Compact E-Cash and Simulatable VRFs Revisited

Mira Belenkiy[1], Melissa Chase[2], Markulf Kohlweiss[3], and Anna Lysyanskaya[4]

[1] Microsoft, mibelenk@microsoft.com
[2] Microsoft Research, melissac@microsoft.com
[3] KU Leuven, ESAT-COSIC / IBBT, markulf.kohlweiss@esat.kuleuven.be
[4] Brown University, anna@cs.brown.edu

Abstract. Efficient non-interactive zero-knowledge proofs are a powerful tool for solving many cryptographic problems. We apply the recent Groth-Sahai (GS) proof system for pairing product equations (Eurocrypt 2008) to two related cryptographic problems: compact e-cash (Eurocrypt 2005) and simulatable verifiable random functions (CRYPTO 2007). We present the first efficient compact e-cash scheme that does not rely on a random oracle. To this end we construct efficient GS proofs for signature possession, pseudo randomness and set membership. The GS proofs for pseudorandom functions give rise to a much cleaner and substantially faster construction of simulatable verifiable random functions (sVRF) under a weaker number theoretic assumption. We obtain the first efficient fully simulatable sVRF with a polynomial sized output domain (in the security parameter).

1 Introduction

Since their invention [BFM88] non-interactive zero-knowledge proofs played an important role in obtaining feasibility results for many interesting cryptographic primitives [BG90,GO92,Sah99], such as the first chosen ciphertext secure public key encryption scheme [BFM88,RS92,DDN91]. The inefficiency of these constructions often motivated independent practical instantiations that were arguably conceptually less elegant, but much more efficient ([CS98] for chosen ciphertext security).

We revisit two important cryptographic results of pairing-based cryptography, compact e-cash [CHL05] and simulatable verifiable random functions [CL07], that have very elegant constructions based on non-interactive zero-knowledge proof systems, but less elegant practical instantiations. Our results combine the best of both worlds, a clean design and an efficient implementation.

Compact e-cash. Electronic cash (e-cash) was introduced by Chaum [Cha83] as an electronic analogue of physical money and has been a subject of ongoing research since then [CFN90,FY92,CP93,Bra93,SPC95,FTY96,Tsi97]. The participants in an e-cash system are users who withdraw and spend e-cash; a bank that creates e-cash and accepts it for deposit, and merchants who offer goods and services in exchange for e-cash, and then deposit the e-cash to the bank. The main security requirements are (1) anonymity: even if the bank and the merchant and all the remaining users collude with each other, they still cannot distinguish Alice's purchases from Bob's; (2) unforgeability: even if all the users and all the merchants collude against the bank, they still cannot deposit more money than they withdrew.

Unfortunately, it is easy to see that, as described above, e-cash is useless. The problem is that here money is represented by data, and it is possible to copy data. Unforgeability will guarantee that the bank will only honor at most one of copy of a given coin for deposit and will reject the others. Anonymity will guarantee that there is no recourse against such a cheating Alice. So one of the merchants will be cheated. There are two known remedies against this double-spending behavior. The first remedy is on-line e-cash [Cha83], where the bank is asked to vet a coin before the spend protocol can terminate successfully. The second remedy is off-line e-cash, introduced by Chaum, Fiat and Naor [CFN90]. The additional requirement of an offline e-cash system is (informally) that no coin can be double-spent without revealing the identity of the perpetrator.

A further development in the literature on e-cash was compact e-cash [CHL05]. In compact e-cash, the user withdraws N coins in a withdrawal protocol whose complexity is $O(\log N)$ rather than $O(N)$. Similarly, the resulting wallet requires storage size $(\log N)$ rather than $O(N)$. The main idea is as follows: in the withdrawal protocol, a user obtains the Bank's signature on (x, s, t), where s and t are random seeds of a pseudorandom function (PRF) $F_{(\cdot)}(\cdot)$ and x is the user's identifier. In the spend protocol, a serial number of the ith coin is computed as $S = F_s(i)$, and a double spending equation is computed as $T = x + RF_t(i)$, where R is a random challenge by the merchant. The coin itself consists of (S, T, R, π), where π is a non-interactive zero-knowledge proof of knowledge of the following values: x, s, t, i, σ where σ is the Bank's signature on (x, s, t), $1 \leq i \leq N$, $S = F_s(i)$ and $T = x + RF_t(i) \bmod q$. If g is a generator of a group G of order q, and G is the range of the PRF $F_{(\cdot)}(\cdot)$, then the double-spending equation can instead be computed as $T = g^x F_t(i)^R$. It is easy to see that two double-spending equations for the same t, i but different R's allow us to compute g^x. It was shown that this approach yields a compact e-cash scheme [CHL05]. Later, this was extended to so-called e-tokens [CHK$^+$06] that allow up to k anonymous transactions per time period (for example, this would correspond to subscriptions to interactive game sites or anonymous sensor reports).

Thus, we see that compact e-cash and variants such as e-tokens can be obtained from a signature scheme, a pseudorandom function, and a non-interactive zero-knowledge (NIZK) proof system for the appropriate language. However, until now no efficient instantiations of the NIZK proofs could be given, and all practical instantiations of compact e-cash had to derive the non-interactive proofs from interactive proofs via the Fiat-Shamir heuristic [FS87] which is known not to yield provably secure constructions [GK03]. It seemed that, perhaps, random oracle based techniques were necessary to achieve such schemes efficiently. We show here that this is not the case.

Challenges and Techniques. Until the recent proof system of Groth and Sahai [GS07], there were no efficient NIZK proof systems for languages most heavily used in cryptographic constructions (such as languages of true statements about discrete logarithm representations and bilinear pairings). However, constructing an efficient provably-secure compact e-cash scheme is not simply a matter of replacing the Fiat-Shamir based NIZK proofs with the Groth-Sahai system. There are several issues that arise when we attempt to apply the Groth-Sahai proofs. First, recall that the Groth-Sahai system only works for proofs of particular types of statements. Thus, we must find a PRF and a signature scheme where verification can be phrased in terms of such statements. In the case of the PRF, we use a modification of the Dodis-Yampolskiy VRF [DY05], which outputs elements of bilinear group G_1. We show that this is secure under the assumption that DDHI holds in this group. [5]

For the signature scheme, we note that verification of Boneh-Boyen signatures [BB04b] can be phrased as a pairing product equation. However, as noted in Belenkiy et al. [BCKL08], because Groth-Sahai proofs are only partially extractable, we need a stronger unforgeability. Here we need that it be impossible to produce $F(m), \mathsf{Sign}_{sk}(m)$ for an unsigned message m, where $F(m)$ is a value that can be extracted from a commitment to m. Belenkiy et al. gave a construction which satisfies this definition, but only allows signatures on a single message. We need the bank to be able to sign multiple message blocks, thus we extend that construction to construct a multi-block P-signature scheme. We also show that issuing can be done efficiently using more recent techniques given in [BCC$^+$09]. (The original [BCKL08] construction relied on general two party computation for arithmetic circuits.)

We also need to be able to prove that the coin value falls within a given range. The original Camenisch et al. construction uses a technique by [Bou00], which relies on the fact that the underlying RSA group has unknown order. Groth-Sahai proofs, on the other hand, rely on the cryptographic bilinear group model, and it is not known how to construct such groups with unknown order. Thus, we must use a different technique for our range proofs. We follow the basic concept of [TS06,CCS08], and implement the range proofs using the new P-signatures mentioned above.

Finally, while Groth and Sahai present a NIZK proof system for a large class of statements, their simpler witness indistinguishable proof system is much more efficient. Thus, we specifically design our protocols to use NIZK proofs only when necessary. As a result, we obtain a construction that is almost competitive in efficiency with the original Camenisch et al. construction.

E-cash construction. Our construction is in the common parameters model and relies on several number-theoretic assumptions. Our first building block is a signature scheme and an unconditionally binding commitment scheme that allows for an efficient proof of knowledge of a signature on a set of commit-

[5] We note that the original Camenisch et al. [CHL05] construction used a similar PRF based on DDHI in a standard prime order group (without a bilinear map). They then proved correctness of each PRF output using the Fiat-Shamir heuristic.

ted values, as well as for an efficient protocol for getting a committed value signed. This is done by extending the P-signature construction of Belenkiy et al. [BCKL08], which only allows to sign single values, and incorporating the techniques from [BCC$^+$09]. In our construction we will also use P-signatures, together with the techniques of [CCS08] (that relied on interactive proofs) to obtain efficient non-interactive interval proofs.

Our second building block is a pseudorandom function and an unconditionally binding commitment scheme $\mathsf{Com}(.,.)$ (the same as for the P-signature scheme) with an efficient proof system for the serial number S and the double spending tag T.

Simulatable verifiable random functions. Our main observation is that the NIZK proof for a compact e-cash serial number, a proof of the language $L_F = \{S, C_y, C_s \mid \exists s, y, r_s, r_y \text{ such that } S = F_s(y), C_y = \mathsf{Com}(y, r_y), C_s = \mathsf{Com}(s, r_s)\}$ is a special case of a simulatable verifiable random function (sVRF), introduced by Chase and Lysyanskaya [CL07]. Chase and Lysyanskaya gave an efficient construction of a multi-theorem non-interactive zero-knowledge proof system for any language L from a single-theorem one for the same language (while other single-theorem to multi-theorem transformations required the Cook-Levin reduction [Coo71] to an NP-complete language first).

Chase and Lysyanskaya [CL07] gave two constructions for sVRFs. The first is based on generic non-interactive zero-knowledge proofs and is therefore impractical. The second construction is based on composite order bilinear pairings [BGN05,FST06], and has several shortcomings. In particular, its range is either only logarithmic in the security parameter or it is only weakly simulatable. Our fully simulatable construction is thus more efficient by a factor of the security parameter; it is also designed in a way that is more modular and therefore easier to understand (and improve). Finally, it relies on a somewhat weaker assumption. Therefore, we believe this result will be of independent interest.

Our contribution and outline of the paper. We present the first P-signature scheme for multiple messages, the first fully simulatable VRF with polynomial sized output domain, and the first efficient compact e-cash scheme that does not rely on random oracles. (The security of conventional e-cash was, e.g., studied in [JLO97,STS99,Tro05].) The rest of the paper is organized as follows. In Section 2 we discuss our assumptions and recall useful results about non-interactive zero-knowledge. In Section 3 we define and construct our new P-signature scheme for message blocks. Section 4 and Section 5 revisit simulatable verifiable random functions and compact e-cash respectively.

2 Preliminaries

In this section we list our assumptions and recall some useful results about non-interactive zero-knowledge proofs (NIZK).

A function ν is *negligible* if, for every integer c, there exists an integer K such that for all $k > K$, $|\nu(k)| < 1/k^c$. A problem is said to be *hard* (or *infeasible)* if there exists no probabilistic polynomial time (p.p.t.) algorithm to solve it.

Bilinear Pairings. Let G_1, G_2, and G_T be groups of prime order p. The map $e : G_1 \times G_2 \rightarrow G_T$ must satisfy the following properties: (a) *Bilinearity*: a map $e : G_1 \times G_2 \rightarrow G_T$ is bilinear if $e(a^x, b^y) = e(a, b)^{xy}$; (b) *Non-degeneracy*: for all generators $g \in G_1$ and $h \in G_2$, $e(g, h)$ generates G_T; (c) *Efficiency*: There exists a p.p.t. algorithm $\mathsf{BMGen}(1^k)$ that outputs $(p, G_1, G_2, G_T, e, g, h)$ to generate the bilinear map and an efficient algorithm to compute $e(a, b)$ for any $a \in G_1$, $b \in G_2$.

Assumptions. The security of our scheme is based on previously proposed number-theoretic assumptions. The unforgeability of our P-signature construction relies on the TDH [BCKL08] and the HSDH [BW07] assumptions; pseudo-randomness is based on the q-DDHI assumption [BB04a,CHL05]; and the zero-knowledge of the Groth-Sahai proof system rests on the XDH or DLIN assumption [GS07].

Definition 1 (Triple DH). *On input $g, g^x, g^y \in G_1$, $h, h^x \in G_2$, and $\{c_i, g^{1/(x+c_i)}\}_{i=1\ldots q}$ for random x, y, and $c_1, \ldots, c_q$, it is computationally infeasible to output a tuple $(h^{\mu x}, g^{\mu y}, g^{\mu xy})$ for $\mu \neq 0$.*

Definition 2 (Hidden SDH). *On input $g, g^x, u \in G_1$, $h, h^x \in G_2$ and $\{g^{1/(x+c_\ell)}, h^{c_\ell}, u^{c_\ell}\}_{\ell=1\ldots q}$ for random x and $c_1, \ldots c_q$, it is computationally infeasible to output a new tuple $(g^{1/(x+c)}, h^c, u^c)$.*

Definition 3 (q-DDHI). *On input $g, g^\alpha, g^{\alpha^2}, \ldots g^{\alpha^q} \in G$ for a random $\alpha \leftarrow Z_p$, it is computationally infeasible to distinguish $g^{\frac{1}{\alpha}}$ from a random element of G with probability non-negligibly better than $1/2$.*

Our sVRF requires that the q-DDHI assumption holds either in G_1 or G_2. Without loss of generality we fix this group to be G_1. Note that this is slightly stronger than the assumption used in [DY05] to construct an efficient VRF (there the challenge is $e(g, h)^{\frac{1}{\alpha}}$ or a random element of G_T). However, it is still weaker than the BDHBI assumption used in the sVRF construction in [CL07].

Composable Non-Interactive Proofs. We review composable non-interactive proof systems. Let $R(\cdot, \cdot)$ be any polynomial-time computable relation. A non-interactive proof system for an NP language allows a prover to convince a verifier of the truth of the statement $\exists x : R(y, x)$ about instance y using witness x. Non-interactive proof systems use a common reference string *params* as output by $\mathsf{Setup}(1^k)$ that is common input to both the $\pi \leftarrow \mathsf{Prove}(\mathit{params}, y, x)$ and $\mathsf{accept/reject} \leftarrow \mathsf{Verify}(\mathit{params}, x, \pi)$ algorithms. This notion can be generalized for a relation $R(\mathit{params}, y, x)$ parameterized by *params*.

Informally, zero-knowledge captures the notion that a verifier learns nothing from the proof but the truth of the statement. Witness-indistinguishability is a weaker notion that guarantees that the verifier learns nothing about which witness was used in the proof.

In a *composable* (under the definition of Groth and Sahai [GS07]) non-interactive *witness indistinguishable* proof system there exists a $\mathsf{SimSetup}$ algorithm that outputs *params* together with a trapdoor *sim*, such that (1) *params* output by $\mathsf{SimSetup}$ are indistinguishable from those output by Setup; (2) the output of Prove using these parameters is perfectly witness-indistinguishable (in other words, even if there are two witnesses to a statement, they induce identical distributions on the proofs). Composable non-interactive *zero-knowledge* further means that there exists an algorithm $\mathsf{SimProve}$ that outputs a simulated proof using *sim* and the output of $\mathsf{SimProve}$ is distributed identically to that of Prove when given the simulated parameters. The big advantage of a composable definition is that it is fairly simple and easy to work with, and yet it still implies the standard multi-theorem definitions.

Composable proofs about commitments. The prover and verifier frequently get some set of commitments $(C_1, \ldots, C_n)$ as common input. The prover wants to show that a statement about instance $y = (C_1, \ldots, C_n, \mathsf{Condition})$ holds. The witness to the statement is $(x_1, open_1, \ldots, x_n, open_n, z)$, where $(x_i, open_i)$ is the opening of commitment C_i, while z is some value that has nothing to do with the commitments. The relation is $R = \{(params, y, x) | C_1 = \mathsf{Com}(params, x_1, open_1) \wedge \ldots \wedge C_n = \mathsf{Com}(params, x_n, open_n)$
$\wedge\ \mathsf{Condition}(params, x_1, \ldots, x_n, z)\}$.

Summary of Groth-Sahai proofs. Groth and Sahai [GS07] give a composable witness-indistinguishable proof system that lets us efficiently prove statements in the context of groups with bilinear maps. Let $params_{BM} = (p, G_1, G_2, G_T, e, g, h)$ be the setup for pairing groups of prime order p.

In a Groth-Sahai proof, the prover and the verifier both know $\{a_q\}_{q=1\ldots Q} \in G_1$, $\{b_q\}_{q=1\ldots Q} \in G_2$, $t \in G_T$, and $\{\alpha_{q,m}\}_{q=1\ldots Q, m=1\ldots M}, \{\beta_{q,n}\}_{q=1\ldots Q, n=1\ldots N} \in Z_p$. In addition, they both know commitments $\{C_m\}_{m=1\ldots M}$ and $\{D_n\}_{n=1\ldots N}$ to values in G_1 and G_2 respectively. For each commitment C_m and D_n the prover knows the opening information and the committed value $x_m \in G_1$ or $y_n \in G_2$ respectively ($m = 1 \ldots M$, $n = 1 \ldots N$).

Groth-Sahai proofs prove that the values in these commitments fulfill the pairing product equation $\prod_{q=1}^{Q} e(a_q \prod_{m=1}^{M} x_m^{\alpha_{q,m}}, b_q \prod_{n=1}^{N} y_n^{\beta_{q,n}}) = t$.

Groth-Sahai commitments. Throughout the paper we will use Groth-Sahai commitments (GSCom) in our constructions. Under the parameters output by Setup they are perfectly binding. We will sometimes make use of the fact that they are also extractable.

3 A Multi-block P-Signature Scheme

Belenkiy et al. [BCKL08] intruduced signatures with efficient non-interactive proofs of signature possession. Their construction can only be used to sign a single message block. In this section, we briefly review the definition of a P-signature scheme and construct a multi-block P-signature scheme.

Before defining and constructing P-signatures, we recall some particulars about the way Belenkiy et al. use Groth Sahai proofs. In addition to the zero-knowledge or witness indistinguishability property they rely on the fact that they are partially extractable (f-extractable [BCKL08]) proofs of knowledge about committed values. By '$x \,\mathrm{in}\, C$' we denote that there exists *open* such that $C = \mathsf{Com}(x, \mathit{open})$. Following Camenisch and Stadler [CS97a] and Belenkiy et al. [BCKL08], we use the following notation to express an f-extractable NIPK for instance $y = (C_1, \ldots, C_n, \mathsf{Condition})$ with witness $w = (x_1, \mathit{open}_1, \ldots, x_n, \mathit{open}_n, z)$:

$$\pi \leftarrow \mathsf{NIPK}[x_1 \,\mathrm{in}\, C_1, \ldots, x_n \,\mathrm{in}\, C_n]\{(\ f(\mathit{params},\ (x_1, \mathit{open}_1, \ldots, x_n, \mathit{open}_n, y)\ )\ ) : \\ \mathsf{Condition}(\mathit{params}, x_1, \ldots, x_n, z)\}.$$

For such a proof there exists a polynomial-time extractor ($\mathsf{ExtractSetup}, \mathsf{Extract}$). $\mathsf{ExtractSetup}(1^k)$ outputs $(\mathit{td}, \mathit{params})$ where *params* is distributed identically to the output of $\mathsf{Setup}(1^k)$. For all p.p.t. adversaries $\mathcal{A}$, the probability that $\mathcal{A}(1^k, \mathit{params})$ outputs (y, π) such that $\mathsf{Verify}(\mathit{params}, y, \pi) = \mathsf{accept}$ and $\mathsf{Extract}(\mathit{td}, y, \pi)$ fails to extract $f(\mathit{params}, (x_1, \mathit{open}_1, \ldots, x_n, \mathit{open}_n, z))$, such that x_i is the content of the commitment C_i, and $\mathsf{Condition}(\mathit{params}, x_1, \ldots, x_n, z)$ is satisfied is negligible in k.

Groth-Sahai proofs use commitments $\mathsf{GSCom}(x, \mathit{open})$ that allow to extract the value x but not the opening *open*. In short, Groth-Sahai proofs are f-extractable proofs of the following form

$$\mathsf{NIPK}[\{x_m \,\mathrm{in}\, C_m\}_{m=1}^{M}, \{y_n \,\mathrm{in}\, D_n\}_{n=1}^{N}]\{(x_1, ..., x_M, y_1, ..., y_N) : \\ \prod_{q=1}^{Q} e(a_q \prod_{m=1}^{M} x_m^{\alpha_{q,m}}, b_q \prod_{n=1}^{N} y_n^{\beta_{q,n}}) = t\}.$$

In our P-signature scheme we will commit to a message $m \in Z_p$ as $\mathsf{Com}(m, (\mathit{open}_1, \mathit{open}_2)) = (\mathsf{GSCom}(h^m, \mathit{open}_1), \mathsf{GSCom}(u^m, \mathit{open}_2))$. Such a commitment allows to extract $F(m) = (h^m, u^m)$.

3.1 Definition of Multi-block P-Signatures

A signature scheme consists of four algorithms: Setup, Keygen, Sign, and $\mathsf{VerifySig}$. $\mathsf{Setup}(1^k)$ generates the public parameters *params*. $\mathsf{Keygen}(\mathit{params})$ generates a signing key pair $(\mathit{pk}, \mathit{sk})$. $\mathsf{Sign}(\mathit{params}, \mathit{sk}, m)$ computes a signature σ on m. $\mathsf{VerifySig}(\mathit{params}, \mathit{pk}, m, \sigma)$ outputs accept if σ is a valid signature on m, reject otherwise. We extend this definition to support multi-block messages $\boldsymbol{m} = (m_1, \ldots m_n)$.

Definition 4 (F-Secure Signature Scheme [BCKL08]). *Let F be an efficiently computable bijection. With not necessarily efficient inverse F^{-1}. We say that a signature scheme is F-secure (against adaptive chosen message attacks) if it has the following properties: (a)* Correctness: $\mathsf{VerifySig}$ *always accepts a signature σ obtained using the* Sign *algorithm; (b)* F-Unforgeability: *no adversary should be able to output values $(F_1, \ldots, F_n, \sigma)$ such that for $\boldsymbol{m} = (F^{-1}(F_1), \ldots, F^{-1}(F_n))$ algorithm* $\mathsf{VerifySig}(\mathit{params}, \mathit{pk}, \boldsymbol{m}, \sigma) = \mathsf{accept}$ *unless he has previously obtained a signature on $\boldsymbol{m}$.*

Definition 5 (P-Signature Scheme [BCKL08]). *A P-Signature scheme combines an F-secure signature scheme with a commitment scheme and three protocols:*

1. *An algorithm* $\mathsf{SigProve}(params, pk, \sigma, \boldsymbol{m} = (m_1, \dots, m_n))$ *that generates commitments* $(C_1, \dots, C_n)$ *and a NIZK proof* $\pi \leftarrow \mathsf{NIPK}[m_1 \operatorname{in} C_1, \dots, m_n \operatorname{in} C_n]\{ (F(m_1), \dots F(m_n), \sigma) : \mathsf{VerifySig}(params, pk, \boldsymbol{m}, \sigma) = \mathsf{accept}\}$, *and the corresponding* $\mathsf{VerifyProof}(params, pk, \pi, (C_1, \dots, C_n))$ *algorithm.*
2. *A composable non-interactive zero-knowledge proof system for proving equality of committed values, i.e., a proof of relation* $R = \{(params, (x, y), (open_x, open_y)) \mid C = \mathsf{Com}(params, x, open_x) \wedge D = \mathsf{Com}(params, y, open_y) \wedge x = y\}$.
3. *A secure two party computation [JS07] that lets a signer issue a signature on a committed message vector* $\boldsymbol{m}$ *without learning any information about* $\boldsymbol{m}$. *The protocol consists of interactive algorithms* $\mathsf{SigIssue}(params, sk, C_1, \dots C_n)$ *and* $\mathsf{SigObtain}(params, pk, \boldsymbol{m}, open_1, \dots, open_n)$.

3.2 Construction of a Multi-Block P-Signature Scheme

We first construct an F-secure multi-block signature scheme.

$\mathsf{Setup}(1^k)$. Let $(p, G_1, G_2, G_T, e, g, h) \leftarrow \mathsf{BMGen}(1^k)$ be the parameters of a bilinear map, let u be an additional generator for G_1, and let $params_{GS}$ be the parameters for the corresponding Groth-Sahai NIZK proof system (either in the XDH or the DLIN setup). Output parameters $params = ((q, G_1, G_2, G_T, g, h), u, params_{GS}, z = e(g, h))$.

$\mathsf{Keygen}(params)$ picks random $\alpha, \beta_1, \dots, \beta_n \leftarrow Z_p$. The signer calculates $v = h^\alpha$, $\tilde{v} = g^\alpha$, $w_i = h^{\beta_i}$, $\tilde{w_i} = g^{\beta_i}$, $1 \leq i \leq n$. The secret-key is $sk = (\alpha, \boldsymbol{\beta})$. The public-key is $pk = (v, \boldsymbol{w}, \tilde{v}, \tilde{\boldsymbol{w}})$. The public key can be verified by checking that $e(g, v) = e(\tilde{v}, h)$ and $e(g, w_i) = e(\tilde{w}_i, h)$ for all i.

$\mathsf{Sign}(params, (\alpha, \beta), \boldsymbol{m})$ chooses a random $r \leftarrow Z_p \setminus \{-(\alpha + \beta_1 m_1 + \dots + \beta_n m_n)\}$ and calculates $\sigma_1 = g^{1/(\alpha + r + \beta_1 m_1 + \dots + \beta_n m_n)}$, $\sigma_2 = h^r$, $\sigma_3 = u^r$. The signature is $(\sigma_1, \sigma_2, \sigma_3)$.

$\mathsf{VerifySig}(params, (v, \boldsymbol{w}, \tilde{v}, \tilde{\boldsymbol{w}}), \boldsymbol{m}, (\sigma_1, \sigma_2, \sigma_3))$ outputs accept if $e(\sigma_1, v\sigma_2 \prod_{i=1}^n w_i^{m_i}) = z$ and $e(u, \sigma_2) = e(\sigma_3, h)$.

Theorem 1. *Let $F(m) = (h^m, u^m)$. The above signature scheme is F-secure given the HSDH and TDH assumptions.* See Appendix **??** for the proof.

We need to augment the multi-block signature scheme with the three P-Signature protocols.

1. $\mathsf{SigProve}(params, (v, \boldsymbol{w}, \tilde{v}, \tilde{\boldsymbol{w}}), (\sigma_1, \sigma_2, \sigma_3), \boldsymbol{m})$ is defined as follows: We use Com to commit to the m_i as follows: $\mathsf{Com}(m_i, (open_{i,1}, open_{i,2})) = (\mathsf{GSCom}($

$h^{m_i}, open_{i,1}), \mathsf{GSCom}(u^{m_i}, open_{i,2})) = (H_i, U_i) = C_i$; then we form the Groth-Sahai proof:

$$\begin{aligned}\pi \leftarrow \mathsf{NIZK}[&h^{m_1} \text{ in } H_i, u^{m_1} \text{ in } U_1, \dots, h^{m_n} \text{ in } H_n, u^{m_n} \text{ in } U_n]\{\\ &(h^{m_1}, u^{m_1}, w_1^{m_1}, \dots, h^{m_n}, u^{m_n}, w_n^{m_n}, \sigma_1, \sigma_2, \sigma_3) :\\ &e(\sigma_1, v\sigma_2 \textstyle\prod_{i=1}^n w_i^{m_i}) = z \wedge\\ &e(u, \sigma_2)e(\sigma_3, h^{-1}) = 1 \wedge \{e(\tilde{w}_i, h^{m_i})e(g^{-1}, w_i^{m_i}) = 1 \wedge\\ &e(u, h^{m_i})e(u^{m_i}, h^{-1}) = 1\}_{i=1}^n\}\end{aligned}$$

$\mathsf{VerifyProof}(params, pk, \pi, (C_1, \dots, C_n))$ simply verifies the proof π.
To see that the witness indistinguishable proof π is also zero-knowledge, the simulation setup sets $u = g^a$. The simulator can then pick $s, m_1, \dots m_n \leftarrow Z_p$ and compute $\sigma_1 = g^{1/s}$. We implicitly set $r = s - (\alpha + \sum_{i=1}^n m_i\beta_i)$. Note that the simulator does not know r and α. However, he can compute $h^r = h^s/(v \prod_{i=1}^n w_i^{m_i})$ and $u^r = u^s/(\tilde{v} \prod_{i=1}^n \tilde{w_i}^{m_i})^a$. Now he can use h^{m_1}, u^{m_1}, $w_1^{m_1}$,..., h^{m_n}, u^{m_n}, $w_n^{m_n}$, σ_1, $\sigma_2 = h^r$, $\sigma_3 = u^r$ as a witness and construct the proof π in the same way as the real Prove protocol. By the witness indistinguishability, a proof using the faked witnesses is indistinguishable from a proof using a real witness. See also [BCKL08].

2. The second protocol is a proof of equality of committed values. It is of the form $\mathsf{NIPK}[x \text{ in } C; y \text{ in } D]\{(x, y, h^\theta) : e(x/y, h^\theta) = 1 \wedge e(g, h^\theta) = e(g, h)\}$. Groth and Sahai [GS07] show that such witness-indistinguishable proofs are also zero-knowledge. A simulator that knows the simulation trapdoor *sim* for the GS proof system can simulate the two conditions by setting θ to 0 and 1 respectively. In this way he can fake the proofs for arbitrary commitments.
3. The third protocol is a secure two-party computation for signing a committed value. One could use the same technique as in Belenkiy et al. [BCKL08] to reduce computing a signature to computing an arithmetic circuit using the Jarecki and Shmatikov [JS07] secure two-party computation protocol. Alternatively, we suggest the use of a more efficient protocol based on homomorphic encryption as for example done in [BCC$^+$09,CKW04].

Theorem 2. *The above construction is a secure P-Signature scheme given the HSDH and TDH assumption, either the SXDH or DLIN assumption, and the security of the two-party computation protocol.*

The proof follows from the F-unforgeability of the multi-block signature scheme and the security of the Groth-Sahai proofs, which depend on either the SXDH or DLIN assumptions. The zero-knowledge simulations are done as sketched above. For details we refer to [GS07,BCKL08,BCC$^+$09].

4 Strongly Simulatable Verifiable Random Functions

Here we present our new construction for sVRFs. Later, we will show that an extension of this construction (as described in sections 4.2 and 4.3) can be used to construct provably secure e-cash.

At a high level, a sVRF is an extension of a pseudorandom function (PRF) (and also of a slightly weaker extension, called a VRF [MRV99]). It includes a key generation procedure that generates a seed for the PRF along with a corresponding public key. It also includes a proof system for proving that a particular output is correct with respect to a given input and a given public key. We require fairly strong hiding properties from this proof system – in particular, we do not want it to interfere with the pseudorandomness properties of the PRF. For the full definition, see [CL07].

4.1 A New sVRF Construction

Our construction will be in the bilinear group setting where $(p, G_1, G_2, G_T, e, g, h) \leftarrow \mathsf{BMGen}(1^k)$. We will use the function $F_s(x) = g^{\frac{1}{s+x}}$ to build an efficient Simulatable VRF.[6] Note that the base function is similar to the Dodis-Yampolskiy VRF [DY05], which uses the function $F_s(x) = e(g,h)^{\frac{1}{s+x}}$ and thus gives output in G_T. Moving our function to output elements in G_1 is the crucial step which allows us to use the Groth-Sahai proof techniques.

Theorem 3. *Let $D_k \subset Z$ denote a family of domains of size polynomial in k. Let p, g, e, G_1, G_2, G_T be as described above where $|p| = k$. If the DDHI assumption holds in G_1, then the set $\{g^{\frac{1}{s+x}}\}_{x \in D_k}$ is indistinguishable from the set $\{g^{r_x}\}_{x \in D_k}$ where $s, \{r_x\}_{x \in D_k}$ are chosen at random from Z_p.* The proof is very similar to that in [DY05].

We will build an sVRF based on this function as follows:

$\mathsf{Setup}(1^k)$. Let $(p, G_1, G_2, G_T, e, g, h) \leftarrow \mathsf{BMGen}(1^k)$ be the parameters of a bilinear map and let $params_{GS}$ be the parameters for the corresponding Groth-Sahai NIZK proof system (either in the XDH or the DLIN setup). Output parameters $params_{VRF} = ((p, G_1, G_2, G_T, g, h), params_{GS})$.

$\mathsf{Keygen}(params_{VRF})$. Pick a random seed $s \leftarrow Z_p$ and random opening information $open_s$, and output secret key $sk = (s, open_s)$ and public key $pk = \mathsf{GSCom}(h^s, open_s)$.

$\mathsf{Eval}(params_{VRF}, sk = (s, open_s), x)$. Compute $y = g^{1/(s+x)}$.

$\mathsf{Prove}(params_{VRF}, sk = (s, open_s), x)$. Compute $y = g^{1/(s+x)}$ and $C_y = \mathsf{GSCom}(y, open_y)$ from random opening $open_y$. Next create the following two proofs: π_1, a composable NIZK proof that C_y is a commitment to y; this is proof that the value v committed to in C_y fulfills the pairing product equation $e(v/y, h^\theta) = 1 \wedge e(g, h^\theta) = e(g, h)$ (see [GS07] for details); π_2, a GS composable witness indistinguishable proof that C_y is a commitment to Y and pk is a commitment to S such that $e(Y, Sh^x) = e(g, h)$. Output $\pi = (C, \pi_1, \pi_2)$.

$\mathsf{Verify}(params, pk, x, y, \pi = (C, \pi_1, \pi_2))$. Use the Groth-Sahai verification to Verify π_1, π_2 with respect to C, x, pk, y.

Theorem 4. *This construction with domain size p is a strong sVRF under the q-DDHI for G_1 and under the assumption that the Groth-Sahai proof system is secure.* For proof, consult the full version of the paper.

[6] This function is also known as a Weak Boneh-Boyen signature [BB04b].

4.2 A NIZK Protocol for Pseudo-random Functions

In some applications, we need something stronger than an sVRF. In our e-cash application, we need to be certain that the proofs will reveal no information about which wallet was used, which means that they should completely hide the seed used. Furthermore, we do not want to reveal which coin in the wallet is being spent, thus we also want to hide the input x.

Thus, we will build a composable NIZK proof for the following language:

$$\mathcal{L}_S = \{C_s, C_x, y | \exists x, s, open_x, open_s \text{ such that } C_s = \mathsf{Com}(s, open_s) \wedge C_x = \mathsf{Com}(x, open_x) \wedge y = F_s(x)\}$$

Note that there are four points where an sVRF proof is weaker than a full NIZK proof. First, the sVRF public key is not guaranteed to hide the secret key, only to hide enough information to preserve the pseudorandomness of the output values. However, this is not a problem in the above construction, since our public key is formed as a commitment. Second, an sVRF has a fixed public key, while we want to be able to compute unlinkable proofs for many different values of the PRF. This again is not relevant in the above construction: since we form our public key using a commitment scheme, we can easily use a different value in each proof. Third, in the sVRF proof, the input x is given in the clear. We can fix this fairly easily by replacing x by a commitment and proof. The final difference is that the sVRF proof need not be fully zero knowledge - the sVRF simulator is given the secret key as input (in our construction, the opening of the commitment C_s). We resolve this last point by adding extra commitments C'_s, C'_x (whose opening the zero-knowledge simulator will know), and zero-knowledge proofs that they commit to the same values as C_s, C_x.

On input (C_s, C_x, y) and $(x, s, open_x, open_s)$ a NIZK proof of membership in $\mathcal{L}_S$ is done as follows: We first compute commitment C'_s to h^s. Then we compute C_y, π_1 as in the sVRF Prove protocol, with $pk = C'_s$. Next we compute a commitment C'_x to h^x, and a GS composable witness-indistinguishable proof π_2 that C_y is a commitment to Y, C'_x is a commitment to X, and C'_s is a commitment to S such that $e(Y, SX) = e(g, h)$. Finally, to make the construction zero-knowledge, we add composable NIZK proofs π_s and π_x that C_s and C'_s, and C_x and C'_x are commitments to the same values. Let v be s or x, respectively. Then each proof is a proof that the values v and v' committed to in C_v and C_v fulfill the pairing product equation $e(v/v', h^\theta) = 1 \wedge e(g, h^\theta) = e(g, h)$. See [GS07] for why this is zero-knowledge. The final proof is $\pi = (C'_s, C'_x, C'_y, \pi_1, \pi_2, \pi_s, \pi_x)$.

The proof is verified using the Groth-Sahai verification techniques to check $\pi_1, \pi_2, \pi_3, \pi_4$ with respect to $C_s, C_x, y, C'_s, C'_x, C'_y$.

Theorem 5. *The above proof system is a secure composable zero knowledge proof system for the language $\mathcal{L}_S(params)$, where params is output by* Setup. For proof appears in the full version.

4.3 NIZK Proofs Doublespending Equations: A More Complex Language

In our application, we use NIZKs about PRFs in two different places. The first is to prove that a given serial number has been computed correctly as $F_s(x)$ according to a committed seed s and committed input x. That can be done using the NIZK protocol described in the previous section. However, we also need to be able to prove that the doublespending value T has been computed correctly. Thus, we also need a proof system for the following language:

$$\begin{aligned}\mathcal{L}_T = \{C_s, C_x, C_{sk}, tag, ch \mid \exists x, s, sk, open_x, open_s, open_{sk} \text{ such that}\\ C_s = \mathsf{Com}(s, open_s) \wedge C_x = \mathsf{Com}(x, open_x)\\ \wedge\, C_{sk} = \mathsf{Com}(sk, open_{sk}) \wedge tag = (g^{sk})^{ch} F_s(x)\}\end{aligned}$$

We can generalize our above proof system to handle this as well. For the construction see the full version.

4.4 Efficiency comparison with previous sVRF construction

As described above, our sVRF proof requires 1 commitment in G_1, 1 Groth-Sahai proof, and one zero-knowledge proof of equality of values in G_1. Thus, if we instantiate the proofs under the SXDH assumption, our construction requires 14 elements of G_1 and 14 elements of G_2 to give a proof, and the sVRF outputs a random element of the group G_1. Note that the group size is exponential in the security parameter k, so this really produces k bits of pseudorandomness.

We compare this to the previous contruction of sVRFs given by Chase and Lysyanskaya [CL07]. That construction was based on composite order bilinear groups. For the order of such groups to resist factorization they must be of a much greater size to achieve the same security as prime order groups. We assume a conservative factor of 5 for this difference [7]. As pairing operations (and exponentiation) have cubic complexity, it is fair to assume that composite order pairings are at least two orders of magnitude slower than prime order pairings.

In addition, the basic construction of [CL07] is only weakly simulatable: for each input value there was a certain restricted set of outputs for which the simulator could output a simulated proof. Finally, the simulator also required some trapdoor information about the desired output value (in the construction it was a discrete logarithm). In order to obtain full simulatability, in which the simulator could produce a simulated proof for any output value in the range of the function with no additional information, this result applied an extractor to the output of the weak sVRF to extract a single bit. The simulator could then sample values from the simulatable range together with some trapdoor information, until it had found one on which the extractor produced the appropriate bit. Clearly extending this approach to achieve more than $O(\log k)$ bits of randomness would be infeasible.

[7] http://www.keylength.com/en/3/

Each proof generated by this construction requires 3 elements of the composite order group G. Thus, in order to produce k bits of randomness, even if we assume that we extended the construction to extract $\log k$ bits, we would need $k/\log k$ proofs, for a total of $3 * k/\log k$ elements of G.

5 New Compact E-Cash Scheme

We construct a compact e-cash scheme using our multi-block P-signatures and sVRF protocols. Compact e-cash as defined by Camenisch et al. [CHL05] lets a user withdraw multiple e-coins simultaneously. There are three types of players: a bank $\mathcal{B}$ as well as many users $\mathcal{U}$ and merchants $\mathcal{M}$ (though merchants are treated as a special type of user). Please refer to [CHL05] for protocol specifications and a definition of security.[8] We now show how to construct compact e-cash.

$\mathsf{CashSetup}(1^k)$. The setup runs $\mathsf{SigSetup}(1^k)$ and returns the P-signature parameters $params$. Our construction is non-blackbox: we reuse the GS NIPK proof system parameters $params_{GS}$ that are contained in $params$. The parameters $params_{GS}$ in turn contain the setup for a bilinear pairing $params_{BM} = (p, G_1, G_2, G_T, e, g, h)$ for a paring $e : G_1 \times G_2 \to G_T$ for groups of prime order p.

$\mathsf{BankKG}(params, n)$. The bank creates two P-signature key pairs, $(pk_w, sk_w) \leftarrow \mathsf{SigKeygen}(params)$ for issuing wallets and $(pk_c, sk_c) \leftarrow \mathsf{SigKeygen}(params)$ for signing coin indices. Then the bank computes a P-signature on the n coin indices $\Sigma_1, \ldots, \Sigma_n$, where $\Sigma_i = \mathsf{SigSign}(sk_c, i)$.[9] The bank's secret-key is $sk_{\mathcal{B}} = (sk_w, sk_c)$ and the bank's public-key is $(pk_w, pk_c, \Sigma_1, \ldots, \Sigma_n)$.

$\mathsf{UserKG}(params)$. The user picks $sk_{\mathcal{U}} \leftarrow Z_p^*$ and returns $(pk_{\mathcal{U}} = e(g, h)^{sk_{\mathcal{U}}}, sk_{\mathcal{U}})$. Merchants generate their keys in the same way but also have a publicly known identifier $id_{\mathcal{M}} = f(pk_{\mathcal{M}})$ associated with their public keys (f is some publicly known mapping).

$\mathsf{Withdraw}(\mathcal{U}(params, pk_{\mathcal{B}}, sk_{\mathcal{U}}, n), \mathcal{B}(params, pk_{\mathcal{U}}, sk_{\mathcal{B}}, n))$. The user withdraws a wallet of coins from the bank.

1. The user picks $s', t' \leftarrow Z_p$; computes commitments $comm_{sk} = \mathsf{Com}(sk_{\mathcal{U}}, open_{sk_{\mathcal{U}}})$, $comm_{s'} = \mathsf{Com}(s', open_{s'})$, and $comm_{t'} = \mathsf{Com}(t', open_{t'})$; and sends $comm_{sk}$, $comm_{s'}$, and $comm_{t'}$ to the bank. The user proves

[8] The original [CHL05] definition had an interactive Spend protocol, while we break it up into two non-interactive protocols: $\mathsf{SpendCoin}(params, W, pk_{\mathcal{M}}, info)$ and $\mathsf{VerifyCoin}(params, pk_{\mathcal{M}}, pk_{\mathcal{B}}, coin)$. The merchant sends the user a $info$, the user runs $\mathsf{SpendCoin}$ and gives the resulting e-coin for the merchant to verify using $\mathsf{VerifyCoin}$. We prefer to use a non-interactive spend protocol because often two-way communication is not available or impractical, e.g. when sending an e-coin by email.

[9] This will allow us to use the range proof approach from [TS06] and [CCS08], where a user proves that a value (the coin index) is in a list (the list $\{1, \ldots, N\}$) by proving knowledge of a signature on that value.

in zero-knowledge that he knows the opening to these values, and that comm_{sk} corresponds to the secret key used for computing $pk_{\mathcal{U}}$.[10]

2. If the proofs verify, the bank sends the user random values $s'', t'' \in Z_p$.
3. The user picks random $\mathit{open}_s, \mathit{open}_t$, commits to $\mathit{comm}_s = \mathsf{Com}(s' + s'', \mathit{open}_s)$, and $\mathit{comm}_t = \mathsf{Com}(t' + t'', \mathit{open}_t)$, sends comm_s and comm_t to the bank, and proves that they are formed correctly. Let $s = s' + s''$ and $t = t' + t''$.
4. The user and bank run $\mathsf{SigObtain}(\mathit{params}, pk_w, (sk_{\mathcal{U}}, s, t), (\mathit{open}_{sk}, \mathit{open}_s, \mathit{open}_t)) \leftrightarrow \mathsf{SigIssue}(\mathit{params}, sk_w, (\mathit{comm}_{sk}, \mathit{comm}_s, \mathit{comm}_t))$ respectively. The user obtains a P-signature σ on $(sk_{\mathcal{U}}, s, t)$. The user stores the wallet $W = (s, t, pk_{\mathcal{B}}, \sigma, n)$; the bank stores tracing information $T_W = pk_{\mathcal{U}}$.

$\mathsf{SpendCoin}(\mathit{params}, (s, t, pk_{\mathcal{B}}, \sigma, J), pk_{\mathcal{M}}, \mathit{info})$. The user calculates a serial number $S = F_s(J) = g^{1/(s+J)}$. The user needs to prove that he knows a signature σ on $(sk_{\mathcal{U}}, s, t)$ and a signature Σ_J on J such that $S = F_s(J)$. Next the user constructs a double-spending equation $T = (g^{id_{\mathcal{M}} \| \mathit{info}})^{sk_{\mathcal{U}}} F_t(J)$.[11] The user proves that T is correctly formed for the $sk_{\mathcal{U}}, t, J$, signed in σ and Σ_J.

All these proofs need to be done non-interactively. We now give more details. The user runs $\mathsf{SigProve}$, first on σ and pk_w to obtain commitments and proof $((C_{id}, C_s, C_t), \pi_1) \leftarrow \mathsf{SigProve}(\mathit{params}, pk_w, \sigma, (sk_{\mathcal{U}}, s, t))$ for $sk_{\mathcal{U}}, s, t$ respectively and second on Σ_J and pk_c to obtain commitment and proof $(C_J, \pi_2) \leftarrow \mathsf{SigProve}(\mathit{params}, pk_c, \Sigma_J, J)$ for J.
Then the user constructs non-interactive zero-knowledge proofs that indeed $(S, T, C_{id}, C_s, C_t, C_J, id_{\mathcal{M}} \| \mathit{info})$ are well formed. This is done by computing two proofs π_F and π_T: π_F proves that $(C_s, C_J, S) \in \mathcal{L}_S$ and is computed as described in Section 4.2, where $\mathcal{L}_S$ is defined as:

$$\begin{aligned}\mathcal{L}_S = \{C_s, C_x, y | \exists x, s, \mathit{open}_x, \mathit{open}_s \text{ such that}\\ C_s = \mathsf{Com}(s, \mathit{open}_s) \wedge C_x = \mathsf{Com}(x, \mathit{open}_x) \wedge y = F_s(x)\} \ ;\end{aligned}$$

π_T proves that $(C_t, C_J, C_{id}, T, (id_{\mathcal{M}} | \mathit{info})) \in \mathcal{L}_T$ and is computed as described in Section 4.3, where $\mathcal{L}_T$ is defined as:

$$\begin{aligned}\mathcal{L}_T = \{C_s, C_x, C_{sk}, \mathit{tag}, \mathit{ch} \mid \exists x, s, sk, \mathit{open}_x, \mathit{open}_s, \mathit{open}_{sk} \text{ such that}\\ C_s = \mathsf{Com}(s, \mathit{open}_s) \wedge C_x = \mathsf{Com}(x, \mathit{open}_x) \wedge\\ C_{sk} = \mathsf{Commit}(sk, \mathit{open}_{xsk}) \wedge \mathit{tag} = (g^{sk})^{\mathit{ch}} F_s(x)\} \ .\end{aligned}$$

The user outputs a $\mathit{coin} = (S, T, C_{id}, C_s, C_t, C_J, \pi_1, \pi_2, \pi_S, \pi_T, id_{\mathcal{M}} \| \mathit{info})$.

[10] These and the rest of the proofs in the issue protocol can be done using efficient sigma protocols [CS97b,Dam02] and their zero-knowledge compilers [Dam00].

[11] The merchant is responsible for assuring that *info* is locally unique. Coins which have the same serial number and the same $id_{\mathcal{M}} \| \mathit{info}$ cannot be deposited and the damage lies with the merchant. The dangers that users get cheated by verifiers that do not accept coins with correct *info* can be mitigated using techniques such as endorsed e-cash [CLM07].

$\mathsf{VerifyCoin}(params, pk_{\mathcal{M}}, pk_{\mathcal{B}}, coin)$. To verify parses $coin$ as $(S, (T, C_{id}, C_s, C_t, C_J, \pi_1, \pi_2, \pi_S, \pi_T), id_{\mathcal{M}}' \| info)$ and checks that the following checks succeed: (1) Check that $id_{\mathcal{M}}' = f(pk_{\mathcal{M}})$. (2) $\mathsf{SigVerify}(params, pk_w, \pi_1, (C_{id}, C_s, C_t)) = \mathsf{accept}$. (3) $\mathsf{SigVerify}(params, pk_c, \pi_2, C_J) = \mathsf{accept}$. (4) $\mathsf{Verify}_{\mathcal{L}_S}(params_{GS}, (C_s, C_J, S), \pi_S) = \mathsf{accept}$. (5) $\mathsf{Verify}_{\mathcal{L}_T}(params_{GS}, (C_t, C_J, C_{id}, T, (id_{\mathcal{M}} \| info)), \pi_T) = \mathsf{accept}$.

Note that the merchant is responsible for assuring that $info$ is unique over all of his transactions. Otherwise his deposit might get rejected by the following algorithm.

$\mathsf{Deposit}(params, pk_{\mathcal{B}}, pk_{\mathcal{M}}, coin, state_{\mathcal{B}})$. The algorithm parses the coin as $coin = (S, T, C_{id}, C_s, C_t, C_J, \pi_1, \pi_2, \pi_S, \pi_T, id_{\mathcal{M}} \| info)$ and performs the same checks as $\mathsf{VerifyCoin}$. The bank maintains a database $state_{\mathcal{B}}$ of all previously accepted coins. The output of the algorithm is an updated database $state'_{\mathcal{B}} = state_{\mathcal{B}} \cup \{coin\}$ and the flag $result$, that is computed as follows:

(i) If the coin verifies and if no coin with serial number S is stored in $state_{\mathcal{B}}$, $result = \mathsf{accept}$ to indicate that the coin is correct and fresh. The bank deposits the value of the e-coin into the merchant's account and adds $coin$ to $state_{\mathcal{B}}$.

(ii) If the coin doesn't verify or if there is a coin with the same serial number and the same $id_{\mathcal{M}} \| info$ already stored in $state_{\mathcal{B}}$, $result = \mathsf{merchant}$ to indicate that the merchant cheated. The bank refuses to accept the e-coin because the merchant failed to properly verify it.

(iii) If the coin verifies but there is a coin with the same serial number S but different $id_{\mathcal{M}} \| info$ in $state_{\mathcal{B}}$, $result = \mathsf{user}$ to indicate that a user doublespent. The bank pays the merchant (who accepted the e-coin in good faith) and punishes the double-spending user.

$\mathsf{Identify}(params, pk_{\mathcal{B}}, coin_1, coin_2)$ allows the bank to identify a double-spender. Parse $coin_1 = (S, (T, C_{id}, C_s, C_t, C_J, \pi_1, \pi_2, \pi_S, \pi_T), id_{\mathcal{M}1} \| info_1)$ and $coin_2 = (S', (T', C'_{id}, C'_s, C'_t, C'_J, \pi'_1,$
$\pi'_2, \pi'_S, \pi'_T), id_{\mathcal{M}2} \| info_2)$.

The algorithm aborts if one of the coins doesn't verify, if $S \neq S'$, or if $id_{\mathcal{M}1} \| info_1 = id_{\mathcal{M}2} \| info_2$. Otherwise, the algorithm outputs $T_W = pk_{\mathcal{U}} = e((T/T')^{1/(id_{\mathcal{M}1} \| info_1 - id_{\mathcal{M}2} \| info_2)}, h)$, which the bank compares to the trace information it stores after each withdrawal transaction.

Theorem 6. *This e-cash scheme is a secure compact e-cash scheme given the security of the P-signature scheme, the PRF, and the Groth-Sahai NIZK proof system.*

In the full version we provide a proof and a performance analysis of our scheme.

Acknowledgements. Markulf Kohlweiss was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT and IST programmes under the following contracts: ICT-2007-216483 PRIMELIFE, ICT-2007-216676 ECRYPT II, and IST-015964 AEOLUS. The information in this document reflects only the authors' views.

References

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective id secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT '04*, volume 3027 of *LNCS*. Springer-Verlag, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT '04*, volume 3027 of *LNCS*, pages 54–73. Springer-Verlag, 2004.

[BCC$^+$09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Delegatable anonymous credentials. In Shai Halevi, editor, *CRYPTO '09*. Springer 2009.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC '08*, volume 4948 of *LNCS*, pages 356–374. Springer-Verlag, 2008.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC '88*, pages 103–112, Chicago, Illinois, 2–4 May 1988.

[BG90] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge. In Gilles Brassard, editor, *CRYPTO '89*, volume 435 of *LNCS*, pages 194–211. Springer-Verlag, 1990.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC '05*, volume 3378 of *LNCS*, pages 325–341. Springer-Verlag, 2005.

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *EUROCRYPT '00*, volume 1807 of *LNCS*, pages 431–444. Springer Verlag, 2000.

[Bra93] Stefan Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI, April 1993.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC '07*, pages 1–15, 2007.

[CCS08] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient protocols for set membership and range proofs. In *ASIACRYPT '08*, volume 5350 of *LNCS*, pages 234–252. Springer-Verlag, 2008.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *CRYPTO '88*, volume 403 of *LNCS*, pages 319–327. Springer Verlag, 1990.

[Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO '82*, pages 199–203. Plenum Press, 1983.

[CHK$^+$06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *CCS '06*, pages 201–210, New York, NY, USA, 2006. ACM Press.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact E-cash. In Ronald Cramer, editor, *EUROCRYPT '05*, volume 3494 of *LNCS*, pages 302–321. Springer-Verlag, 2005.

[CKW04] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In Carlo Blundo and Stelvio Cimato, editors, *SCN '04*, volume 3352 of *LNCS*, pages 134–148. Springer-Verlag, 2004.

[CL07] Melissa Chase and Anna Lysyanskaya. Simulatable vrfs with applications to multi-theorem nizk. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 303–322. Springer-Verlag, 2007.

[CLM07] Jan Camenisch, Anna Lysyanskaya, and Mira Meyerovich. Endorsed e-cash. In *IEEE Symposium on Security and Privacy*, pages 101–115, 2007.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71*, pages 151–158, New York, NY, USA, 1971. ACM.

[CP93] David Chaum and Torben Pryds Pedersen. Transferred cash grows in size. In Rainer A. Rueppel, editor, *EUROCRYPT '92*, volume 658 of *LNCS*, pages 390–407. Springer-Verlag, 1993.

[CS97a] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In Burt Kaliski, editor, *CRYPTO '97*, volume 1296 of *LNCS*, pages 410–424. Springer Verlag, 1997.

[CS97b] Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical Report TR 260, Institute for Theoretical Computer Science, ETH Zürich, March 1997.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO '98*, volume 1642 of *LNCS*, pages 13–25. Springer Verlag, 1998.

[Dam00] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In Bart Preneel, editor, *EUROCRYPT '00*, volume 1807 of *LNCS*, pages 431–444. Springer Verlag, 2000.

[Dam02] Ivan Damgård. On Σ-protocols. Available at `http://www.daimi.au.dk/~ivan/Sigma.ps`, 2002.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC '91*, pages 542–552, 1991.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *PKC '05*, volume 3386 of *LNCS*, pages 416–432, 2005.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer Verlag, 1987.

[FST06] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. Cryptology ePrint Archive, Report 2006/372, 2006. `http://eprint.iacr.org/`.

[FTY96] Yair Frankel, Yiannis Tsiounis, and Moti Yung. "Indirect discourse proofs:" Achieving efficient fair off-line E-cash. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT '96*, volume 1163 of *LNCS*, pages 286–300. Springer Verlag, 1996.

[FY92] Matthew Franklin and Moti Yung. Towards provably secure efficient electronic cash. Technical Report TR CUSC-018-92, Columbia University, Dept. of Computer Science, April 1992. Also in: Proceedings of ICALP 93, Lund, Sweden, July 1993, volume 700 of *LNCS*, Springer Verlag.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS '03*, pages 102–115. IEEE Computer Society Press, 2003.

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent. In Ernest F. Brickell, editor, *CRYPTO '92*, volume 740 of *LNCS*, pages 228–244. Springer-Verlag, 1992.

[GS07] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. http://eprint.iacr.org/2007/155, 2007.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *CRYPTO '97*, volume 1294 of *LNCS*, pages 150–164. Springer-Verlag, 1997.

[JS07] Stanislaw Jarecki and Vitaly Shmatikov. Efficient two-party secure computation on committed inputs. In *EUROCRYPT '07*, volume 4515 of *LNCS*, pages 97–114. Springer-Verlag, 2007.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *FOCS '99*, pages 120–130. IEEE Computer Society Press, 1999.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO '91*, volume 576 of *LNCS*, pages 433–444. Springer-Verlag, 1992.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS '99*, pages 543–553. IEEE Computer Society Press, 1999.

[SPC95] Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. Fair blind signatures. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *EUROCRYPT '95*, volume 921 of *LNCS*, pages 209–219. Springer Verlag, 1995.

[STS99] Tomas Sander and Amnon Ta-Shma. Auditable, anonymous electronic cash extended abstract. In *CRYPTO '99* , volume 1666 of *LNCS*, pages 555–572. Springer-Verlag, 1999.

[Tro05] Mårten Trolin. A universally composable scheme for electronic cash. In Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan, editors, *INDOCRYPT '05*, volume 3797 of *LNCS*, pages 347–360. Springer-Verlag, 2005.

[TS06] Isamu Teranishi and Kazue Sako. *k*-times anonymous authentication with a constant proving cost. In *PKC '06*, volume 3958 of *LNCS*, pages 525–542. Springer-Verlag, 2006.

[Tsi97] Yiannis S. Tsiounis. *Efficient Electonic Cash: New Notions and Techniques.* PhD thesis, Northeastern University, Boston, Massachusetts, 1997.