committed.

[2] If one assumes a validity period for coins (after which the issuing key is changed), it suffices to keep a receipt only as long as the respective coin is valid.

- Any user who spends a coin twice is detected.
- As long as a user keeps all his receipts, he cannot be wrongfully accused of double spending, even if everyone else colludes against him.
- A user is anonymous even against collusions of the manager, the bank, the double-spending detector, merchants, and other users.
- Transfers of coins are *unlinkably* anonymous to collusions possibly comprising the manager, the bank, merchants, and other users. (The double-spending detector must necessarily be able to link two spendings of the same coin.)

Our construction is secure in the standard security model (i.e., without relying on the random oracle idealization [BR93])[3] and its security is based on a new (though natural) assumption that holds in the generic group model [Sho97].

1.4 Organization of the Paper

In the next section, we state the employed assumptions. In Sect. 3, we describe our new *Partially-Blind Certification* primitive, and apply it to group signatures in Sect. 4. In Sect. 5, we extend some techniques of Groth-Sahai, recapitulating re-randomization of commitments and introducing proofs for relations of values committed under *different* keys. In Sect. 6, we combine everything to construct our e-cash system.

2 Assumptions

We present the assumptions on bilinear groups on which our security results build. A *bilinear group* is a tuple $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ where $(\mathbb{G}, +)$ and $(\mathbb{G}_T, \cdot)$ are two cyclic groups of order p, G is a generator of $\mathbb{G}$, and $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map, i.e., $\forall U, V \in \mathbb{G}\ \forall a, b \in \mathbb{Z}\colon\ e(aU, bV) = e(U, V)^{ab}$, and $e(G, G)$ is a generator of $\mathbb{G}_T$.

The first two of the following assumptions are classical [DH76, BBS04]. The third is a simple extension of the Hidden Strong Diffie-Hellman Problem proposed by Boyen and Waters in [BW07].

Definition 1. *The* Computational Diffie-Hellman (CDH) Assumption *states that the following problem is intractable*[4]*: given* $(G, \alpha G, \beta G) \in \mathbb{G}^3$*, for* $\alpha, \beta \in \mathbb{Z}_p$*, output* $\alpha\beta G$*.*

[3] Note that in our context, due to re-randomization of encryptions (cf. Sect. 6.2 for details), it seems even impossible to replace the Groth-Sahai techniques with the Fiat-Shamir heuristic [FS87] to improve efficiency at the expense of relying on the random oracle model.

[4] We say that a computational problem is *intractable* if no probabilistic polynomial-time (p.p.t.) adversary can solve it with non-negligible probability. A decisional problem is *intractable* if no p.p.t. adversary can decide it with probability of non-negligibly more than $1/2$.

Definition 2. *The* Decisional Linear (DLIN) Assumption *states that the following problem is intractable: given* $(U, V, G, \alpha U, \beta V, \gamma G) \in \mathbb{G}^6$*, decide whether* $\gamma = \alpha + \beta$ *or not.*

Definition 3. *The* q-Double Hidden Strong Diffie-Hellman (DHSDH) Assumption *states that the following problem is intractable: given* $(G, H, K, \Gamma = \gamma G) \in \mathbb{G}^4$ *and* $q-1$ *tuples*

$$\left(X_i = x_i G,\, X'_i = x_i H,\, Y_i = y_i G,\, Y'_i = y_i H,\, A_i = \tfrac{1}{\gamma + x_i}(K + y_i G)\right)$$

with $x_i, y_i \leftarrow \mathbb{Z}_p^*$ $(1 \le i \le q-1)$*, output a new tuple* $\left(X = xG,\, X' = xH,\, Y = yG,\, Y' = yH,\, A = \tfrac{1}{\gamma + x}(K + yG)\right)$.

Note that a tuple (X, X', Y, Y', A) has the above format if and only if it satisfies

$$e(X, H) = e(G, X') \qquad e(Y, H) = e(G, Y') \qquad e(A, \Gamma + X) = e(K + Y, G)$$

Remark 4. Boneh and Boyen [BB04] introduced the Strong Diffie-Hellman (SDH) assumption stating that given a $(q+1)$-tuple $(G, \gamma G, \gamma^2 G, \ldots, \gamma^q G) \in \mathbb{G}^{q+1}$ for a random $\gamma \leftarrow \mathbb{Z}_p^*$, it is infeasible to output a pair $(x, \frac{1}{\gamma + x} G) \in \mathbb{Z}_p \times \mathbb{G}$. Hardness of SDH implies hardness of the following two problems (the first implication is proven in [BB04], the second in the full version [FPV09]):

(I) Given $G, \gamma G \in \mathbb{G}$ and $q-1$ distinct pairs $(x_i, \frac{1}{\gamma + x_i} G) \in \mathbb{Z}_p \times \mathbb{G}$, output a *new* pair $(x, \frac{1}{\gamma + x} G) \in \mathbb{Z}_p \times \mathbb{G}$.

(II) Given $G, K, \gamma G \in \mathbb{G}$ and $q-1$ distinct triples $\left(x_i, y_i, \frac{1}{\gamma + x_i}(K + y_i G)\right) \in \mathbb{Z}_p^2 \times \mathbb{G}$, output a *new* triple $\left(x, y, \frac{1}{\gamma + x}(K + yG)\right) \in \mathbb{Z}_p^2 \times \mathbb{G}$.

The Hidden SDH problem defined in [BW07] is a variant of Problem (I), where instead of giving the x_i's explicitly, they are given as $(x_i G, x_i H)$. Similarly, the goal is to output a new triple $(xG, xH, \frac{1}{\gamma + x} G)$. Now the Double Hidden SDH assumption (Definition 3) transforms Problem (II) the same way: instead of being given explicitly, x_i and y_i are given as $(x_i G, x_i H, y_i G, y_i H)$. In the full version [FPV09] we discuss assumptions derived from SDH and their relations.

3 Partially-Blind Certification

3.1 Model

Definition 5. *A* partially-blind certification scheme $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{User}, \mathsf{Verif})$ *is a 4-tuple of (interactive) probabilistic polynomial-time Turing machines (PPTs) such that:*

– Setup *is a PPT that takes as input an integer* k *and outputs a pair* (pk, sk) *of* public *(resp.* secret*)* key*. We call* k *the* security parameter*.*

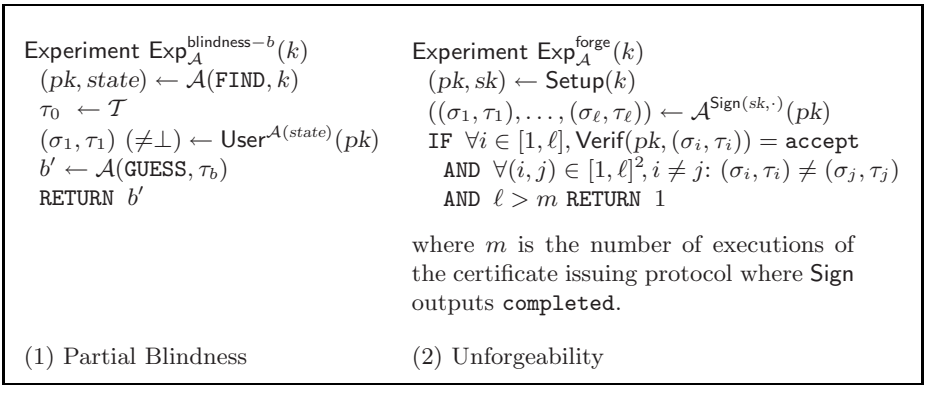

Fig. 1. Security experiments for partially-blind certificates

- Sign *and* User *are interactive PPTs such that* User *takes as inputs a public key* pk *and* Sign *takes as input the matching secret key* sk. Sign *and* User *engage in the* certificate-issuing protocol *and when they stop,* Sign *outputs* `completed` *or* `not-completed` *while* User *outputs a pair of bit strings* (σ, τ) *or* $\perp$.
- Verif *is a deterministic polynomial-time Turing machine that on input a public key* pk *and a pair of bit strings* (σ, τ) *outputs either* `accept` *or* `reject`.

For all $k \in \mathbb{N}$, *all pairs* (pk, sk) *output by* $\mathsf{Setup}(k)$, *if* Sign *and* User *follow the certificate issuing protocol with input* sk *and* pk *respectively, then* Sign *outputs* `completed` *and* User *outputs a pair* (σ, τ) *that satisfies* $\mathsf{Verif}(pk, (\sigma, \tau)) = \mathtt{accept}$. *A pair* (σ, τ) *is termed* valid with regard to pk *if on input* $(pk, (\sigma, \tau))$ Verif *outputs* `accept`, *in which case, we say that* (σ, τ) *is a* certificate *for* pk *and* τ *is termed the* blind component *of the certificate. We denote* $\mathcal{T} \subset \{0, 1\}^*$ *the set of bit-strings which are blind component of some certificate.*

Partial Blindness. To define partial blindness, we consider the *real-or-random* game (i.e., random experiment) among an adversarial signer $\mathcal{A}$ and a challenger presented in Fig. 1 (1).

- We define the advantage of $\mathcal{A}$ in breaking partial blindness by its advantage in distinguishing the two above experiments (with $b = 0$ or $b = 1$):

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{blindness}}(k) \;:=\; \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{blindness}-1}(k) = 1] \;-\; \Pr[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{blindness}-0}(k) = 1] \;,$$

  where the probability is taken over the coin tosses made by the challenger and $\mathcal{A}$.
- The scheme $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{User}, \mathsf{Verif})$ is said to be *partially blind* if no adversary $\mathcal{A}$ running in probabilistic polynomial time has a non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{blindness}}$.

Unforgeability. To define unforgeability, we introduce the game among an adversarial user $\mathcal{A}$ and an honest signer Sign depicted in Fig. 1 (2).

(1) User Choose $r, y_1 \leftarrow \mathbb{Z}_p$, compute and send: $R_1 := r(K + y_1 G), \quad T := rG$ and zero-knowledge proofs of knowledge of r and y_1 satisfying the relations (cf. Remark 7).

(2) Sign Choose $x, y_2 \leftarrow \mathbb{Z}_p$ and compute $R := R_1 + y_2 T$ (note that $R = r(K + yG)$ with $y := y_1 + y_2$.)

Send $\left(S_1 := \frac{1}{\omega + x} R,\ S_2 := xG,\ S_3 := xH,\ S_4 := y_2 G,\ S_5 := y_2 H\right)$

(3) User Check whether $(S_1, S_2, S_3, S_4, S_5)$ is correctly formed:

$$e(S_2, H) \stackrel{?}{=} e(G, S_3) \qquad e(S_4, H) \stackrel{?}{=} e(G, S_5) \qquad e(S_1, \Omega + S_2) \stackrel{?}{=} e(R, G)$$

If so, compute a certificate

$$\left(A := \tfrac{1}{r} S_1,\ X := S_2,\ X' := S_3,\ Y := y_1 G + S_4 = yG,\ Y' := y_1 H + S_5 = yH\right)$$

Fig. 2. Partially-blind certificate-issuing protocol

– We define the *success* of $\mathcal{A}$ in this game by

$$\mathbf{Succ}^{\mathsf{unforge}}_{\mathcal{A}}(k) \ := \ \Pr[\mathsf{Exp}^{\mathsf{forge}}_{\mathcal{A}}(k) = 1] \ ,$$

where the probability is taken over the coin tosses made by $\mathcal{A}$, Setup and Sign.

– The scheme (Setup, Sign, User, Verif) is said to be *unforgeable* if no adversary $\mathcal{A}$ running in probabilistic polynomial time has a non-negligible success $\mathbf{Succ}^{\mathsf{unforge}}_{\mathcal{A}}$.

Remark 6. In the experiment $\mathsf{Exp}^{\mathsf{forge}}_{\mathcal{A}}$, depending on the security model, the executions of the certificate issuing protocol are run sequentially or in a concurrent and interleaving way.

3.2 Instantiation

Let $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ be a bilinear group and $G, H, K \in \mathbb{G}$ be public parameters; define the signer's key pair as $sk := \omega \leftarrow \mathbb{Z}_p$ and $pk = \Omega := \omega G$. A certificate is defined as

$$\mathsf{Crt}(\omega\,;\, x, y) \ := \ \begin{cases} A = \dfrac{1}{\omega + x}(K + yG) & X \ = xG \quad Y \ = yG \\ & X' = xH \quad Y' = yH \end{cases}$$

for $x, y \leftarrow \mathbb{Z}_p$, with $\sigma := (A, X, X', Y)$ and the blind component $\tau := Y' \in \mathbb{G}$. It satisfies:

$$e(X, H) = e(G, X') \quad e(Y, H) = e(G, Y') \quad e(A, \Omega + X) = e(K + Y, G) \qquad (1)$$

Fig. 2 depicts an efficient protocol to interactively generate such a certificate between the signer (issuer) that controls x and the user that partially controls

y: at the end, the signer has no information about y, except that it is uniformly distributed.

Remark 7. In the first round of the User protocol, one can use interactive Schnorr-like zero-knowledge proofs of knowledge (ZKPoK) [Sch90]. Extraction is then only possible for constant-depth concurrency [Oka06]. To achieve *full* concurrency, and at the same time reduce interactivity to 2 moves, one can use the following technique: Make linear commitments [GOS06] (cf. Sect. 5.1) to the bits of r and y_1 (which are extractable) and use the proof techniques from [FP09, Appendix A.3 of the full version]. The drawbacks of this approach are that security holds in the common reference string (CRS) model and we incur a loss of efficiency.

3.3 Security Results

Theorem 8. *Under DHSDH, the above certificates are unforgeable.*

Proof. Let $\mathcal{A}$ be an adversary impersonating corrupt users running the issuing protocol up to $q-1$ times and then outputting q different valid certificates. We build $\mathcal{B}$ solving q-DHSDH with the same probability by simulating the signer: $\mathcal{B}$ gets a q-DHSDH-instance $\big(G, H, K, \Omega, (A_i, X_i, X'_i, Y_i, Y'_i)_{i=1}^{q-1}\big)$. If the ZKPoK are non-interactive, it sets the CRS so that it can extract r and y_1 used in R_1 and T—if they are interactive, $\mathcal{B}$ rewinds $\mathcal{A}$ to extract. In each issuing, $\mathcal{A}$ first sends $(R_{1,i}, T_i)$ and proofs of knowledge. If the proofs are correct, $\mathcal{B}$ extracts r_i, $y_{1,i}$ from them and sends $\big(S_{1,i} := r_i A_i$, $S_{2,i} := X_i$, $S_{3,i} := X'_i$, $S_{4,i} := Y_i - y_{1,i}G$, $S_{5,i} := Y'_i - y_{2,i}H\big)$. Finally, $\mathcal{B}$ checks the q certificates and forwards one different from those in the DHSDH-instance to its own challenger. □

Theorem 9. *Under DLIN, the above certificates are partially blind.*

Proof. Consider $\mathcal{A}$, which after an execution of the blind issuing protocol can decide whether the blind component $\tau = Y'$ is real or random in $\mathbb{G}$. We build $\mathcal{B}$ deciding DLIN with the success probability of $\mathcal{A}$. The algorithm $\mathcal{B}$ gets a DLIN-instance (H, G, T, Z, K, R_1), i.e., it has to decide whether

$$R_1 \stackrel{?}{=} (\log_H Z + \log_G K)\, T \tag{2}$$

It gives $\mathcal{A}$ the triple (G, H, K) as public parameters (and a simulating CRS in case we use non-interactive ZKPoK) and gets Ω, the issuer's public key from $\mathcal{A}$. $\mathcal{B}$ runs the protocol User with $\mathcal{A}$, starting by sending R_1, and T from its DLIN instance and simulating the PoK.

After getting back $(S_1, \ldots, S_5)$, $\mathcal{B}$ checks its correctness and gives $\mathcal{A}$ the following: $Y' := Z + S_5$, with Z from its DLIN instance. ($\mathcal{B}$ can verify correctness of S without knowledge of y_1 and r by checking $e(S_2, H) = e(G, S_3)$, $(S_4, H) = e(G, S_5)$ and $e(S_1, \Omega + S_2) = e(R, G)$. Also note that $\mathcal{B}$ only needs to produce the last (blind) component of the certificate.) Finally $\mathcal{A}$ outputs a guess b', which $\mathcal{B}$ forwards to its DLIN challenger.

– If the DLIN instance is not a linear tuple then Z and therefore Y' is independently random.
– If (H, G, T, Z, K, R_1) is linear, then with $y_1 := \log_H Z$, $\kappa := \log_G K$, and $r := \log_G T$, we have $R_1 = (y_1 + \kappa)T$ by (2). Furthermore, for public parameters (G, H, K), we have

$$T = rG \qquad R_1 = (y_1 + \kappa)T = (y_1 + \kappa)rG = r(K + y_1 G) \qquad Z = y_1 H$$

which means that $Y' = Z + S_5$ is the blind component of a correctly produced certificate.

If $\mathcal{B}$ outputs the bit returned by $\mathcal{A}$, its success probability is equal to $\mathbf{Adv}^{\mathsf{blindness}}_{\mathcal{A}}$. □

4 A Fully-Secure Group Signature from Partially-Blind Certificates

As a first application of the certification protocol from Sect. 3.2, we construct *fully-secure* dynamic group signatures (in the sense of [BSZ05], in particular satisfying non-frameability and CCA-anonymity) without random oracles. We construct a *certified-signature* scheme, to which can then be applied Groth's [Gro07] methodology of transforming certified signatures that respect a certain structure into group signatures using Groth-Sahai NIZK proofs [GS08] and Kiltz' tag-based encryption [Kil06], both relying exclusively on the DLIN assumption.

The resulting scheme is less efficient than that from [Gro07]; however, it is based on a more natural assumption, while at the same time being of the same order of magnitude—especially compared to the first instantiations of fully-secure signatures in the standard model (e.g., [Gro06]). We think of the scheme as somehow being the "natural" extension of [BW07] in order to satisfy non-frameability.

Certified Signatures. A certified-signature scheme consists of a setup algorithm, a key-generation algorithm for the certification authority, an interactive protocol between the authority ("issuer") and a user letting the latter obtain a triple (*cert*, *vk*, *sk*), where *vk* is a verification key for a signature scheme, *sk* is the corresponding signing key (unknown to the issuer) and *cert* is a certificate on *vk*.

Besides correctness, Groth [Gro07] gives two security criteria that a certified signature must satisfy to be transformable into a secure group signature scheme: *Unfakeability* requires that no user can create a certificate for and a signature under a verification key that was not certified by the issuer. *Unforgeability* means that even a corrupt authority issuing a tuple (*cert*, *vk*, *sk*) cannot forge a signature under *vk*.

Our Instantiation. Our certified signature is constructed from a certificate (A, X, X', Y, Y') by using (Y, Y') as a pair of public and secret key for Waters' signature scheme [Wat05]. A certified signature consists thus of the first four components of the certificate prepended to a Waters signature. Note that what is called *cert* above corresponds to (A, X, X') here, and (vk, sk) would be

Let $(U_i)_{i=0}^n \in \mathbb{G}^{n+1}$ be part of the public parameters; let Ω be the issuer's public key.

Certificate Generation. Run the certificate-creation protocol in Fig. 2, except that the issuer running Sign sends an extractable commitment of $S_4 = y_2 G$ before phase (1) and opens it in phase (2).

Signing. For a message $M = (m_1, \ldots, m_n) \in \{0,1\}^n$, denote $\mathcal{F}(M) := U_0 + \sum_{i=1}^n m_i U_i$. Given a certificate $C = (A, X, X', Y, Y')$, a signature on M using randomness $s \in \mathbb{Z}_p$ is defined as

$$\mathsf{Sig}(C, M\,;\, s) := (A, X, X', Y, Y' + s\mathcal{F}(M), -sG) \ .$$

Verification. A certified signature (A, X, X', Y, Z, Z') on message M is verified by checking

$$e(X,H) = e(G,X') \quad e(Y,H) = e(G,Z)\, e(Z', \mathcal{F}(M)) \quad e(A, \Omega{+}X) = e(K{+}Y, G)$$

Fig. 3. Chosen-message secure certified signature

(Y, Y'). The scheme is given in Fig. 3. Our construction satisfies the security requirements given by Groth:

Theorem 10. *The certified-signature scheme in Fig. 3 is perfectly correct, unfakeable under DHSDH, and existentially unforgeable under chosen-message attack under CDH.*

Proof. Correctness follows by inspection. The two other properties are proven similarly to Theorems 8 and 9, we thus highlight the differences.

(1) Unfakeability means that after running the issuing protocol multiple times with the issuer, no user is able to produce a valid tuple (A, X, X', Y, Z, Z') with Y different from those in the obtained certificates. The proof works similarly to that of Theorem 8 with the following modifications: For $0 \le i \le n$, $\mathcal{B}$ chooses $\mu_i \leftarrow \mathbb{Z}_p$ and sets the public parameters $U_i := \mu_i G$. In the issuing protocol, $\mathcal{B}$ simulates the additional commitment at the beginning. From a valid (A, X, X', Y, Z, Z') returned by $\mathcal{A}$, $\mathcal{B}$ can then extract a new certificate by setting $Y' := Z + (\mu_0 + \sum m_i \mu_i) Z'$.

(2) Existential unforgeability under chosen-message attack (EUF-CMA) follows from partial blindness of certificates and security of Waters signatures, which is implied by CDH (Def. 1): Let $\mathcal{A}$ be an adversary impersonating the issuer and mounting a chosen-message attack. We construct $\mathcal{B}$ against EUF-CMA of Waters signatures. $\mathcal{B}$ is given a Waters public key $V \in \mathbb{G}$ and a signing oracle.

$\mathcal{B}$ runs the certificate-generation protocol playing the role of User with $\mathcal{A}$. When $\mathcal{A}$ sends a commitment to S_4 in the first phase of the protocol, $\mathcal{B}$ extracts S_4 from it. It then chooses r, sends $R_1 := r(K + V - S_4)$ and $T := rG$ and simulates the zero-knowledge proofs. (Note that this implicitly sets $V = (y_1 + y_2)G$.) If $\mathcal{A}$ returns a valid tuple $(S_1, S_2, S_3, S_4, S_5)$, $\mathcal{B}$ can compute an (incomplete) certificate $(A := \frac{1}{r} S_1,\ X := S_2,\ X' := S_3,\ Y := V)$ which suffices to answer $\mathcal{A}$'s

signing queries, as $\mathcal{B}$ can get the last two components by querying its own oracle. When $\mathcal{A}$ returns a successful forgery, $\mathcal{B}$ returns the last two components, i.e., a Waters signature under public key V. □

5 New Techniques for Groth-Sahai Proof Systems

5.1 Preliminaries

We briefly review the results of [GS08] relevant to our paper: witness-indistinguishable (WI) proofs that elements in $\mathbb{G}$ that were committed to via *linear commitments* satisfy *pairing-product equations*. We refer to the original work for more details and proofs.

Let $P \in \mathbb{G}$ be a generator. We define a key for *linear commitments*. Choose $\alpha, \beta, r_1, r_2 \in \mathbb{Z}_p$ and define $U = \alpha P$, $V = \beta P$, and

$$\mathbf{u}_1 := (U, 0, P) \qquad \mathbf{u}_2 := (0, V, P) \qquad \mathbf{u}_3 := (W_1, W_2, W_3) \tag{3}$$

where $W_1 := r_1 U$, $W_2 := r_2 V$, and W_3 is either

- soundness setting: $W_3 := (r_1 + r_2)P$ (which makes $\vec{\mathbf{u}}$ a binding key)
- WI setting: $W_3 := (r_1 + r_2 - 1)P$ (which makes $\vec{\mathbf{u}}$ a hiding key)

Under key $ck = (U, V, W_1, W_2, W_3)$, a commitment to a group element $X \in \mathbb{G}$ using randomness $(s_1, s_2, s_3) \leftarrow \mathbb{Z}_p^3$ is defined as (with $\iota(X) := (0, 0, X)$)

$$\begin{aligned}\mathsf{Com}\big(ck, X; (s_1, s_2, s_3)\big) &:= \iota(X) + \textstyle\sum_{i=1}^{3} s_i \mathbf{u}_i \\ &= (s_1 U + s_3 W_1,\ s_2 V + s_3 W_2,\ X + s_1 P + s_2 P + s_3 W_3)\ .\end{aligned}$$

Note that in the soundness setting, given the *extraction key* $ek := (\alpha, \beta)$, the committed value can be extracted from a commitment $\mathbf{c} = (c_1, c_2, c_3)$:

$$\begin{aligned}&\mathsf{Extr}((\alpha, \beta), \mathbf{c}) := c_3 - \tfrac{1}{\alpha} c_1 - \tfrac{1}{\beta} c_2 \\ &\quad = X + (s_1 + s_2 + s_3(r_1 + r_2))P - \tfrac{1}{\alpha}(s_1 + s_3 r_1)U - \tfrac{1}{\beta}(s_2 + s_3 r_2)V) = X\ ,\end{aligned}$$

since $\frac{1}{\alpha} U = P$ and $\frac{1}{\beta} V = P$. On the other hand, in the WI setting we have (with $s_1' := s_1 + s_3 r_1$ and $s_2' = s_2 + s_3 r_2$): $\mathbf{c} = (s_1' U, s_2' V, X + (s_1' + s_2' - s_3)P)$, which is equally distributed for every X. The two settings are indistinguishable by DLIN since for soundness (W_1, W_2, W_3) is linear w.r.t. (U, V, P), whereas in the WI setting it is not.

For the sake of readability and consistency with the work of [GS08], we stick to their abstract notation, which we sketch briefly:

- For a vector $\vec{\mathcal{X}} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)^\top \in \mathbb{G}^n$, let $\vec{\mathcal{X}} \cdot \vec{\mathcal{Y}} := \prod_{i=1}^{n} e(\mathcal{X}_i, \mathcal{Y}_i)$.
- Bold letters denote triples, e.g., $\mathbf{d} = (d_1, d_2, d_3) \in \mathbb{G}^{1\times 3}$, $\vec{\mathbf{d}}$ denotes a column vector of triples, thus a matrix in $\mathbb{G}^{n\times 3}$. Furthermore, define $\widetilde{F}(\mathbf{c}, \mathbf{d}) := \big[e(c_i, d_j)\big]_{i,j=1,3} \in \mathbb{G}_T^{3\times 3}$. In $\mathbb{G}_T^{3\times 3}$, "+" denotes entry-wise multiplication of matrix elements. Define $\mathbf{c} \bullet \mathbf{d} := \sum_{i=1}^{n} \big(1/2\, \widetilde{F}(c_i, d_i) + 1/2\, \widetilde{F}(d_i, c_i)\big)$.

A *pairing-product equation* is an equation for variables $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in \mathbb{G}$ of the form

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{Y}_i) \prod_{i=1}^{n} \prod_{j=1}^{n} e(\mathcal{Y}_i, \mathcal{Y}_j)^{\gamma_{i,j}} \;=\; t_T \;,$$

with $\mathcal{A}_i \in \mathbb{G}$, $\gamma_{i,j} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$. Setting $\Gamma := \left[\gamma_{i,j}\right]_{i,j=1,\ldots,n} \in \mathbb{Z}_p^{n \times n}$, this can be written as

$$(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})\,(\vec{\mathcal{Y}} \cdot \Gamma \vec{\mathcal{Y}}) \;=\; t_T \;. \tag{4}$$

Set $H_1 := \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, H_2 := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, H_3 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$, and $\iota_T(t_T) := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & t_T \end{bmatrix}$ for $t_T \in \mathbb{G}_T$.

Let $\vec{\mathbf{d}}$ be a vector of commitments to $\vec{\mathcal{Y}}$, i.e., $\vec{\mathbf{d}} := \iota(\vec{\mathcal{Y}}) + S\vec{\mathbf{u}}$ with $S \leftarrow \mathbb{Z}_p^{n \times 3}$ and $\iota(\vec{\mathcal{Y}}) := [\iota(\mathcal{Y}_i)]_{i=1,\ldots,n}$. The proof that the values committed in $\vec{\mathbf{d}}$ satisfy (4) is defined as

$$\Phi \;:=\; S^\top \iota(\vec{\mathcal{A}}) + S^\top \Gamma \iota(\vec{\mathcal{Y}}) + S^\top \Gamma^\top \iota(\vec{\mathcal{Y}}) + S^\top \Gamma S \vec{\mathbf{u}} + \textstyle\sum_{i=1}^{3} r_i H_i \vec{\mathbf{u}} \;, \tag{5}$$

with $r_1, r_2, r_3 \leftarrow \mathbb{Z}_p$, and is verified by

$$\iota(\vec{\mathcal{A}}) \bullet \vec{\mathbf{d}} + \vec{\mathbf{d}} \bullet \Gamma \vec{\mathbf{d}} \;=\; \iota_T(t_T) + \vec{\mathbf{u}} \bullet \Phi \;. \tag{6}$$

Soundness and WI of the proofs. In the soundness setting, if $\vec{\mathbf{d}}$ satisfies (6) for some Φ, then Extr extracts $\vec{\mathcal{Y}}$ satisfying (4). In the WI setting, let $\vec{\mathbf{c}}$ and $\vec{\mathbf{d}}$ be commitments to $\vec{\mathcal{X}}$ and $\vec{\mathcal{Y}}$, resp., which both satisfy (4). Then Φ and Φ' constructed as in (5) for $\vec{\mathbf{c}}$ and $\vec{\mathbf{d}}$, resp., are equally distributed.

5.2 Commitment Re-randomization and Proof Updating

As observed by [FP09] and [BCC+09], commitments of this form can be *re-randomized* and the corresponding proofs adapted without knowledge of the committed values nor the used randomness: Given a commitment $\vec{\mathbf{d}}$, set $\vec{\mathbf{c}} := \vec{\mathbf{d}} + \widetilde{S}\vec{\mathbf{u}}$, for $\widetilde{S} \leftarrow \mathbb{Z}_p^{n \times 3}$, and *update* the proof Φ for $\vec{\mathbf{d}}$ to $\widetilde{\Phi}$ for $\vec{\mathbf{c}}$:

$$\widetilde{\Phi} \;:=\; \Phi + \widetilde{S}^\top \iota(\vec{\mathcal{A}}) + \widetilde{S}^\top \Gamma \vec{\mathbf{d}} + \widetilde{S}^\top \Gamma^\top \vec{\mathbf{d}} + \widetilde{S}^\top \Gamma \widetilde{S} \vec{\mathbf{u}} + \textstyle\sum_{i=1}^{3} \widetilde{r}_i H_i \vec{\mathbf{u}} \tag{7}$$

with $\widetilde{r}_i \leftarrow \mathbb{Z}_p$. The pair $(\vec{\mathbf{c}}, \widetilde{\Phi})$ satisfies (6) and some calculation shows that $\widetilde{\Phi}$ is constructed as in (5) for $\vec{\mathbf{c}}$ being a commitment to $\vec{\mathcal{Y}}$ using randomness $S + \widetilde{S}$. (In particular (7) yields the same $\widetilde{\Phi}$ as (5) if in the latter the randomness used for the proof is $(r_i + \alpha_i + \widetilde{r}_i)_{i=1}^3$, where (r_1, r_2, r_3) is the randomness of Φ and $\alpha_1, \alpha_2, \alpha_3$ are such that $A := \widetilde{S}^\top \Gamma^\top S - S^\top \Gamma \widetilde{S} = \sum_{i=1}^{3} \alpha_i H_i$; such α_i exist since

A satisfies $\vec{\mathbf{u}} \bullet A\vec{\mathbf{u}} = 0$ and the H_i's form a basis for the matrices of this form; cf. [GS08, Chapter 4].)

5.3 Linear Equations and Different Commitment Keys

Consider two commitments $\mathbf{c}, \mathbf{d}$ of Y, Z under *different* commitment keys $\vec{\mathbf{u}}$ and $\vec{\mathbf{u}}'$, respectively. We construct a re-randomizable WI proof that the committed values satisfy

$$e(H, Y) \;=\; e(G, Z) \;. \tag{8}$$

Let $\mathbf{c}$ be a commitment to Y w.r.t. key $\vec{\mathbf{u}}$: $\mathbf{c} := (s_{Y1}U + s_{Y3}W_1,\ s_{Y2}V + s_{Y3}W_2,\ Y + s_{Y1}P + s_{Y2}P + s_{Y3}W_3)$. The proof that the committed value Y satisfies (8) (in which Z is considered as a constant) is[5] $\pi := (s_{Y1}H, s_{Y2}H, s_{Y3}H)$, which is verified by

$$e(\pi_1, U)\, e(\pi_3, W_1) \;=\; e(H, c_1) \tag{9a}$$
$$e(\pi_2, V)\, e(\pi_3, W_2) \;=\; e(H, c_2) \tag{9b}$$
$$e(Z, G)\, e(\pi_1, P)\, e(\pi_2, P)\, e(\pi_3, W_3) \;=\; e(H, c_3) \tag{9c}$$

Regarding (9) as a set of equations over variables $c_1, c_2, c_3, Z, \pi_1, \pi_2, \pi_3$, we could just use the Groth-Sahai proof system a second time by committing to the new variables under key $\vec{\mathbf{u}}'$ and making proofs for the equations in (9). However, this can be optimized, since we need not commit to c_1, c_2 and c_3. Correctness and soundness follow from a simple hybrid argument.

Let us consider witness indistinguishability. We show that every pair (Y, Z) satisfying (8) generates the same distribution of proofs once both keys $\vec{\mathbf{u}}$ and $\vec{\mathbf{u}}'$ are replaced by hiding keys. Let (Y, Z) satisfying (8) be arbitrarily fixed. Since $\mathbf{u}$ is perfectly hiding, for any given $\mathbf{c}$ there exist (s_1, s_2, s_3) s.t. $\mathbf{c} = \iota(Y) + \sum_{i=1}^{3} s_i \mathbf{u}_i$. Now WI under key $\vec{\mathbf{u}}'$ (of the second layer of commitments/proofs) ensures that every (Z, π_1, π_2, π_3) satisfying (9) (with the c_i's fixed!) generates identically distributed proofs. Thus for $Z := (\log_G Y)H$, $\pi_i := s_i H$, the proof does not leak anything. We present the details:

We make commitments to Z, $\pi_1 = s_{Y1}H$, $\pi_2 = s_{Y2}H$, $\pi_3 = s_{Y3}H$ w.r.t. $\vec{\mathbf{u}}'$:

$$\mathbf{d} := \begin{bmatrix} s_{Z1}U' + s_{Z3}W_1' \\ s_{Z2}V' + s_{Z3}W_2' \\ Z + s_{Z1}P' + s_{Z2}P' + s_{Z3}W_3' \end{bmatrix} \quad \mathbf{p}_i := \begin{bmatrix} t_{i,1}U' + t_{i,3}W_1' \\ t_{i,2}V' + t_{i,3}W_2' \\ s_{Yi}H + t_{i,1}P' + t_{i,2}P' + t_{i,3}W_3' \end{bmatrix} \tag{10}$$

for $1 \le i \le 3$. The proof ψ_i for the i-th equation in (9) is defined as follows:

$$\begin{aligned} &\psi_{1,j} := t_{1,j}U + t_{3,j}W_1 \qquad\qquad \psi_{2,j} := t_{2,j}V + t_{3,j}W_2 \\ &\psi_{3,j} := s_{Zj}G + t_{1,j}P + t_{2,j}P + t_{3,j}W_3 \qquad\qquad \text{for } 1 \le j \le 3 \end{aligned} \tag{11}$$

[5] Groth-Sahai proofs for linear equations reduce to 3 group elements; see Sect. 6.1 of the full version of [GS08].

The final verification relations are the following:

$$
\begin{aligned}
\text{For (9a):}\quad & e(p_{1,1},U)\,e(p_{3,1},W_1) &&= e(\psi_{1,1},U')\,e(\psi_{1,3},W_1')\\
& e(p_{1,2},U)\,e(p_{3,2},W_1) &&= e(\psi_{1,2},V')\,e(\psi_{1,3},W_2')\\
& e(p_{1,3},U)\,e(p_{3,3},W_1) &&= e(H,c_1)\,e(\psi_{1,1},P')\,e(\psi_{1,2},P')\,e(\psi_{1,3},W_3')\\
\text{For (9b):}\quad & e(p_{2,1},V)\,e(p_{3,1},W_2) &&= e(\psi_{2,1},U')\,e(\psi_{2,3},W_1')\\
& e(p_{2,2},V)\,e(p_{3,2},W_2) &&= e(\psi_{2,2},V')\,e(\psi_{2,3},W_2')\\
& e(p_{2,3},V)\,e(p_{3,3},W_2) &&= e(H,c_2)\,e(\psi_{2,1},P')\,e(\psi_{2,2},P')\,e(\psi_{2,3},W_3')\\
\text{For (9c):}\quad & e(d_1,G)\,e(p_{1,1},P)\,e(p_{2,1},P)\,e(p_{3,1},W_3) &&= e(\psi_{3,1},U')\,e(\psi_{3,3},W_1')\\
& e(d_2,G)\,e(p_{1,2},P)\,e(p_{2,2},P)\,e(p_{3,2},W_3) &&= e(\psi_{3,2},V')\,e(\psi_{3,3},W_2')\\
& e(d_3,G)\,e(p_{1,3},P)\,e(p_{2,3},P)\,e(p_{3,3},W_3) &&=\\
& && \quad e(H,c_3)\,e(\psi_{3,1},P')\,e(\psi_{3,2},P')\,e(\psi_{3,3},W_3')
\end{aligned}
$$

Re-randomization. Given commitments $\mathbf{c},\mathbf{d},\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3$ and proofs ψ_1,ψ_2,ψ_3, we can re-randomize the commitments by choosing $s'_{Yi}, s'_{Zi}, t'_{i,j} \leftarrow \mathbb{Z}_p$ for $1 \le i,j \le 3$ and setting (cf. Sect. 5.2)

$$
\widetilde{\mathbf{c}} := \begin{bmatrix} c_1 + s'_{Y3}U' + s'_{Y3}W'_1 \\ c_2 + s'_{Y2}V' + s'_{Y3}W'_2 \\ c_3 + s'_{Y3}P' + s'_{Y2}P' + s'_{Y3}W'_3 \end{bmatrix} \qquad \widetilde{\mathbf{d}} := \begin{bmatrix} d_1 + s'_{Z1}U' + s'_{Z3}W'_1 \\ d_2 + s'_{Z2}V' + s'_{Z3}W'_2 \\ d_3 + s'_{Z1}P' + s'_{Z2}P' + s'_{Z3}W'_3 \end{bmatrix}
$$

$$
\widetilde{\mathbf{p}}_i := \begin{bmatrix} p_{i,1} + t'_{i,1}U' + t'_{i,3}W'_1 \\ p_{i,2} + t'_{i,2}V' + t'_{i,3}W'_2 \\ p_{i,3} + s'_{Yi}H + t'_{i,1}P' + t'_{i,2}P' + t'_{i,3}W'_3 \end{bmatrix} \qquad \text{for } 1 \le i \le 3
$$

Note that $\widetilde{\mathbf{p}}_i$ not only re-randomizes $\mathbf{p}_i$ but at the same time updates the committed proofs π_i to the new randomness for the commitments to Y. The proofs ψ_i are updated as follows:

$$
\begin{aligned}
\widetilde{\psi}_{1,j} &:= \psi_{1,j} + t'_{1,j}U + t'_{3,j}W_1 \\
\widetilde{\psi}_{2,j} &:= \psi_{2,j} + t'_{2,j}V + t'_{3,j}W_2 \qquad\qquad \text{for } 1 \le j \le 3 \\
\widetilde{\psi}_{3,j} &:= \psi_{3,j} + s'_{Zj}G + t'_{1,j}P + t'_{2,j}P + t'_{3,j}W_3
\end{aligned}
$$

5.4 Proofs That Commitments Open to the Same Value

Given the extraction key, one can prove that two commitments open to the same value without knowledge of the randomness used when committed. We start by showing how to prove that a commitment (c_1,c_2,c_3) opens to zero: given the extraction key $ek = (\alpha,\beta)$ define the proof as $(\pi_1 := \frac{1}{\alpha}c_1, \pi_2 := \frac{1}{\beta}c_2)$. It satisfies the following relations: $e(\pi_1,U) = e(c_1,P)$, $e(\pi_2,V) = e(c_2,P)$, $c_3 = \pi_1 + \pi_2$.

It is easily seen that the proofs are perfectly correct and perfectly sound. In addition, they do not leak information about the opener's secret key, since they can be produced without knowledge of ek, given the randomness used to commit

and the "trapdoor" (r_1, r_2) for the W_i's: $c_1 = s_1 U + s_3 W_1 = \alpha(s_1 + s_3 r_1)P$, thus $\pi_1 = (s_1 + s_3 r_1)P$, and similarly $\pi_2 = (s_2 + s_3 r_2)P$. Now to show that $\mathbf{c}$ and $\mathbf{d}$ are two commitments to the same value, it suffices to prove that $\mathbf{c} - \mathbf{d}$ opens to 0.

6 Transferable Anonymous Constant-Size Fair E-Cash from Certificates

6.1 Formal Model

In our model for e-cash, there are the following protagonists: *users* $\mathcal{U}_i$ that—after registering—can withdraw, transfer and spend coins; the *system manager* $\mathcal{S}$, authorizing users to join the system; the *bank* $\mathcal{B}$, able to issue coins; *merchants* $\mathcal{M}_i$ who deposit the coins at the bank; the *double-spending detector* $\mathcal{D}$, that can detect if a coin was spent twice; and the *tracing authority* $\mathcal{T}$, able to trace users that misbehave in some way (e.g., tracing of a double spender or prosecution of criminal activities). The system comprises the following protocols and algorithms:

Setup	A protocol between $\mathcal{S}$ (who gets the manager key mk), $\mathcal{B}$ (who gets the issuing key ik), $\mathcal{D}$ (who gets dk), and $\mathcal{T}$ (who gets tk). The protocol also outputs the public parameters pp.
Join	A protocol between a user and $\mathcal{S}$ that registers the user in the system and gives him usk.
Withdraw	A protocol permitting a user to withdraw a coin from $\mathcal{B}$.
Transfer	A protocol between two users $\mathcal{U}_i$ and $\mathcal{U}_j$, where $\mathcal{U}_j$ gets a coin and a receipt from $\mathcal{U}_i$.
Spend	A protocol between a user and a merchant to spend a coin.
Detect	An algorithm enabling $\mathcal{D}$ to check for double spendings (without identifying the defrauder).
$\mathsf{Trace}_{\mathrm{DS}}$	A protocol conducted by $\mathcal{T}$ in order to trace a double spender.
$\mathsf{Trace}_{\mathrm{C}}$	An algorithm enabling $\mathcal{T}$ to match a withdrawal and a spending transcript of the same coin.
$\mathsf{Trace}_{\mathrm{S}}$	An algorithm that lets $\mathcal{T}$ reveal the identity of a spender from a spending transcript.

Besides correctness, which requires that honestly issued coins are accepted when transferred or spent by honestly registered users, and that the tracing algorithms work correctly, we define the following security notions for our model: *Anonymity of withdrawal* means that not even the bank colluding with the (double-spending) detector can tell to which withdrawal a coin corresponds. *Anonymity of transfer (or spending)* ensures that when transferring/spending a coin a user remains anonymous even with respect to the bank and malicious users the coin was transferred by.

Traceability of double spenders states that for each time a user spends a coin more than once he will be accused, whereas *Detectability of double spending*

$\mathbf{Exp}_{\mathcal{A}}^{\text{anon-with}}(k)$

- Experiment plays: honest users $\mathcal{U}_0$ and $\mathcal{U}_1$
- $\mathcal{A}$ impersonates: $\mathcal{S}$, $\mathcal{B}$, $\mathcal{D}$, users
- $\mathcal{U}_0$, $\mathcal{U}_1$ run Join and Withdraw with $\mathcal{A}$ impersonating $\mathcal{S}$ and $\mathcal{B}$, resp.
- $b \leftarrow \{0,1\}$; $\mathcal{A}$ receives the coin of $\mathcal{U}_b$
- $\mathcal{A}$ wins if it guesses b correctly

$\mathbf{Exp}_{\mathcal{A}}^{\text{trace-DS}}(k)$

- Experiment plays: honest $\mathcal{S}$, $\mathcal{B}$
- $\mathcal{A}$ impersonates: users
- $\mathcal{A}$ gets keys: tk, dk (thus $\mathcal{T}$, $\mathcal{D}$ semi-honest)
- $\mathcal{A}$ gets oracles Join, Withdraw, Spend to communicate with $\mathcal{S}$, $\mathcal{B}$ and $\mathcal{D}$, resp.
- The experiment runs Detect and Trace on the spent coins
- Let q and d be the number of Withdraw and Spend queries, resp.; let a be the number of accusations by Trace. Then $\mathcal{A}$ wins if $a < d - q$

$\mathbf{Exp}_{\mathcal{A}}^{\text{detect-DS}}(k)$

- Experiment plays: honest $\mathcal{B}$
- $\mathcal{A}$ impersonates: users, $\mathcal{S}$, $\mathcal{T}$
- $\mathcal{A}$ gets keys: dk (thus $\mathcal{D}$ semi-honest)
- $\mathcal{A}$ gets oracles Withdraw, Spend to communicate with $\mathcal{B}$ and $\mathcal{D}$, resp.
- The experiment runs Detect on the spent coins
- $\mathcal{A}$ wins if there where more accepted Spend than Withdraw calls and $\mathcal{D}$ does not detect double spending.

$\mathbf{Exp}_{\mathcal{A}}^{\text{anon-trans}}(k)$

- Experiment plays: honest users $\mathcal{U}_0$ and $\mathcal{U}_1$
- $\mathcal{A}$ impersonates: $\mathcal{S}$, $\mathcal{B}$, users
- $\mathcal{U}_0$ and $\mathcal{U}_1$ run Join with $\mathcal{A}$ impersonating $\mathcal{S}$
- $\mathcal{A}$ can ask withdrawals, transfers and spendings of $\mathcal{U}_0$ and $\mathcal{U}_1$.
- $b \leftarrow \{0,1\}$, $\mathcal{U}_b$ runs Transfer with $\mathcal{A}$ playing a user.
- $\mathcal{A}$ wins if it guesses b correctly.

$\mathbf{Exp}_{\mathcal{A}}^{\text{trace-C/S}}(k)$

- Experiment plays: honest $\mathcal{S}$, $\mathcal{B}$
- $\mathcal{A}$ impersonates: users, $\mathcal{D}$
- $\mathcal{A}$ gets keys: tk (thus $\mathcal{T}$ semi-honest)
- Oracles for $\mathcal{A}$: Join, Withdraw
- $\mathcal{A}$ spends a coin and wins if
  - the spending cannot be matched to a withdrawal (traceability of coins); or
  - $\mathsf{Trace}_{\mathrm{S}}$ returns $\perp$ (spender traceability)

$\mathbf{Exp}_{\mathcal{A}}^{\text{non-fram}}(k)$

- Experiment plays an honest user $\mathcal{U}^*$
- $\mathcal{A}$ can impersonate: $\mathcal{S}$, $\mathcal{B}$, $\mathcal{D}$, $\mathcal{T}$, users
- $\mathcal{U}^*$ runs Join with $\mathcal{A}$ impersonating $\mathcal{S}$
- $\mathcal{A}$ can ask the user to withdraw coins, transfer and receive them and spend coins
- $\mathcal{A}$ wins if
  - it outputs a proof accusing $\mathcal{U}^*$ of double spending, which $\mathcal{U}^*$ cannot contest.
  - $\mathcal{U}^*$ is accused of a spending it did not perform

Fig. 4. Security experiments for constant-size e-cash

means that Detect will determine if a coin was spent multiple times. *Non-frameability* guarantees that even if everyone else colludes against an honest user, he cannot be wrongfully accused of a spending he did not perform, nor of double spending. See Fig. 4 for the details of the experiments. As for the BSZ-model of group signatures, we call protagonists *semi-honest* if $\mathcal{A}$ impersonates them but however follows protocols as prescribed. Note that in the experiment for non-frameability, $\mathcal{U}^*$ behaves honestly, so if he is asked to spend more coins than he withdrew he refuses; moreover, a malicious tracer can always *accuse* an honest user of not having a receipt, which the latter counters by showing it.

We say an e-cash system is traceable, non-frameable, etc., if no p.p.t. adversary can win the respective game with non-negligible probability (non-negligibly more than $1/2$ for the anonymity notions).

6.2 Instantiation

Overview. The core of a coin in our system is a certificate from Sect. 3.2. Defining withdrawal as partially blind issuing guarantees that the bank does not know the last component C_5. Certificates were designed to consist of elements of $\mathbb{G}$ so that their verification relations are paring-product equations; the user can thus encrypt (in Groth-Sahai terminology: commit to) the coin and prove validity. Moreover, each time the coin is transferred, the receiver can re-randomize the encryption (cf. Sect. 5.2), which guarantees unlinkable anonymity.

To check for double spendings, the detector will get the decryption key to compare encrypted certificates. However, this straight-forward approach would not guarantee user anonymity when bank and detector cooperate. The blind component C_5 is thus encrypted under a *different* key than the rest (in Sect. 5.3 we showed how to construct the corresponding proofs). The detector gets only the key to decrypt C_5, which suffices to detect double spending. Since the the first 4 components remain hidden from the detector, *partial* blindness of certificates suffices. The other decryption key is given to the tracer, which enables tracing of a coin by comparing C_3 which is known to the bank.

The receipts, given when transferring and spending coins, are group signatures on them, the signing keys for which the users get when joining the system. This guarantees user traceability, while preserving anonymity (only the tracer, holding the group-signature opening key, can reveal users' identities). To identify a double spender, the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts. To guarantee soundness of tracing, we must ensure that each signature corresponds to at most one transfer. We achieve this by having the receiver choose a nonce which is added to the message the sender must sign.

Details. Let $\mathcal{GS} = (\mathsf{Setup}_{\mathcal{GS}}, \mathsf{Join}_{\mathcal{GS}}, \mathsf{GSign}_{\mathcal{GS}}, \mathsf{GVer}_{\mathcal{GS}})$ be a dynamic non-frameable group-signature scheme.[6] Let $\mathcal{H}\colon \mathbb{G}^* \to \{0,1\}^n$ be a collision-resistant hash function.

Setup.
- Set up a group signature scheme $\mathcal{GS}$ such that $\mathcal{S}$ is the group's issuer (group manager) and $\mathcal{T}$ gets opening key ok. The group verification key gvk is added to pp.
- Produce two keys for linear commitments $ck_{\mathcal{T}}$ and $ck_{\mathcal{D}}$. The corresponding extraction keys $ek_{\mathcal{T}}$ and $ek_{\mathcal{D}}$ are given to $\mathcal{T}$ (thus $tk = (ek_{\mathcal{T}}, ek_{\mathcal{D}}, ok)$. $\mathcal{D}$ receives $dk := ek_{\mathcal{D}}$.

[6] Encrypting the certified signatures from Sect. 4 and proving validity by adding a Groth-Sahai proof yields a (CPA-anonymous) non-frameable group signature scheme that does not require any further assumptions.

– Set up the CRS (if any) for the blind certificate-issuing scheme from Sect. 3.2. $\mathcal{B}$ picks issuing key $ik := \omega \leftarrow \mathbb{Z}_p$, adds $\Omega := \omega G$ to pp, and gets a group signing key $gsk_{\mathcal{B}}$ by joining $\mathcal{GS}$.

Join. A user $\mathcal{U}_i$ joins the system by running $\mathsf{Join}_{\mathcal{GS}}$ with $\mathcal{S}$ to obtain her group signing key gsk_i.

Withdraw. $\mathcal{U}_i$ runs the issuing protocol (Fig. 2) with $\mathcal{B}$ to get $(C_1, \dots C_5) \in \mathbb{G}^5$ satisfying

$$e(C_1, \Omega + C_2) = e(K + C_4, G) \qquad \begin{aligned} e(C_2, H) &= e(G, C_3) \\ e(C_4, H) &= e(G, C_5) \end{aligned} \tag{12}$$

$\mathcal{B}$ also gives the user a "receipt" $R_{\mathcal{B}} \leftarrow \mathsf{GSign}_{\mathcal{GS}}(gsk_{\mathcal{B}}, \mathcal{H}(C_1, C_2, C_3, \mathcal{U}_i))$.[7] $\mathcal{U}_i$ verifies the certificate and $R_{\mathcal{B}}$ and makes the following commitments:

$$\mathbf{c}_i := \mathsf{Com}(ck_{\mathcal{T}}, C_i), \quad \text{for } 1 \le i \le 4 \qquad \mathbf{c}_5 := \mathsf{Com}(ck_{\mathcal{D}}, C_5)$$

and proofs Φ_1, Φ_2, Φ_3 for the committed values satisfying each of the equations in (12). Φ_1 and Φ_2 are regular Groth-Sahai proofs; for the last equation on commitments under different keys, see Sect. 5.3. We call $(\vec{\mathbf{c}}, \vec{\Phi})$ a *coin*, and refer to the full version [FPV09] for its concrete construction.

Transfer / Spend. When user $\mathcal{U}_i$ transfers a coin $(\vec{\mathbf{c}}, \vec{\Phi})$ to user $\mathcal{U}_j$, she sends $R \leftarrow \mathsf{GSig}_{\mathcal{GS}}(gsk_{\mathcal{U}_i}, \mathcal{H}(\vec{\mathbf{c}}, \mathcal{U}_j, N))$ as well, where N is a nonce set by $\mathcal{U}_j$. The receiver $\mathcal{U}_j$ checks correctness of $(\vec{\mathbf{c}}, \vec{\Phi})$ and R, re-randomizes $\vec{\mathbf{c}}$ and updates $\vec{\Phi}$ (cf. Sects. 5.2 and 5.3). Spending is defined as transferring.

Detect. After receiving new a coin, $\mathcal{D}$ uses extraction key $ek_{\mathcal{D}}$ to open $\mathbf{c}_5$: $C_5 := \mathsf{Extr}(ek_{\mathcal{D}}, \mathbf{c}_5)$ (cf. Sect. 5.1). He compares the tag C_5 with that of previously received coins to see if a coin was spent twice, in which case he charges $\mathcal{T}$ to trace the double spender.

Tracing of DS

– If multiple spendings $(\vec{\mathbf{c}}^{(i)}, \vec{\Phi}^{(i)}, R^{(i)})$ with $\mathsf{Extr}(ek_{\mathcal{D}}, \mathbf{c_5}^{(i)}) = C_5^*$ for all i were detected, the tracer uses the key ok to open the signatures $R^{(i)}$ in order to reveal users $\mathcal{U}_0^{(i)}$.

– Each $\mathcal{U}_0^{(i)}$ has to *prove legal acquisition* of his coin, which a user $\mathcal{U}$ does as follows:

- If the coin was obtained from the bank, show $C = (C_1, \dots, C_5)$ and the receipt $R_{\mathcal{B}}$.
  $\mathcal{T}$ accepts if C is valid, $\mathsf{GVer}_{\mathcal{GS}}(gvk, \mathcal{H}(C_1, C_2, C_3, \mathcal{U}), R_{\mathcal{B}}) = 1$ and $C_5 = C_5^*$.

[7] Abusing notation slightly, we let $\mathcal{U}_i$ be a unique encoding of the user's identity in $\mathbb{G}$. Note that for the receipts from the issuer, no nonce is required, since the user contributes to the randomness of the certificate.

- If the coin was received from a user, show the receipt R received with it, and show $(\vec{\mathbf{c}}', \vec{\Phi}')$, the received coin (i.e., before re-randomizing it), and the nonce N.
  $\mathcal{T}$ accepts if $(\vec{\mathbf{c}}', \vec{\Phi}')$ is valid, $\mathsf{GVer}_{\mathcal{GS}}(gvk, \mathcal{H}(\vec{\mathbf{c}}', \mathcal{U}, N), R) = 1$ and $\mathsf{Extr}(ek_{\mathcal{D}}, \mathbf{c}'_5) = C^*_5$.

– In the second case (receipt produced by a user), $\mathcal{T}$ opens R to $\mathcal{U}_1^{(i)}$, who in turn has to prove legal acquisition of the coin. Moreover, the tracer only accepts a receipt if it has not been given to him before.
– Continuing this process for every i produces a chain of users $\mathcal{U}_0^{(i)}, \mathcal{U}_1^{(i)}, \ldots$ which either ends with the bank, or with a user failing to prove legal acquisition—in which case that user is accused.
– Correctness of tracing is proven by proving correctness of opening of group signatures and proving that two commitments contain the same certificate using the techniques from Sect. 5.4.

Tracing of coins and users. Given $ek_{\mathcal{T}}$, the tracer can recover C_3 from a coin and thus match withdrawn coins to spent coins. Spender anonymity is revoked by opening the group signature.

6.3 Security Results

We briefly argue why our instantiation satisfies the security definitions from Sect. 6.1. Each property follows by a straight-forward reduction to the security of the underlying building blocks.

Detectability and traceability of double spenders. (I) Assuming an honest bank, every certificate is only issued once with all but negligible probability; (II) by unforgeability of certificates (Theorem 8) and soundness of the WI proofs, opening all d spent coins leads to at most q different certificates, where q is the number of $\mathsf{Withdraw}$ queries. This proves detectability.

For every i let $s^{(i)}$ be the number of times certificate $C^{(i)}$ was spent. Then the tracing algorithm produces $s^{(i)}$ lists of users, beginning with the spenders and linked by their certificates. Unforgeability of group signatures and (I) guarantees that only one such list ends with the bank. Since $s^{(i)} - 1$ users are thus accused and by (II), we have $a = \sum_{i=1}^{q}(s^{(i)} - 1) = d - q$, which proves traceability.

Non-frameability. If $\mathcal{U}^*$ uses a random nonce each time then by collision resistance of $\mathcal{H}$, the probability of receiving the same valid receipt twice is negligible. The user can only be provably accused if he spent/transferred a coin of which he cannot justify acquisition. Non-frameability of group signatures guarantees that $\mathcal{U}^*$ only has to justify coins he actually transferred—and for each such coin he possesses a valid receipt. Note that if a malicious user transfers the same coin (possibly as two different randomizations) twice to $\mathcal{U}^*$ then $\mathcal{U}^*$ has two different signatures (due to the nonce) and can thus justify both coins.

Anonymity. Anonymity of withdrawal follows from partial blindness of issuing (indistinguishability of C_5) and witness indistinguishability of the commitments

$(\mathbf{c}_1, \ldots, \mathbf{c}_4)$ under key $ck_{\mathcal{T}}$. Anonymity of transfer follows from WI of commitments under $ck_{\mathcal{T}}$ and $ck_{\mathcal{D}}$ and anonymity of group signatures.

Traceability. Traceability of coins follows from soundness of the WI proofs and unforgeability of certificates; traceability of spenders follows from traceability of group signatures.

Acknowledgments

The authors would like to thank the members of the PACE research project for the fruitful discussions that led to the new primitive discussed in this paper. This work was supported by the French ANR 07-TCOM-013-04 PACE Project, the European Commission through the IST Program under Contract ICT-2007-216646 ECRYPT II, and EADS.

References

[AO00] Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 271–286. Springer, Heidelberg (2000)

[BB04] Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)

[BBS04] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)

[BCC+09] Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable proofs and delegatable anonymous credentials. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer, Heidelberg (2009)

[BFPW07] Boldyreva, A., Fischlin, M., Palacio, A., Warinschi, B.: A closer look at PKI: Security and efficiency. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 458–475. Springer, Heidelberg (2007)

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)

[BSZ05] Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: The case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005)

[BW07] Boyen, X., Waters, B.: Full-domain subgroup hiding and constantsize group signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 1–15. Springer, Heidelberg (2007)

[CG07] Canard, S., Gouget, A.: Divisible E-cash systems can be truly anonymous. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 482–497. Springer, Heidelberg (2007)

[CG08] Canard, S., Gouget, A.: Anonymity in transferable E-cash. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 207–223. Springer, Heidelberg (2008)

[Cha83] Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO 1982, pp. 199–203. Plenum Press, New York (1983)

[Cha84] Chaum, D.: Blind signature system. In: Chaum, D. (ed.) CRYPTO 1983, p. 153. Plenum Press, New York (1984)

[CP92] Chaum, D., Pedersen, T.P.: Transferred cash grows in size. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 390–407. Springer, Heidelberg (1993)

[Cv91] Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)

[DH76] Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)

[EO94] Eng, T., Okamoto, T.: Single-term divisible electronic coins. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 306–319. Springer, Heidelberg (1995)

[FP09] Fuchsbauer, G., Pointcheval, D.: Proofs on encrypted values in bilinear groups and an application to anonymity of signatures. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 132–149. Springer, Heidelberg (2009), `http://eprint.iacr.org/2008/528`

[FPV09] Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Transferable anonymous constant-size fair e-cash. Cryptology ePrint Archive, Report 2009/146 (2009), `http://eprint.iacr.org/`

[FS87] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)

[GOS06] Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (2006)

[Gro06] Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)

[Gro07] Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)

[GS08] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

[Kil06] Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)

[NHS99] Nakanishi, T., Haruna, N., Sugiyama, Y.: Unlinkable electronic coupon protocol with anonymity control. In: Zheng, Y., Mambo, M. (eds.) ISW 1999. LNCS, vol. 1729, pp. 37–46. Springer, Heidelberg (1999)

[Oka06] Okamoto, T.: Efficient blind and partially blind signatures without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 80–99. Springer, Heidelberg (2006)

[OO90] Okamoto, T., Ohta, K.: Disposable zero-knowledge authentications and their applications to untraceable electronic cash. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 481–496. Springer, Heidelberg (1990)

[OO92] Okamoto, T., Ohta, K.: Universal electronic cash. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 324–337. Springer, Heidelberg (1992)

[PS00] Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)

[Sch90] Schnorr, C.-P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)

[Sho97] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)

[SPC95] Stadler, M., Piveteau, J.-M., Camenisch, J.: Fair blind signatures. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 209–219. Springer, Heidelberg (1995)

[vSN92] von Solms, S.H., Naccache, D.: On blind signatures and perfect crimes. Computers & Security 11(6), 581–583 (1992)

[Wat05] Waters, B.R.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)