New DLOG-Based Convertible Undeniable Signature Schemes in the Standard Model

Le Trieu Phong, Kaoru Kurosawa, Wakaha Ogata

August 11, 2009

Abstract

We propose discrete-logarithm-based undeniable signature schemes supporting both selective and all conversion, with formal security analysis in the standard model. The schemes are *the first* practical ones of their type, enjoying relatively short signatures and efficient confirmation/disavowal protocols, while for security relying on the strong Diffie-Hellman assumption and the decision linear assumption.

Keywords: Undeniable signature, selective/all conversion, discrete logarithm, standard model.

1 Introduction

1.1 Background

Almost twenty years ago, Chaum and van Antwerpen [10] introduced the concept of undeniable signature (US) scheme, where a signature is not publicly verifiable, which is in contrast to ordinary signature schemes. The verification of an undeniable signature requires the cooperation of the signer through the zero-knowledge confirmation protocol (for validity of signatures) and zero-knowledge disavowal protocol (for invalidity of signatures). A mandatory property of a US scheme thus is *invisibility*, namely without interacting with the signer, it is hard to decide whether a signature is valid or not. Also, it is worth noting that either the confirmation or disavowal protocol must be successful if the signer is honest; and the case both protocols fail formally implies that the signer is not cooperating (or cheating).

Undeniable signature is useful when we sign on sensitive data such as software [10], electronic cash [5,11,26], confidential business agreement [12]. There have been a wide range of research on the concept [4, 9, 12, 13, 17–24, 28], to list just a few. Most of the papers are in the random oracle model, with (even arbitrary) short signatures [23], or extensive security consideration of a classical scheme [24]. In the standard model, the first efficient proposal is that of Laguillaumie and Vergnaud [21] (but relying on a non-standard and strong assumption for invisibility).

In order to link undeniable signature to regular signature, Boyar et al [4] proposed the concept of conversion. In *all conversion*, the signer releases a piece of information so that all issued undeniable signatures can be publicly-verifiable. In *selective conversion*, the signer publishes a piece of information so that a single undeniable signature

is publicly-verifiable. The paper [4] gave a generic construction of US scheme with selective and all conversion from one-way function, but the construction is not practical. Note that selectively-convertible undeniable signature schemes play a central role in fair payment protocols [5], so the more efficient the former is, the more practical the latter can be realized. For more applications, the readers may find in [4, 12]. We also note that the above mentioned work of Laguillaumie and Vergnaud [21], while producing very short signatures (of about 170 bits), does not support any kinds of conversion.

In an attempt to realize practical US schemes supporting conversions, Damgard and Pedersen [12] proposed two dlog-based schemes, but they could not formally prove the invisibility of their schemes, and just conjectured on it. Recently, another attempt was made by Yuen et al [28] using pairings, but their scheme suffers from a big (exponential) loss factor in security reduction, so that the signer is only able to produce very few (less than 128) signatures. The scheme in [28] is claimed to satisfy invisibility, but in Appendix A, we point out that the claim is incorrect, and the scheme is totally broken.

Based on the above literature, we therefore say that, in the discrete-log-based setting, there is still no practical undeniable signature scheme supporting (selective, all) conversions, with formally-proved security in the standard model. We remark that in the RSA-based setting, Kurosawa and Takagi [20] proposed some efficient schemes with selective conversion (but not all conversion).

1.2 Our contribution

We propose two undeniable signature schemes, called SCUS_1 and SCUS_2, with the following properties:

- supporting both selective and all conversion.
- formally-proven security in the standard model, relying on the strong Diffie-Hellman assumption and the decision linear assumption. Furthermore, the factor loss in security reduction is polynomial.
- Both the confirmation and disavowal protocols are of constant moves (4 moves)[1].
- The signature size is about $70 + 3 \cdot |q|$ (resp, $4 \cdot |q|$) bits for SCUS_1 (resp, SCUS_2) where $|q| \approx 170$. The piece of information for all conversion is of $2 \cdot |q|$ bits for both schemes. For each selective conversion, which employs the NIZK proof of Groth and Sahai [15] in a novel way, the signer needs to totally release $13 \cdot |q|$ bits.

Above, the scheme SCUS_1 produces shorter signatures than SCUS_2, but the public key of SCUS_1 (of $164 \cdot |q|$ bits) is much longer than that of SCUS_2 (of $6 \cdot |q|$ bits). Choosing which one to use thus depends on specific applications. Also note that our signature sizes are much shorter than those of Kurosawa-Takagi [20] (of $4 \cdot 1024$ bits in the standard model), since we are in the dlog-based setting.

Let us now look at the ways to obtain the above results. We first focus on the ideas behind SCUS_1.

[1] We remark that the 3-move scheme of Kurosawa and Heng [18] is insecure, as shown by Ogata et al in [24] (Sect.V.D, page 2013), who furthermore point out that any 3-move (HVZK) confirmation/disavowal protocols are not secure against active attacks.

Sign-then-Encrypt Paradigm. We re-utilize an elegant paradigm introduced by Damgard and Pedersen [12] in which the the undeniable signature σ of a message m is of the form $\sigma = \mathsf{Encrypt}_{pk_2}(\mathsf{Sign}_{sk_1}(m))$, where $\mathsf{Encrypt}$ and Sign are respectively some regular encryption and signature scheme. For all conversion, the signer publishes the secret key sk_2 of the encryption scheme, so that everyone can decrypt σ to get the regular signature $\mathsf{Sign}_{sk_1}(m)$ and then check its validity. For selective conversion, the signer releases the regular signature $\mathsf{Sign}_{sk_1}(m)$.

Some difficulties when using the above paradigm are: (1) designing efficient zero-knowledge confirmation and disavowal protocols, (2) proving the invisibility of the designed scheme, and (3) releasing $\mathsf{Sign}_{sk_1}(m)$ in a provable way (that it is the signature encrypted in σ). Damgard and Pedersen [12] have overcome (1) but not (2). For (3), they suggested a storage-costly method of storing all randomness previously used in signing (while we manage to avoid that method by using the efficient NIZK proof of Groth and Sahai [15], as seen later).

To overcome (1) (and (3) in an efficient way), one needs to properly choose simple (but-secure-enough) ingredients. To design SCUS_1, we choose the Generic Bilinear Map (GBM) signature [16] and the linear encryption [2] (LE) scheme. A GBM signature on m is of the form $(s, \rho = H(m)^{1/(x+s)})$ for a random s, a standard model hash function H and the secret key $sk_1 = x$. We use the LE scheme to encrypt ρ in the ciphertext $(u_1 = g_1^{r_1}, u_2 = g_2^{r_2}, u_3 = \rho \cdot g^{r_1+r_2})$ for randomness r_1, r_2. The undeniable signature $\sigma = (s, u_1, u_2, u_3)$.

Intuitively, σ seems random-like, unrelated to m, (and thus invisible) because s is random and (u_1, u_2, u_3) is random-like under the decision linear assumption. However, the scheme is in fact *not* invisible. The reason is in the malleability of LE scheme. In particular, if $\sigma = (s, u_1, u_2, u_3)$ is valid on a message m (resp, σ is random), then $\sigma' = (s, u_1 g_1^{\alpha}, u_2 g_2^{\beta}, u_3 g^{\alpha+\beta})$ is also valid on m (resp, σ' is random) for adversarily-chosen randomness α and β. The fact causes a simple attack on the invisibility of (m, σ) as follows: the adversary first asks the signer for converting (m, σ'), so that it knows the validity of the pair, and hence it also is aware of whether the corresponding (m, σ) is valid. (See Definition 3 for a formal definition on invisibility, which also contains some new insights.)

Fortunately, we can overcome the above attack as follows: we authenticate the randomness r_1, r_2 by signing on u_1 and u_2. In our proposed SCUS_1 scheme (in Sect.4), the values $(u_1 = g_1^{r_1}, u_2 = g_2^{r_2})$ are generated first, then the GBM signature on m, u_1, u_2 is created: $\big(s, \rho = H(m \parallel u_1 \parallel u_2)^{1/(x+s)}\big)$. After all, set $u_3 = \rho \cdot g^{r_1+r_2}$ and let the undeniable signature $\sigma = (s, u_1, u_2, u_3)$. With the authentication on the randomness, the adversarily-formed σ' above becomes invalid regardless of whether σ is valid on m, so that the validity of σ' cannot be used to decide that of σ. We succeed in proving the invisibility of our proposed scheme in Theorem 6.

On Confirmation and Disavowal Protocol. Now we give ideas on constructing the confirmation and disavowal protocol for SCUS_1. To confirm $\big(m, \sigma = (s, u_1, u_2, u_3)\big)$, the signer needs to prove for secrets $x_1 (= \log_{g_1} g), x_2 (= \log_{g_2} g)$, and x:

$$\frac{u_3}{u_1^{x_1} u_2^{x_2}} = H(m \parallel u_1 \parallel u_2)^{\frac{1}{x+s}}.$$

Namely, the LE decryption of (u_1, u_2, u_3) gives the GBM signature on m, u_1, u_2. Or equivalently,

$$u_3^x \cdot u_1^{-x_1(x+s)} \cdot u_2^{-x_2(x+s)} = H(m \parallel u_1 \parallel u_2) \cdot u_3^{-s},$$

which is a proof of representation of public value $H(m \parallel u_1 \parallel u_2) \cdot u_3^{-s}$, and can be realized by standard techniques, using constant moves.

Now we turn to the disavowal protocol. Given $\big(m, \sigma = (s, u_1, u_2, u_3)\big)$, the signer needs to prove for secrets x_1, x_2, x:

$$\frac{u_3}{u_1^{x_1} u_2^{x_2}} \neq H(m \parallel u_1 \parallel u_2)^{\frac{1}{x+s}},$$

or equivalently,

$$u_3^{x+s} \cdot u_1^{-x_1(x+s)} \cdot u_2^{-x_2(x+s)} \cdot H(m \parallel u_1 \parallel u_2)^{-1} \neq 1.$$

Employing the technique of Camenisch and Shoup [8], we choose $r \xleftarrow{\$} Z_q$ and set

$$U = \big(u_3^{x+s} \cdot u_1^{-x_1(x+s)} \cdot u_2^{-x_2(x+s)} \cdot H(m \parallel u_1 \parallel u_2)^{-1}\big)^r.$$

The signer sends U to the verifier, who checks that $U \neq 1$. Then both execute a proof of representation of U, where the signer holds the secrets r, x, x_1, x_2. The zero-knowledge protocol can also be accomplished via standard techniques, also using constant moves. Moreover, since we will work on a pairing group, the disavowal protocol can be made non-interactive, again thanks to the NIZK proof of Groth-Sahai [15], interestingly yielding a way to efficiently "convert" (namely, make publicly-verifiable) *even invalid* signatures.

More Schemes. The above ideas work well if we replace the GBM signature by the signature of Boneh and Boyen [1], which is of the form $(s, g_0^{1/(x+H(m)+ys)})$ for random $s \in Z_q$, $g_0 \in G$, and secret signing key x, y. The replacement creates our SCUS_2 described in Sect.5. Furthermore, in the random oracle model, one can use the BLS signature [3] so that the unforgeability of the resulting undeniable scheme relies on the CDH assumption in bilinear group. We do not explicitly consider the random oracle scheme in this paper.

2 Syntax and definitions

We begin with the syntax of selectively-convertible undeniable signature (SCUS for short) schemes. We focus on the syntax of schemes with selective conversion here and do not explicitly describe the syntax of all conversion since the latter is very simple in our proposals.

Definition 1 (SCUS scheme)**.** *A selectively-convertible undeniable scheme SCUS = (KeyGen, Usign, Convert, Verify, Confirm, Disavowal) consists of four algorithms and two protocols whose descriptions are as follows.*

– KeyGen$(1^\kappa) \to (pk, sk)$: This algorithm generates the pubic key pk and the secret key (signing key) sk for user.

– USign$(sk, m) \to \sigma$: Using the secret key sk, this algorithm produces a signature σ on a message m.

– *Convert*$(sk, m, \sigma) \rightarrow cvt/\perp$*: Using sk, this algorithm releases a converter cvt if the message-signature* (m, σ) *pair is valid, enabling everyone to check the validity of the pair. If the pair is invalid, the output of the algorithm is* $\perp$*.* [2]

– *Verify*$(pk, m, \sigma, cvt) \rightarrow 0/1$*: Using the converter cvt, everyone can check the validity of* (m, σ) *by this algorithm.*

– *Confirm: This is a protocol between the signer and a verifier, on common input* (pk, m, σ)*, the signer with sk proves that* (m, σ) *is a valid message-signature pair in zero-knowledge.*

– *Disavowal: This is a protocol between the signer and a verifier, on common input* (pk, m, σ)*, the signer with sk proves that* (m, σ) *is an invalid message-signature pair in zero-knowledge.*

The following definitions describe securities that SCUS schemes should meet.

Definition 2 (Unforgeability and strong unforgeability of SCUS schemes)**.** *A selectively convertible undeniable scheme SCUS is said to be existential unforgeable under adaptive chosen message attack if no p.p.t (probabilistic polynomial time) forger* $\mathcal{F}$ *has a non-negligible advantage in the following game.*

1) $\mathcal{F}$ *is given the public key pk.*

2) $\mathcal{F}$ *is permitted to issue a series of queries shown below.*

– *Signing queries:* $\mathcal{F}$ *submits a message m to the signing oracle and receives a signature* σ *on m. These queries are adaptive, namely the next query can depend on previous ones.*

– *Convert queries:* $\mathcal{F}$ *submits a message-signature pair* (m, σ) *to the convert oracle, and receives a converter cvt. These queries are also adaptive.*

– *Confirmation/disavowal queries:* $\mathcal{F}$ *submits a message-signature pair of the form* (m, σ) *to the confirmation/disavowal oracle. We will consider active attack, where the oracle first checks the validity of* (m, σ)*. If it is a valid pair, the oracle returns* 1 *and executes the confirmation protocol with* $\mathcal{F}$ *(acting as a cheating verifier). Otherwise, the oracle returns* 0 *and executes the disavowal protocol with* $\mathcal{F}$*.*

3) At the end of the game, $\mathcal{F}$ *outputs a pair* (m^*, σ^*)*.*

In the definition of unforgeability, the forger $\mathcal{F}$ *wins the game if the pair* (m^*, σ^*) *is a valid message-signature pair, and* m^* *has never been queried to the signing oracle. The advantage of* $\mathcal{F}$ *is defined to be* $\mathbf{Adv}^{forge}_{SCUS}(\mathcal{F}) = \Pr[\mathcal{F}\ wins]$*.*
In the definition of strong unforgeability, the only different point is that (m^*, σ^*) *does not coincide with any* (m, σ) *at signing queries. We denote* $\mathcal{F}$*'s advantage in this case by* $\mathbf{Adv}^{sforge}_{SCUS}(\mathcal{F}) = \Pr[\mathcal{F}\ wins]$*.*

The notion of invisibility intuitively ensures that no-one (without contacting the signer) can tell whether a message-signature pair is valid or not, and is formally given below.

[2]Note that only valid undeniable signatures can be converted, and the signer has no responsibility to convert ill-formed ones. These properties are natural, and sufficient enough for application (e.g., [5]). However, we note in our proposed schemes, the signer can even "convert" invalid signatures by making the disavowal protocol non-interactive (yet without Fiat-Shamir heuristics).

Definition 3 (Invisibility)**.** *A selectively-convertible undeniable scheme* SCUS *satisfies invisibility under adaptive chosen message attack if no p.p.t distinguisher* $\mathcal{D}$ *has a non-negligible advantage in the following game.*

1) $\mathcal{D}$ *is given the public key* pk*.*
2) $\mathcal{D}$ *is permitted to issue a series of queries: signing queries, convert queries, confirmation/disavowal queries, as in Definition 2.*
3) At some point, $\mathcal{D}$ *outputs an arbitrary message* m^**, and requests a challenge signature* σ^* *on* m^**. The challenge signature* σ^* *is generated based on a hidden bit* b*. If* $b = 0$*, then* σ^* *is generated as usual using the signing algorithm; otherwise* σ^* *is chosen randomly from the signature space of the scheme.*
4) The distinguisher $\mathcal{D}$ *additionally issues signing queries, convert queries, confirmation/disavowal queries with the only restriction that no confirmation/disavowal query and convert query* (m^*, σ^*) *are allowed.*
5) At the end, $\mathcal{D}$ *outputs a bit* b' *as the guess for* b*.*

The distinguisher wins the game if $b' = b$ *and its advantage in this game is defined as* $\mathbf{Adv}^{iv}_{SCUS}(\mathcal{D}) = |\Pr[b' = b] - 1/2|$.

Remark 1 (On the definition of invisibility)**.** *Above, there are some subtleties. First, at step 4, we do allow the distinguisher to submit convert queries of the form* (m^*, σ) *with* $\sigma \neq \sigma^*$*. We clarify this point here for later use in Appendix A.*

Second, $\mathcal{D}$ *can make signing query* m^**, even in multiple times, even before and after step 3. Intuitively, a scheme meeting the definition enables the signer to sign on the same message many times without any loss in invisibility, so that the scheme is very suitable and easy to use at least in licensing software (which is one of the main applications). This second subtlety makes our definition differ from and stronger than previous ones (say, that of [24]). A scheme meeting the (weak) definition as in [24] can be turned into another one satisfying our definition by ensuring that the signing messages are pairwise different (via randomness, the time when signing, etc).*

Definition 4 (Standard signature scheme and its security)**.** *A signature scheme* $\mathsf{S} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vrf})$ *is as follows. On input* 1^κ*, the key generation algorithm* Kg *produces the public key* pk *and the secret signing key key* sk*. On input* sk *and a message* m*, the signing algorithm* Sign *produces a signature* σ*, which is publicly-verifiable using the verification algorithm* Vrf *on input* pk *and* σ*.*

The unforgeability under chosen message attack (uf-cma security) of a signature scheme S *is defined essentially the same as that of SCUS in Definition 2, except that the forger* $\mathcal{F}$ *against* S *only issues signing queries. We denote the advantage of* $\mathcal{F}$ *by* $\mathbf{Adv}^{uf-cma}_{\mathsf{S}}(\mathcal{F}) = \Pr[\mathcal{F}\ wins]$*. The strong unforgeability (suf-cma security) is defined in a similar manner and we have the advantage* $\mathbf{Adv}^{suf-cma}_{\mathsf{S}}(\mathcal{F}) = \Pr[\mathcal{F}\ wins]$.

3 Preliminaries

Pairing Group. We call $\mathbb{PG} = (G, G_T, q = |G|, g, \hat{e} : G \times G \to G_T)$ a pairing group if G and G_T are cyclic groups of prime order q, where the bit length $|q| = \kappa \approx 170$. The element g is a generator of G, and the mapping $\hat{e}$ satisfies the following properties: $\hat{e}(g, g) \neq 1$, and $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.

Decision Linear Assumption. Given a pairing group $\mathbb{PG}$, the assumption, first formalized in [2], asserts that the following advantage of a p.p.t adversary $\mathcal{A}$ is negligible in the security parameter κ.

$$\mathbf{Adv}_G^{dlin}(\mathcal{A}) = \Big|\Pr\Big[\alpha,\beta,\gamma \xleftarrow{\$} Z_q; g_1,g_2,g_3 \xleftarrow{\$} G; T_0 \leftarrow g_3^{\alpha+\beta}; T_1 \leftarrow g_3^{\gamma}; \\ b \xleftarrow{\$} \{0,1\}; b' \xleftarrow{\$} \mathcal{A}(\mathbb{PG}, g_1,g_2,g_3,g_1^{\alpha},g_2^{\beta},T_b)\colon b'=b\Big] - \frac{1}{2}\Big|.$$

Known Dlog-Based ZKIP. We use known techniques for proving statements about discrete logarithms, such as (1) proof of knowledge of discrete logarithm [27]; (2) proof of knowledge of an element representation in a prime order group [25]; and the $\wedge$ proof of (1) and (2). (The $\wedge$ proof is easily designed by choosing the same challenge while asking the prover to prove both (1) and (2) in parallel.) These proofs need four moves to become zero-knowledge.

When referring to the proofs above, we use the following kind of notation. For instance, PoK$\{(x_1,x_2)\colon y = g^{x_1} \wedge U = u_1^{x_1}u_2^{x_2}\}$ denotes a zero-knowledge proof of knowledge of x_1 and x_2 such that $y = g^{x_1}$ and $U = u_1^{x_1}u_2^{x_2}$. All values except (x_1,x_2) are assumed to be known to the verifier.

Known NIZK proof. We also utilize the non-interactive zero-knowledge (NIZK) proof for proving that *a system* of linear equations of the form $g_0 = \Pi_{j=1}^{m} g_j^{X_j}$, over a group G (with pairing as above) is satisfiable, where X_j are variables and $g_0,\cdots,g_m$ are constants in G. This is derived from the result of Groth and Sahai [15]. We will mention more about the NIZK proofs later.

4 Our proposed SCUS_1

In this section, we describe our first selectively convertible undeniable signature (SCUS) scheme and analyze its securities.

We first need the following ingredients, which operate on a common pairing group $\mathbb{PG} = (G, G_T, q = |G|, g, \hat{e}: G \times G \to G_T)$. The pairing group is implicitly included in the public keys of the following schemes.

Generic Bilinear Map Signature Scheme GBM [16]. The signature scheme GBM = (GBM.Kg, GBM.Sign, GBM.Vrf) is briefly recalled with some minor modifications as follows.

GBM.Kg(1^{κ}): Generate $x \xleftarrow{\$} Z_q$, $X \leftarrow g^x$, and $H: \{0,1\}^* \to G$. Return the verifying key $pk_1 = (X, H, \eta)$ where $\eta = 70$ and the signing key $sk_1 = x$. (The public key size $|pk_1| \approx 162 \cdot \log_2 q$ bits, according to the estimation in [16], due to the concrete description of H.)

GBM.Sign$(sk_1, m \in \{0,1\}^*)$: $s \xleftarrow{\$} \{0,1\}^{\eta}$, $\rho \leftarrow H(m)^{\frac{1}{x+s}} \in G$. Return $(s,\rho) \in \{0,1\}^{\eta} \times G$ as the signature on m.

GBM.Vrf$\big(pk_1, m, (s,\rho)\big)$: Check that $(s,\rho) \in \{0,1\}^{\eta} \times G$ and $\hat{e}(\rho, X \cdot g^s) = \hat{e}(H(m), g)$. Return 1 if all checks pass, else return 0.

The signature scheme is known to be strongly unforgeable (suf-cma secure) under the strong Diffie-Hellman assumption. To be complete, the proof given in [16] is for the uf-cma case, but holds even for suf-cma security.

Linear Encryption [2]. The linear encryption scheme $\mathsf{LE}=(\mathsf{LE.Kg}, \mathsf{LE.Enc}, \mathsf{LE.Dec})$ is as follows.

$\mathsf{LE.Kg}(1^\kappa)$: Generate $x_1, x_2 \xleftarrow{\$} Z_q$ and set $g_1 \leftarrow g^{1/x_1}$, $g_2 \leftarrow g^{1/x_2}$. Return the public key $pk_2 = (g_1, g_2)$ and the secret key $sk_2 = (x_1, x_2)$.

$\mathsf{LE.Enc}(pk_2, m \in G)$: Choose $r_1, r_2 \xleftarrow{\$} Z_q$ and set $u_1 \leftarrow g_1^{r_1}$, $u_2 \leftarrow g_2^{r_2}$, $u_3 \leftarrow m \cdot g^{r_1+r_2}$. Return (u_1, u_2, u_3) as the ciphertext of m.

$\mathsf{LE.Dec}\big(sk_2, (u_1, u_2, u_3)\big)$: Return $u_3/(u_1^{x_1}u_2^{x_2})$.

The scheme is ind-cpa-secure under the decision linear assumption [2].

Our Proposal SCUS_1. The scheme is described as follows.

$\mathsf{KeyGen}(1^\kappa)$: Run $\mathsf{GBM.Kg}(1^\kappa)$ and $\mathsf{LE.Kg}(1^\kappa)$ to get (pk_1, sk_1) and (pk_2, sk_2). Return the public key $pk = (pk_1, pk_2)$ and the signing key $sk = (sk_1, sk_2)$.

$\mathsf{USign}(sk, m)$: First, generate $r_1, r_2 \xleftarrow{\$} Z_q$, and set $u_1 \leftarrow g_1^{r_1}$, $u_2 \leftarrow g_2^{r_2}$, and $\overline{m} = m \parallel u_1 \parallel u_2$. Next, sign on $\overline{m}$ to get $\big(s, \rho = H(\overline{m})^{\frac{1}{x+s}}\big) \xleftarrow{\$} \mathsf{GBM.Sign}(sk_1, \overline{m})$. Then, encrypt ρ in the ciphertext $(u_1, u_2, u_3 = \rho \cdot g^{r_1+r_2})$. Return the undeniable signature $\sigma = (s, u_1, u_2, u_3)$.

$\mathsf{Convert}(sk, m, \sigma)$: Parse σ as $(s, u_1, u_2, u_3) \in \{0,1\}^\eta \times G^3$. Let $\rho \leftarrow u_3/(u_1^{x_1}u_2^{x_2})$. If (s, ρ) is not a GBM signature on $m \parallel u_1 \parallel u_2$ then return $\perp$. Otherwise, return the converter $(\rho, \pi) \in G \times G^{12}$, where π is a NIZK proof proving (with secrets x_1, x_2):

$$g = g_1^{x_1}, g = g_2^{x_2}, u_3/\rho = u_1^{x_1}u_2^{x_2}. \tag{1}$$

Such a NIZK proof π can be efficiently created using the result of Groth and Sahai [15]. See Appendix B for the concrete description of π. For another (but storage-expensive) method of converting, see Footnote[3].

For all conversion, release $sk_2 = (x_1, x_2)$ so that everyone can compute $\rho = u_3/(u_1^{x_1}u_2^{x_2})$ and then check whether (s, ρ) is a valid GBM signature on $m \parallel u_1 \parallel u_2$. Note that in this case, our proposal becomes a regular signature scheme equivalent to the GBM scheme.

$\mathsf{Verify}(pk, m, \sigma, cvt)$: Parse σ as $(s, u_1, u_2, u_3) \in \{0,1\}^\eta \times G^3$ and cvt as $(\rho, \pi) \in G \times G^{12}$. Return 1 (meaning, valid) if π is a valid proof of the equations (1), and (s, ρ) is a valid GBM signature on $m \parallel u_1 \parallel u_2$. Otherwise return 0.

Confirm: On common input $pk, (m, \sigma)$, the signer and the verifier execute

$$\mathrm{PoK}\Big\{(x, a, b) : g^x = y \wedge g_1^a = (yg^s)^{-1} \wedge g_2^b = (yg^s)^{-1} \wedge u_3^x u_1^a u_2^b = H(m \parallel u_1 \parallel u_2) u_3^{-s}\Big\}.$$

Intuitively, the equations first show that $a = -x_1(x+s)$ and $b = -x_2(x+s)$ where $x_1 = \log_{g_1} g$ and $x_2 = \log_{g_2} g$. With the values a, b, the final equation is equivalent to $u_3/(u_1^{x_1}u_2^{x_2}) = H(m \parallel u_1 \parallel u_2)^{1/(x+s)}$. Since $u_1, u_2 \in G$, a cyclic group, there exist r_1, r_2 such that $u_1 = g_1^{r_1}$ and $u_2 = g_2^{r_2}$, and thus $u_1^{x_1} = g^{r_1}$, $u_2^{x_2} = g^{r_2}$. Hence, $u_3 = H(m \parallel g_1^{r_1} \parallel g_2^{r_2})^{1/(x+s)} \cdot g^{r_1+r_2}$, showing that $\sigma = (s, u_1, u_2, u_3)$ is indeed produced by USign on m. The zero-knowledge proof of knowledge can be implemented using known ZKIPs described in Sect. 3.

[3]The method, inspired by Damgard and Pedersen [12], is to store the randomness r_1, r_2 used in signing and later release them as converter. Then, everyone can check $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$ and compute ρ as $u_3/g^{r_1+r_2}$. With the method, the signer needs to store about $340 \cdot q_{cv}$ bits, where q_{cv} is the total number of selective conversion.

Disavowal: On common input $pk, (m, \sigma)$, the signer sends a value $U \neq 1$ to the verifier, and both execute

$$\mathrm{PoK}\Big\{(c, d, f, r) : g^c(y^{-1}g^{-s})^r = g_1^d(yg^s)^r = g_2^f(yg^s)^r = 1 \\ \wedge\, U = u_3^c \cdot u_1^d \cdot u_2^f \cdot H(m \parallel u_1 \parallel u_2)^{-r} \Big\}.$$

Intuitively, the equations of the first line give us $c = r(x + s)$, $d = -rx_1(x + s)$, and $f = -rx_2(x + s)$. Substituting these values to the second line equation and noting that $U \neq 1$ show $u_3/(u_1^{x_1}u_2^{x_2}) \neq H(m \parallel u_1 \parallel u_2)^{1/(x+s)}$, and thus (m, σ) is invalid. The disavowal protocol is also implemented using known ZKIPs or NIZK proof in Sect. 3. Note that the NIZK proof for the disavowal protocol gives a way to "convert" (namely, make publicly-verifiable) invalid signatures.

Above, if the confirmation protocol fails, then the disavowal protocol is run. If both fails, we conclude that the signer is cheating (or not cooperating). We now consider securities of SCUS_1, which are ensured by the following theorems.

Theorem 5 (Strong unforgeability)**.** *The proposed SCUS_1 scheme is strongly unforgeable if the signature scheme GBM is suf-cma-secure. Moreover, given a forger $\mathcal{F}$ against SCUS_1, there exists another forger $\mathcal{F}'$ against the GBM signature scheme such that*

$$\mathbf{Adv}^{sforge}_{\mathsf{SCUS}_1}(\mathcal{F}) \leq \mathbf{Adv}^{suf-cma}_{\mathsf{GBM}}(\mathcal{F}'),$$

$$\mathbf{T}(\mathcal{F}') = O(q_{conf/dis}) \cdot \mathbf{T}(\mathcal{F}),$$

where $q_{conf/dis}$ is the total number of confirmation/disavowal queries $\mathcal{F}$ made, and $\mathbf{T}$ expresses the running time.

Proof. Given a forger $\mathcal{F}$ against the proposed SCUS scheme, we build a forger $\mathcal{F}'$ against the ordinary GBM signature scheme. The input of $\mathcal{F}'$ is $pk_1 = (\mathbb{PG}, X = g^x, H, \eta = 70)$ and $\mathcal{F}'$ has a signing oracle $\mathsf{GBM.Sign}(sk_1 = x, \cdot)$. $\mathcal{F}'$ itself chooses the keys for the linear encryption scheme $sk_2 = (x_1, x_2) \xleftarrow{\$} Z_q^2$, and $pk_2 = (g_1 = g^{1/x_1}, g_2 = g^{1/x_2})$.

The forger $\mathcal{F}'$ gives $pk = (pk_1, pk_2)$ as the public key of the SCUS scheme to $\mathcal{F}$, and begins to simulate the environment for the SCUS forger as follows:

– Signing query m: $\mathcal{F}'$ chooses $r_1, r_2 \xleftarrow{\$} Z_q$ and sets $u_1 \leftarrow g_1^{r_1}$, $u_2 \leftarrow g_2^{r_2}$, and then calls $m \parallel u_1 \parallel u_2$ to its own signing oracle $\mathsf{GBM.Sign}(sk_1 = x, \cdot)$ to obtain the GBM signature (s, ρ). $\mathcal{F}'$ then returns the undeniable signature $(s, u_1, u_2, u_3 = \rho \cdot g^{r_1+r_2})$ to $\mathcal{F}$.

– Confirmation/disavowal query (m, σ): Parse σ as $(s, u_1, u_2, u_3) \in \{0,1\}^\eta \times G^3$. Decrypt (u_1, u_2, u_3) to get ρ (since $\mathcal{F}'$ has sk_2), and then check whether (s, ρ) is a valid GBM signature on $m \parallel u_1 \parallel u_2$ or not. If it is the case, return 1 and run the confirmation protocol with $\mathcal{F}$ (acting as a cheating verifier); otherwise, return 0 and run the disavowal protocol with $\mathcal{F}$ accordingly. The protocols are simulatable using the rewinding technique [14] since they are zero-knowledge.

– Convert query (m, σ): Parse $\sigma = (s, u_1, u_2, u_3) \in \{0,1\}^\eta \times G^3$. Let $\rho \leftarrow u_3/(u_1^{x_1}u_2^{x_2})$. If (s, ρ) is a valid GBM signature on $m \parallel u_1 \parallel u_2$, then compute the NIZK proof π (using secrets x_1, x_2) of the equations (1), and finally return the converter (ρ, π). Otherwise, if (s, ρ) is not a valid GBM signature on $m \parallel u_1 \parallel u_2$, then return $\perp$.

At the end, the forger $\mathcal{F}$ outputs $\big(m^*, \sigma^* = (s^*, u_1^*, u_2^*, u_3^*)\big)$. If $\mathcal{F}$ succeeds, (m^*, σ^*) is a valid pair of the SCUS scheme, we then have

$$\frac{u_3^*}{(u_1^*)^{x_1}(u_2^*)^{x_2}} = H(m^* \parallel u_1^* \parallel u_2^*)^{\frac{1}{x+s^*}}.$$

Based on the above equation, $\mathcal{F}'$ outputs $\big(m^* \parallel u_1^* \parallel u_2^*, (s^*, \frac{u_3^*}{(u_1^*)^{x_1}(u_2^*)^{x_2}})\big)$ as a forgery of the ordinary GBM signature scheme. It is clear that the forgery is valid, and we just need to prove that it is different from all message-signature pairs appeared at the oracle $\mathsf{GBM.Sign}(sk_1 = x, \cdot)$. By the contrary, suppose that $\big(m^* \parallel u_1^* \parallel u_2^*, (s^*, \frac{u_3^*}{(u_1^*)^{x_1}(u_2^*)^{x_2}})\big) = \big(m \parallel u_1 \parallel u_2, (s, \rho)\big)$, a previously-appeared pair at the signing oracle of $\mathcal{F}'$. Thus $m = m^*$, $u_1 = u_1^*$, $u_2 = u_2^*$, $s = s^*$, and furthermore

$$u_3^* = \rho \cdot (u_1^*)^{x_1}(u_2^*)^{x_2} = \rho \cdot (u_1)^{x_1}(u_2)^{x_2} = u_3,$$

and hence $\big(m^*, \sigma^* = (s^*, u_1^*, u_2^*, u_3^*)\big) = \big(m, \sigma = (s, u_1, u_2, u_3)\big)$, which is a contradiction to the success of $\mathcal{F}$.

The running time of $\mathcal{F}'$ is $O(q_{conf/dis})$ times that of $\mathcal{F}$ due to the rewinding used in the simulation of the confirmation and disavowal protocol. □

Theorem 6 (Invisibility)**.** *The SCUS_1 scheme satisfies invisibility. Moreover, given a distinguisher $\mathcal{D}$ against SCUS_1, there exist an $\mathcal{A}_{dlin}$ against the decision linear assumption, and a forger $\mathcal{F}$ against SCUS_1 such that*

$$\mathbf{Adv}^{inv}_{\mathsf{SCUS}_1}(\mathcal{D}) \leq \mathbf{Adv}^{dlin}_{G}(\mathcal{A}_{dlin}) + \mathbf{Adv}^{sforge}_{\mathsf{SCUS}_1}(\mathcal{F}),$$

$$\mathbf{T}(\mathcal{A}_{dlin}) = O(q_{conf/dis}) \cdot \mathbf{T}(\mathcal{D}), \text{ and } \mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{D}),$$

where $\mathbf{T}$ expresses the running time, and $q_{conf/dis}$ is the total number of confirmation/disavowal queries $\mathcal{D}$ makes.

Proof. We proceed in games as follows.

Game 0: This is exactly the definitional game as in Definition 3. Let W_i $(i = 0, 1)$ be the event that the distinguisher $\mathcal{D}$ wins in Game i, we have $\mathbf{Adv}^{inv}_{\mathsf{SCUS}_1}(\mathcal{D}) = \Pr[W_0]$ by definition.

Game 1: This game is the same as Game 0, except that we consider the following distinguisher: $\mathcal{D}$ never issues a convert or confirmation/disavowal query (m, σ) satisfying (1) the pair is valid (namely, $\perp$ or 0 was not returned), and (2) the pair is different from all previously-issued message-signature pairs at the signing oracle.

Obviously, if $\mathcal{D}$ (in Game 0) issues the pair (m, σ) as above, then we can use (m, σ) as a forgery (in the strong sense) of the SCUS_1 scheme. More precisely, we can use $\mathcal{D}$ to build a forger $\mathcal{F}$ against SCUS_1 with $\mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{D})$. Thus, Game 0 and Game 1 are indistinguishable thanks to the strong unforgeability of the scheme, and hence

$$|\Pr[W_0] - \Pr[W_1]| \leq \mathbf{Adv}^{sforge}_{\mathsf{SCUS}_1}(\mathcal{F}).$$

Using the distinguisher $\mathcal{D}$ in Game 1, we now build an adversary $\mathcal{A}_{dlin}$ against the decision linear assumption on G satisfying $\Pr[W_1] \leq \mathbf{Adv}^{dlin}_{G}(\mathcal{A}_{dlin})$. Note that

$$\begin{aligned} \mathbf{Adv}^{inv}_{\mathsf{SCUS}_1}(\mathcal{D}) = \Pr[W_0] &\leq \Pr[W_1] + \mathbf{Adv}^{sforge}_{\mathsf{SCUS}_1}(\mathcal{F}) \\ &\leq \mathbf{Adv}^{dlin}_{G}(\mathcal{A}_{dlin}) + \mathbf{Adv}^{sforge}_{\mathsf{SCUS}_1}(\mathcal{F}), \end{aligned}$$

which completes the proof. Thus the rest is devoted to constructing such $\mathcal{A}_{dlin}$. The input of $\mathcal{A}_{dlin}$ is $(\mathbb{PG}, g_1, g_2, g, g_1^{\alpha}, g_2^{\beta}, T_b)$, where $T_0 = g^{\alpha+\beta}$ and $T_1 = g^{\gamma}$ for $\alpha, \beta, \gamma \xleftarrow{\$} Z_q$. The adversary $\mathcal{A}_{dlin}$ itself sets up the keys for GBM signature scheme: $sk_1 = x \xleftarrow{\$} Z_q$ and $pk_1 = (g^x, H, \eta = 70)$. Then $\mathcal{A}_{dlin}$ gives $pk = (pk_1, g_1, g_2)$ to $\mathcal{D}$ and begins to simulate the environment for the distinguisher as follows:

– Signing query m: $\mathcal{A}_{dlin}$ chooses the randomness $r_1, r_2 \xleftarrow{\$} Z_q$ and $s \xleftarrow{\$} \{0,1\}^{\eta}$, and computes $\rho \leftarrow H(m \parallel u_1 \parallel u_2)^{1/(x+s)}$ where $u_1 = g_1^{r_1}$ and $u_2 = g_2^{r_2}$. It then lets $u_3 \leftarrow \rho \cdot g^{r_1+r_2}$ and returns $\sigma = (s, u_1, u_2, u_3)$ to $\mathcal{D}$ as the undeniable signature on m. The adversary $\mathcal{A}_{dlin}$ internally keeps a record of the values ρ, and also lets $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$ for later use, where $\mathcal{Q}$ is an initially empty set of message-signature pairs appeared so far.

– Convert query (m, σ): If $(m, \sigma) \in \mathcal{Q}$ then return the corresponding recorded ρ and a simulated NIZK proof π_{sim} (of the equations (1)) to $\mathcal{D}$. If $(m, \sigma) \notin \mathcal{Q}$ then return $\perp$ to $\mathcal{D}$. The reasoning behind this simulation is that if $(m, \sigma) \notin \mathcal{Q}$ then the pair must be invalid since we are in Game 1.

– Confirmation/disavowal query (m, σ): Like the simulation for convert query above, if $(m, \sigma) \in \mathcal{Q}$ then return 1 and run the confirmation protocol with $\mathcal{D}$; otherwise return 0 and run the disavowal protocol. The protocols are simulatable using the rewinding technique [14] since they are zero-knowledge.

– Challenge query m^*: Let $u_1^* \leftarrow g_1^{\alpha}$ and $u_2^* \leftarrow g_2^{\beta}$. Choose $s^* \xleftarrow{\$} \{0,1\}^{\eta}$ and then compute $\rho^* \leftarrow H(m^* \parallel u_1^* \parallel u_2^*)^{1/(x+s^*)}$ and $u_3^* \leftarrow \rho^* \cdot T_b$. Return $\sigma^* = (s^*, u_1^*, u_2^*, u_3^*)$ to $\mathcal{D}$.

Note that if $b = 0$ then $T_b = T_0 = g^{\alpha+\beta}$, so that σ^* is a valid undeniable signature on m^*. If $b = 1$ then $T_b = T_1 = g^{\gamma}$ is a random value over G independent of the other values, so that σ^* is also randomly distributed over the signature space $\{0,1\}^{\eta} \times G^3$.

At the end, the distinguisher $\mathcal{D}$ outputs a bit b' as a guess of the hidden bit b. The adversary $\mathcal{A}_{dlin}$ in turn outputs b'. The advantage of $\mathcal{A}_{dlin}$ is exactly the probability $\mathcal{D}$ wins in Game 1, namely $\mathbf{Adv}_G^{dlin}(\mathcal{A}_{dlin}) = \Pr[W_1]$. The running time of $\mathcal{A}_{dlin}$ is $O(q_{conf/dis})$ times that of $\mathcal{D}$ due to the rewinding. □

5 Our proposed SCUS_2

In this section, we describe our second scheme SCUS_2, which is also secure under the same assumptions as those of SCUS_1. The scheme SCUS_2 uses the Boneh-Boyen [1] signature scheme as a component. We first recall the Boneh-Boyen signature scheme, basing on a pairing group $\mathbb{PG} = (G, G_T, q = |G|, g, \hat{e} : G \times G \to G_T)$.

Boneh-Boyen Signature Scheme. The signature scheme $\mathsf{BB} = (\mathsf{BB.Kg}, \mathsf{BB.Sign}, \mathsf{BB.Vrf})$ is as follows.

$\mathsf{BB.Kg}(1^{\kappa})$: Generate $g_0 \xleftarrow{\$} G$, $x, y \xleftarrow{\$} Z_q$, $u \leftarrow g^x$, $v \leftarrow g^y$, $z = \hat{e}(g_0, g)$, and a target collision hash $H : \{0,1\}^* \to Z_q$. Return the verifying key $pk_1 = (g_0, u, v, z, H)$ and the signing key $sk_1 = (x, y)$.

$\mathsf{BB.Sign}(sk_1, m)$: $s \xleftarrow{\$} Z_q$, $\rho \leftarrow g_0^{\frac{1}{x+H(m)+ys}} \in G$. Return $(s, \rho) \in Z_q \times G$ as the signature on m.

$\mathsf{BB.Vrf}\big(pk_1, m, (s,\rho)\big)$: Check that $(s,\rho) \in Z_q \times G$ and $\hat{e}\left(\rho, u \cdot g^{H(m)} \cdot v^s\right) = z$. Return 1 if all checks pass, else return 0.

It was proven in [1] that the above signature scheme is suf-cma-secure under the strong Diffie-Hellman assumption.

Our Proposal SCUS_2. The scheme, whose security analysis is given in Appendix C, is described as follows.

$\mathsf{KeyGen}(1^\kappa)$: Run $\mathsf{BB.Kg}(1^\kappa)$ and $\mathsf{LE.Kg}(1^\kappa)$ to get (pk_1, sk_1) and (pk_2, sk_2). Return the public key $pk = (pk_1, pk_2)$ and the signing key $sk = (sk_1, sk_2)$.

$\mathsf{USign}(sk, m)$: First, generate $r_1, r_2 \xleftarrow{\$} Z_q$, and set $u_1 \leftarrow g_1^{r_1}$, $u_2 \leftarrow g_2^{r_2}$, and $\overline{m} = m \parallel u_1 \parallel u_2$. Next, sign on $\overline{m}$ to get $\left(s, \rho = g_0^{\frac{1}{x+H(\overline{m})+ys}}\right) \xleftarrow{\$} \mathsf{BB.Sign}(sk_1, \overline{m})$. Then, encrypt ρ in the ciphertext $(u_1, u_2, u_3 = \rho \cdot g^{r_1+r_2})$. Return the undeniable signature $\sigma = (s, u_1, u_2, u_3)$.

$\mathsf{Convert}(sk, m, \sigma)$: The same as that of SCUS_1, except now checking whether (s,ρ) is a BB signature or not. Also, for all conversion, release $sk_2 = (x_1, x_2)$, so that our proposal becomes a regular signature scheme equivalent to the BB scheme.

$\mathsf{Verify}(pk, m, \sigma, cvt)$: The same as that of SCUS_1, except now checking whether (s,ρ) is a valid BB signature or not.

$\mathsf{Confirm}$: On common input pk, m, $\sigma = (s, u_1, u_2, u_3)$, the signer and the verifier execute

$$\mathrm{PoK}\Big\{(a,b,c) : g^a = uv^s \wedge g_1^b = g_2^c = \Big(uv^s g^{H(m\|u_1\|u_2)}\Big)^{-1} \wedge u_3^a u_1^b u_2^c = g_0 u_3^{-H(m\|u_1\|u_2)}\Big\}.$$

The first three equations show that $a = x + ys$, $b = -x_1\,(x + H\,(m \parallel u_1 \parallel u_2) + ys)$, and $c = -x_2\,(x + H\,(m \parallel u_1 \parallel u_2) + ys)$, where $x_1 = \log_{g_1} g$ and $x_2 = \log_{g_2} g$. With the values a, b, c, the final equation is equivalent to $u_3/(u_1^{x_1} u_2^{x_2}) = g_0^{1/(x+H(m\|u_1\|u_2)+ys)}$, showing that (m,σ) is valid. The zero-knowledge proof of knowledge can be implemented using known ZKIPs or NIZK proof described in Sect. 3.

$\mathsf{Disavowal}$: On common input pk, m, $\sigma = (s, u_1, u_2, u_3)$, the signer sends a value $U \neq 1$ to the verifier, and both execute

$$\mathrm{PoK}\Big\{(d,e,f,r) : \quad g^d (u g^{H(m\|u_1\|u_2)} v^s)^{-r} = 1 \wedge\ g_1^e (u g^{H(m\|u_1\|u_2)} v^s)^r = 1$$
$$\wedge\ g_2^f (u g^{H(m\|u_1\|u_2)} v^s)^r = 1 \wedge\ U = u_3^d \cdot u_1^e \cdot u_2^f \cdot g_0^{-r}\ \Big\}.$$

Intuitively, the first three equations give us $d = r(x + H(m \parallel u_1 \parallel u_2) + ys)$, $e = -rx_1(x + H(m \parallel u_1 \parallel u_2) + ys)$, and $f = -rx_2(x + H(m \parallel u_1 \parallel u_2) + ys)$. Substituting these values to the last equation and noting that $U \neq 1$ show $u_3/(u_1^{x_1} u_2^{x_2}) \neq g_0^{1/(x+H(m\|u_1\|u_2)+ys)}$, and thus (m,σ) is invalid. The disavowal protocol is also implemented using known ZKIPs or NIZK proof in Sect. 3.

6 Conclusion

We propose two undeniable signature schemes with both selective and all conversion in the dlog-based setting, in the standard model. Our proposals enjoy formally-proved security and being very practical with short signatures. Moreover, the confirmation and disavowal protocols are of (minimal) four moves, and even become non-interactive without the random oracle model. We also point out a flaw in the scheme of [28].

References

[1] D. Boneh and X. Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[2] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[3] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[4] J. Boyar, D. Chaum, I. Damgård, and T. P. Pedersen. Convertible undeniable signatures. In A. Menezes and S. A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 189–205. Springer, 1990.

[5] C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. In K. Ohta and D. Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 1998.

[6] E. F. Brickell, editor. *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*. Springer, 1993.

[7] J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 351–368. Springer, 2009.

[8] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.

[9] D. Chaum. Zero-knowledge undeniable signatures. In *EUROCRYPT*, pages 458–464, 1990.

[10] D. Chaum and H. V. Antwerpen. Undeniable signatures. In G. Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 212–216. Springer, 1989.

[11] D. Chaum and T. P. Pedersen. Wallet databases with observers. In Brickell [6], pages 89–105.

[12] I. Damgård and T. P. Pedersen. New convertible undeniable signature schemes. In *EUROCRYPT*, pages 372–386, 1996.

[13] S. D. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In M. Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 2003.

[14] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[15] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2008.

[16] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In D. Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2008.

[17] K. Kurosawa and J. Furukawa. Universally composable undeniable signature. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 524–535. Springer, 2008.

[18] K. Kurosawa and S.-H. Heng. 3-Move undeniable signature scheme. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2005.

[19] K. Kurosawa and S.-H. Heng. Relations among security notions for undeniable signature schemes. In R. D. Prisco and M. Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 34–48. Springer, 2006.

[20] K. Kurosawa and T. Takagi. New approach for selectively convertible undeniable signature schemes. In X. Lai and K. Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 428–443. Springer, 2006.

[21] F. Laguillaumie and D. Vergnaud. Short undeniable signatures without random oracles: The missing link. In S. Maitra, C. E. V. Madhavan, and R. Venkatesan, editors, *INDOCRYPT*, volume 3797 of *Lecture Notes in Computer Science*, pages 283–296. Springer, 2005.

[22] J. Monnerat and S. Vaudenay. Generic homomorphic undeniable signatures. In P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 354–371. Springer, 2004.

[23] J. Monnerat and S. Vaudenay. Undeniable signatures based on characters: How to sign with one bit. In F. Bao, R. H. Deng, and J. Zhou, editors, *Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2004.

[24] W. Ogata, K. Kurosawa, and S.-H. Heng. The security of the FDH variant of Chaum's undeniable signature scheme. *IEEE Transactions on Information Theory*, 52(5):2006–2017, 2006.

[25] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Brickell [6], pages 31–53.

[26] D. Pointcheval. Self-scrambling anonymizers. In Y. Frankel, editor, *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 259–275. Springer, 2000.

[27] C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[28] T. H. Yuen, M. H. Au, J. K. Liu, and W. Susilo. (Convertible) undeniable signatures without random oracles. In S. Qing, H. Imai, and G. Wang, editors, *ICICS*, volume 4861 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2007.

A A flaw in [28]

We first show that the scheme of Yuen et al [28] does not have invisibility in the sense of Definition 3. Let us briefly recall their undeniable signature scheme. A signature on a message m is of the form $\sigma = (S_1, S_{2,1}, \ldots S_{2,k})$ where $k = 7$ (see the final remark of the paper), and

$$S_1 = g_2^{\alpha} U^r, \quad S_{2,j} = V_j^r (1 \leq j \leq k),$$

where α is in the secret key, r is random, while g_2, U, V_j are publicly-computable values. Notice that the undeniable signature scheme is *not* strongly unforgeable, since $\sigma' = (S_1 U^t, S_{2,1} V_1^t, \ldots S_{2,k} V_k^t)$ is also valid on the same m for an adversarily-chosen randomness t. (The randomness of the signature becomes $r + t$.)

The attack on the scheme uses the same idea as the one we present at Sect.1.1. Namely, the adversary obtains the challenge σ (which is either random or valid) on its challenge query m, and then submits (m, σ') as above for selective conversion. If the answer is $\perp$, then σ' is not valid on m, and so σ is not a signature on m. If the answer is not $\perp$, σ' is valid on m, and so is σ. The attack is sufficient to show that the scheme of [28] does not satisfy invisibility in the sense of Definition 3.

However, Yuen et al [28] use a weaker (and not natural) definition of invisibility which disallows the convert query (m, σ') as above. In that case, the above attack does not apply, but the invisibility proof (Theorem 2 of [28]) is incorrect in that it makes use of strong unforgeability. Specifically, in the simulation of the confirmation/disavowal oracle, the following reasoning is used: *"Let $\mathcal{L}$ is the set of previously-appeared message-signature pairs at the signing oracle. Upon receiving a confirmation/disavowal query (m, σ), if $(m, \sigma) \in \mathcal{L}$ then return 1 and execute the confirmation protocol, otherwise if $(m, \sigma) \notin \mathcal{L}$ then return 0 and execute the disavowal protocol".*

The above simulation is imperfect and incorrect, since if the adversary submits the above (m, σ') as a confirmation/disavowal query, then $(m, \sigma') \notin \mathcal{L}$, but valid, while the simulation will return 0 and execute the disavowal protocol.

In short, if the strong definition of invisibility (Definition 3) is used, the scheme in [28] is totally broken; while if the weaker definition is used, then the invisibility proof provided in [28] is incorrect.

We find it is hard to apply our technique (of authenticating the randomness) in designing the schemes SCUS_1, SCUS_2 to fix the scheme of Yuen et al, because the latter is directly designed from scratch, employing no standard encryption and signature schemes as ours.

B The NIZK proof for selective conversion

We present the concrete NIZK proof of the equations

$$g = g_1^{x_1}, g = g_2^{x_2}, u_3/\rho = u_1^{x_1} u_2^{x_2},$$

used by the $\mathsf{Convert}$ algorithms of SCUS_1 and SCUS_2. The proof is originally developed by Groth and Sahai [15], but here we follows the exposition of Camenisch, Chandran and Shoup [7] (Section 4.4). Recall that we work on a pairing group $\mathbb{PG} = (G, G_T, q = |G|, g, \hat{e} : G \times G \to G_T)$.

First, a common reference string, which can be kept in the public key of the signer, is generated as follows: $\gamma_1, \gamma_2, \gamma_3 \xleftarrow{\$} G$ and $\vec{\gamma} = (\gamma_0, \gamma_0', \gamma_0'') \xleftarrow{\$} G^3$. Let the common reference string be $crs = (\gamma_1, \gamma_2, \gamma_3, \vec{\gamma})$, and define vectors $\vec{\gamma_1} = (\gamma_1, 1, \gamma_3)$, $\vec{\gamma_2} = (1, \gamma_2, \gamma_3)$.

The prover, with secrets x_1, x_2, works as follows. It chooses random $r_{ij} \xleftarrow{\$} Z_q$, where $1 \leq i, j \leq 2$, and computes

$$\begin{aligned}
\vec{\delta_1} &= \vec{\gamma}^{x_1} \cdot \vec{\gamma_1}^{r_{11}} \cdot \vec{\gamma_2}^{r_{12}} = (\gamma_0^{x_1}\gamma_1^{r_{11}}, \gamma_0'^{x_1}\gamma_2^{r_{12}}, \gamma_0''^{x_1}\gamma_3^{r_{11}+r_{12}}) \in G^3, \\
\vec{\delta_2} &= \vec{\gamma}^{x_2} \cdot \vec{\gamma_1}^{r_{21}} \cdot \vec{\gamma_2}^{r_{22}} = (\gamma_0^{x_2}\gamma_1^{r_{21}}, \gamma_0'^{x_2}\gamma_2^{r_{22}}, \gamma_0''^{x_2}\gamma_3^{r_{21}+r_{22}}) \in G^3,
\end{aligned}$$

where exponentiations and products of the vectors are understood (as usual) as exponentiations and products of the corresponding components. The NIZK proof is

$$\pi = \big(\vec{\delta_1}, \vec{\delta_2}, (g_1^{r_{11}}, g_1^{r_{12}}), (g_2^{r_{21}}, g_2^{r_{22}}), (u_1^{r_{11}} \cdot u_2^{r_{21}}, u_1^{r_{12}} \cdot u_2^{r_{22}})\big) \in G^{12}.$$

Define $E : G \times G^3 \to G_T^3$ sending $\big(\alpha, (\alpha_1, \alpha_2, \alpha_3)\big)$ to $\big(\hat{e}(\alpha, \alpha_1), \hat{e}(\alpha, \alpha_2), \hat{e}(\alpha, \alpha_3)\big)$, which is also a bilinear map. To verify whether $\pi = \big(\vec{\delta_1}, \vec{\delta_2}, (p_1, p_2), (p_1', p_2'), (p_1'', p_2'')\big) \in G^{12}$ proves the equations, one checks whether the following holds

$$\begin{aligned}
E(g_1, \vec{\delta_1}) &= E(g, \vec{\gamma}) \cdot E(p_1, \vec{\gamma_1}) \cdot E(p_2, \vec{\gamma_2}), \\
E(g_2, \vec{\delta_2}) &= E(g, \vec{\gamma}) \cdot E(p_1', \vec{\gamma_1}) \cdot E(p_2', \vec{\gamma_2}), \\
E(u_1, \vec{\delta_1}) \cdot E(u_2, \vec{\delta_2}) &= E(u_3/\rho, \vec{\gamma}) \cdot E(p_1'', \vec{\gamma_1}) \cdot E(p_2'', \vec{\gamma_2}).
\end{aligned}$$

Derived from [7], the NIZK proof has perfect completeness, statistical soundness, and computational zero-knowledge (based on the decision linear assumption).

C Security of SCUS_2

We consider the securities of SCUS_2, which are ensured by the following theorems.

Theorem 7 (Strong unforgeability)**.** *The SCUS_2 scheme is strongly unforgeable if the signature scheme BB is suf-cma-secure. Moreover, given a forger $\mathcal{F}$ against SCUS_2, there exists another forger $\mathcal{F}'$ against the BB signature scheme such that*

$$\mathbf{Adv}^{sforge}_{\mathsf{SCUS}_2}(\mathcal{F}) \leq \mathbf{Adv}^{suf-cma}_{\mathsf{BB}}(\mathcal{F}'),$$

$$\mathbf{T}(\mathcal{F}') = O(q_{conf/dis}) \cdot \mathbf{T}(\mathcal{F}),$$

where $q_{conf/dis}$ is the total number of confirmation/disavowal queries, and $\mathbf{T}$ expresses the running time.

Proof. The proof is essentially the same as that of Theorem 5, so we just outline the main ideas here. The forger $\mathcal{F}'$ first generates the keys (pk_2, sk_2) for the LE scheme, which will be used for the simulation of the convert and confirmation/disavowal oracles. For answering signing queries from $\mathcal{F}$, the forger $\mathcal{F}'$ utilizes its own signing oracle. Finally, $\mathcal{F}$ outputs the pair $\big(m^*, \sigma^* = (s^*, u_1^*, u_2^*, u_3^*)\big)$ satisfying

$$\frac{u_3^*}{(u_1^*)^{x_1}(u_2^*)^{x_2}} = g_0^{\frac{1}{x + H(m^* \| u_1^* \| u_2^*) + ys^*}},$$

so that $\mathcal{F}'$ in turn outputs

$$\left(m^* \parallel u_1^* \parallel u_2^*, \left(s^*, \frac{u_3^*}{(u_1^*)^{x_1}(u_2^*)^{x_2}}\right)\right)$$

as the forgery in the strong sense of the BB signature, completing the proof. □

Theorem 8 (Invisibility)**.** *The SCUS_2 scheme satisfies invisibility. Moreover, given a distinguisher $\mathcal{D}$ against SCUS_2, there exist $\mathcal{A}_{dlin}$ and a forger $\mathcal{F}$ against SCUS_2 such that*

$$\mathbf{Adv}^{inv}_{\mathsf{SCUS}_2}(\mathcal{D}) \leq \mathbf{Adv}^{dlin}_{G}(\mathcal{A}_{dlin}) + \mathbf{Adv}^{sforge}_{\mathsf{SCUS}_2}(\mathcal{F}),$$
$$\mathbf{T}(\mathcal{A}_{dlin}) = O(q_{conf/dis}) \cdot \mathbf{T}(\mathcal{D}), \text{ and } \mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{D}),$$

where $\mathbf{T}$ expresses the running time, and $q_{conf/dis}$ is the total number of confirmation/disavowal queries $\mathcal{D}$ makes.

Proof. The proof follows along the line of that of Theorem 6, except that $\mathcal{A}_{dlin}$ generates the keys for the BB signature scheme, and uses them to simulate the signing and challenge oracle for $\mathcal{D}$. The rest remains the same. □