Provably Secure Convertible Undeniable Signatures with Unambiguity

Le Trieu Phong[1], Kaoru Kurosawa[2], and Wakaha Ogata[3]

[1] NICT, Japan, phong@nict.go.jp
[2] Ibaraki University, Japan, kurosawa@mx.ibaraki.ac.jp
[3] Tokyo Institute of Technology, Japan, wakaha@mot.titech.ac.jp

Abstract. This paper shows some efficient and provably-secure convertible undeniable signature schemes (with both selective conversion and all conversion), in the standard model and discrete logarithm setting. They further satisfy unambiguity, which is traditionally required for anonymous signatures. Briefly, unambiguity means that it is hard to generate a (message, signature) pair which is valid for two *different* public-keys. In other words, our schemes can be viewed as anonymous signature schemes as well as convertible undeniable signature schemes. Besides other applications, we show that such schemes are very suitable for anonymous auction.

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

1 Introduction

1.1 Background

Undeniable Signatures. Almost twenty years ago, Chaum and van Antwerpen [11] 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 [5], electronic cash [6, 12, 35], confidential business agreement [13]. There have been a wide range of research on the concept [5, 10, 13, 19, 24–31, 40], to list just a few. Most of the papers are in the random oracle model, with (even arbitrary) short signatures [30], or extensive security consideration of a classical scheme

[31]. In the standard model, the first efficient proposal is that of Laguillaumie and Vergnaud [28] (but relying on a non-standard and strong assumption for invisibility).

In order to link undeniable signature to regular signature, Boyar et al [5] 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 [5] 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 [6], so the more efficient the former is, the more practical the latter can be realized. For more applications, the readers may find in [5, 13]. We also note that the above mentioned work of Laguillaumie and Vergnaud [28], while producing very short signatures (of about 170 bits), does not support any kinds of conversion.

In an attempt to realize practical US schemes with conversions, Damgard and Pedersen [13] 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 [40] 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 [40] is claimed to satisfy invisibility, but in Appendix A, we point out that the claim is incorrect. More recently, El Aimani [14] proposed some generic approaches for building efficient undeniable signature schemes, but with no selective conversion. In the full version [17] of [14], El Aimani claims selective conversion property, but we observe that the claim is correct only if the signer is honest.

However, there exists no convertible undeniable signature scheme which satisfies *unambiguity* which will be explained below.

Anonymous Signatures. The concept is proposed by Yang et al [39] (at PKC '06), and has further study in [1,18,36,41]. Anonymous signatures and undeniable signatures share the same goal of ensuring anonymity (implied by invisibility in this paper) by not revealing the link between signatures and public-keys. However, compared to undeniable signature schemes, anonymous signature schemes do not necessarily have confirmation/disavowal protocols; and yet they have one more security notion called *unambiguity*.

To explain more about anonymous signatures, let us recall its typical application suggested in all previous works, which is anonymous auction where Alice (with pk_A) wishes to place a bid with value bid_A. She wants to be able to claim the bid as hers in case it wins, but otherwise wishes to remain anonymous. The natural solution is to provide, at bidding time, the values bid_A, pk_A, as well as her anonymous signature of bid_A. Later, when the result is announced, and if Alice has won, she can release the relevant opening information to claim her bid.

We however observe that the above usage of anonymous signatures in auction may cause trouble, which is overlooked by previous works. Imagine a situation in which Alice has won, but refuses to provide the opening information. The

natural solution for the auctioneer is to choose the second-highest bid_B of Bob as the winning bid. The real trouble now is that, if Alice and Bob cooperate, they will win every auction! Alice places the highest bid just after Bob, and then refuses to open her signature on the bid, so that Bob will be the winner. This is clearly unfair to other players in the auction. All existing works on anonymous signatures have not noticed the situation that either the winner refuses to open, or there is cooperation between two users[4].

To overcome the above trouble, we then suggest that one should use undeniable signature schemes with selective conversion in anonymous auction, *provided that* they meet all security notions of anonymous signatures. Alice then cannot deny her signature of the bid anymore, since the auctioneer can execute the confirmation and disavowal protocols to check.

Let us now explain the unambiguity notion [1] (aka, unpretendability [36]). It intuitively ensures that if Alice has won, and releases the opening information to claim her bid, then no one else can claim that bid. Previously, unambiguity was not considered as a security notion for undeniable signature schemes. However, to serve in the context of anonymous auction as we suggested above, undeniable signature schemes must satisfy unambiguity.

1.2 Our contribution

We propose two convertible undeniable signature schemes satisfying anonymity, called SCUS_1 and SCUS_2. They have the following properties.

- The schemes support both selective and all conversion. Moreover, they enjoy formally-proven security in the standard model, relying on the strong Diffie-Hellman (sDH) and the decision linear (DLIN) assumption. Their confirmation and disavowal protocols are of (minimal) four moves[5].
- 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, the piece of information is also $2 \cdot |q|$ bits if we accept stateful signers; otherwise, we employ the NIZK proof of Groth and Sahai [21], and need to release a few more bits.
- Both SCUS_1 and SCUS_2 additionally meet the unambiguity notion, under the discrete log assumption. Therefore, they can be used in anonymous auction to detect the winner in case she refuses to open (namely, convert) her signature.

[4] Interestingly, we find that what we discuss for anonymous auction still applies in principle to Yahoo auction in Yapan. Namely, in the Yahoo auction, if two identities (e.g., of one person) cooperate in the way we have described, they will have advantages over ones proceeding honestly. The point is in the Yahoo auction, the winning identity can easily deny contacting the seller for paying process, making the seller to choose the identity with second-highest bid as the winner.

[5] We remark that the 3-move scheme of Kurosawa and Heng [25] is insecure, as shown by Ogata et al in [31] (Sect.V.D, page 2013), who furthermore point out that any 3-move (HVZK) confirmation/disavowal protocols are not secure against active attacks.

It is worth noting that it is unknown whether previous undeniable signature schemes with selective conversion have this additional property.

Above, the scheme SCUS_1 produces shorter signatures than SCUS_2, but the public key of SCUS_1 (of $170 \cdot |q|$ bits) is much longer than that of SCUS_2 (of $12 \cdot |q|$ bits). Choosing which one to use thus depends on specific applications.

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

Sign-then-Encrypt Paradigm. We re-utilize an elegant paradigm introduced by Damgard and Pedersen [13] in which 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 [13] have overcome (1) but not (2). For (3), they suggested a method of storing all randomness previously used in signing. We suggest another method by using the efficient NIZK proof of Groth and Sahai [21], 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 [22] and the linear encryption [3] (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 $x_1 (= \mathrm{dlog}_{g_1} g), x_2 (= \mathrm{dlog}_{g_2} g)$, 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 [9], 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 [21], 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 [2], 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 [4] so that the unforgeability of the resulting undeniable signature scheme relies on the CDH assumption in bilinear group. We do not explicitly consider the random oracle scheme in this paper.

More Related Works. Subsequent to a preliminary version of this work [34] on the Eprint, Schuldt, Matsuura [38], and Huang, Wong [23] have suggested

some other schemes with interesting additional properties. Both works indicate that, if using NIZK proofs in undeniable signatures, the common reference string must be legitimately set up (say, by a trusted party like the CA in PKI). Unfortunately, the scheme of Huang and Wong [23] turned out not satisfying anonymity, as shown in [38]. The scheme of [38], while relying on a more standard assumption, produces longer signatures (or public keys) than the ones in this paper. Both works [23, 38] do not consider unambiguity.

Independently with us, El Aimani [15] also discovered the usage of the NIZK of Groth and Sahai [21] in the context of confirmer signatures. The sign-then-encrypt approach is also used to build confirmer signatures in [16] in an abstract manner. As a trade-off to its generality, the construction in [16] has to employ the cut-and-choose technique for the confirmation and disavowal protocols, and hence the protocols are not of constant rounds (say, 80 rounds to reach 2^{-80} soundness error). In contrast, we take a concrete approach in this paper, resulting in schemes with minimal 4-round protocols.

The above sign-then-encrypt paradigm has also been successfully re-used in [33] in the RSA-based setting, creating RSA-based US schemes supporting (selective and all) conversions, with signatures of $(80 + 2 \cdot 1024)$ bits, converters of 1024 bits, while the securities rely on the strong RSA assumption and the decisional N-th residuosity (DNR) assumption in the standard model. Note that the RSA-based schemes give longer signatures than dlog-based schemes, as usual.

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 signature scheme* $\mathsf{SCUS} = (\mathsf{KeyGen}, \mathsf{USign}, \mathsf{Convert}, \mathsf{Verify}, \mathsf{Confirm}, \mathsf{Disavowal})$ *consists of four algorithms and two protocols whose descriptions are as follows.*

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

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

– $\mathsf{Convert}(sk, m, \sigma) \to cvt/\perp$*: 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* $\perp$*.* [6]

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

[6] 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., [6]). However, we note in our proposed schemes, the signer can even "convert" invalid signatures by making the disavowal protocol non-interactive (via Groth-Sahai result [21], as seen later).

– *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.*

Definition 2 (Unforgeability and strong unforgeability of SCUS) *A selectively convertible undeniable signature scheme SCUS is said to be existential unforgeable under adaptive chosen message attack if no poly-time forger* $\mathcal{F}$ *has a non-negligible advantage in the following game: at the beginning,* $\mathcal{F}$ *is given the public key* pk*. Then* $\mathcal{F}$ *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 the answers of 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* $\mathcal{F}$*.*

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. We note that this definition is new to this work.

Definition 3 (Strong invisibility) *A selectively-convertible undeniable signature scheme SCUS satisfies strong invisibility under adaptive chosen message attack if no poly-time distinguisher* $\mathcal{D}$ *has a non-negligible advantage in the following game. At first,* $\mathsf{KeyGen}(1^\kappa) \rightarrow (pk, sk)$*, and then* $\mathcal{D}$ *is given the public key* pk*. Then* $\mathcal{D}$ *is permitted to issue a series of queries: signing queries, convert queries, confirmation/disavowal queries, as in Definition 2.*

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 (which only depends on the security parameter* κ*, and not on* pk, sk*).*

The distinguisher $\mathcal{D}$ may additionally issue signing queries, convert queries, confirmation/disavowal queries with the only restriction that no confirmation/disavowal query and convert query (m^*, σ^*) are allowed.

At the end, $\mathcal{D}$ outputs a bit b' as the guess for b. The distinguisher wins the game if and only if $b' = b$ and its advantage is defined as $\mathbf{Adv}^{inv}_{\mathsf{SCUS}}(\mathcal{D}) = |\Pr[b' = b] - 1/2|$.

Remarks 1 *Above, there are some subtleties. First, 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 the challenge query. 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, where one piece of software may be signed many times. This second subtlety makes our definition differ from and stronger than previous ones (say, that of [31]). A scheme meeting the (weak) definition as in [31] 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).

Similarly to the second point above, we believe that strong unforgeability is very suitable for undeniable signature schemes, especially in the context of licensing software. Our proposals fortunately meet these strong notions of security.

Another security notion for undeniable signatures is anonymity, intuitively ensuring that given a message-signature pair, it is hard to know who *produces the pair. As pointed out in [19], invisibility implies anonymity if all signers share a common signature space, a condition fulfilled by our proposals. We thus focus on invisibility in the rest of this paper.*

Definition 4 (Standard signature schemes) *A signature scheme S = (Kg, Sign, Vrf) is as follows. On input 1^κ, the key generation algorithm Kg produces the public key pk and the secret signing 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 $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.

Dlog Assumption. The assumption claims that, given $\mathbb{PG}$ as above, and for all poly-time adversary $\mathcal{A}$, $\mathbf{Adv}^{dlog}_{G,\mathbb{PG}}(\mathcal{A}) = \Pr[h = g^x : g, h \xleftarrow{\$} G; x \xleftarrow{\$} \mathcal{A}(g, h, \mathbb{PG})]$ is negligible.

Decision Linear Assumption. Given a pairing group $\mathbb{PG}$, the assumption, first formalized in [3], asserts that the following advantage of a poly-time adversary $\mathcal{A}$ is negligible in the security parameter κ.

$$\mathbf{Adv}^{dlin}_{G}(\mathcal{A}) = \left| \Pr \left[b' = b : \begin{array}{c} \alpha, \beta, \gamma \xleftarrow{\$} Z_q; g_1, g_2, g_3 \xleftarrow{\$} G; \\ T_0 \leftarrow g_3^{\alpha+\beta}; T_1 \leftarrow g_3^{\gamma}; b \xleftarrow{\$} \{0, 1\}; \\ b' \xleftarrow{\$} \mathcal{A}(\mathbb{PG}, g_1, g_2, g_3, g_1^{\alpha}, g_2^{\beta}, T_b) \end{array} \right] - \frac{1}{2} \right|.$$

Known Dlog-Based ZKIP. We use known techniques for proving statements about discrete logarithms, such as (1) proof of knowledge of discrete logarithm [37]; (2) proof of knowledge of an element representation in a prime order group [32]; 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$\{(x_1, x_2)$: $y = g^{x_1} \wedge U = u_1^{x_1} u_2^{x_2}\}$ 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 utilize the non-interactive zero-knowledge (NIZK) proof for proving that *a system* of 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 $g_0, \ldots, g_m$ are constants in G. This is derived from the result of Groth and Sahai [21]. 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.

4.1 Building blocks

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 [22]. The signature scheme GBM = (GBM.Kg, GBM.Sign, GBM.Vrf) is briefly recalled with some minor modifications as follows.

$\mathsf{GBM.Kg}(1^\kappa)$: 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 [22], due to the concrete description of H.)

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

$\mathsf{GBM.Vrf}\big(pk_1, m, (s, \rho)\big)$: 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 [22] is for the uf-cma case, but holds even for suf-cma security.

Linear Encryption [3]. The linear encryption scheme 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 [3].

4.2 The scheme $\mathsf{SCUS_1}$

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 $(s, u_1, u_2, u_3) \in \{0,1\}^\eta \times G^3$, and 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 [21]. See Appendix B for the concrete description of π.

Another method of converting, inspired by Damgard and Pedersen [13], 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 $u_3/g^{r_1+r_2}$.

To do 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.

Verify(pk, m, σ, 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. (We omit details when $cvt = (r_1, r_2)$.)

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

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

Intuitively, the equations first show that $a = -x_1(x+s)$ and $b = -x_2(x+s)$ where $x = \mathrm{dlog}_g(X)$, $x_1 = \mathrm{dlog}_{g_1} g$ and $x_2 = \mathrm{dlog}_{g_2} g$. 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.

In the above PoK, the signer must also prove the knowledge of the secret key corresponding to the public key, namely (x, x_1, x_2) satisfying $g^x = X, g = g_1^{x_1} = g_2^{x_2}$. We omit these types of conditions hereafter in all PoKs for clarity.

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

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

Intuitively, the equations of the first line give us $c = r(x+s)$, $d = -r x_1 (x+s)$, and $f = -r x_2 (x+s)$. 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 in Appendix C.

Theorem 6 (Strong invisibility) *The SCUS_1 scheme satisfies strong 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 $pk_1 = (g^x, H, \eta = 70)$; and generates a simulated crs and a trapdoor t for the NIZK of the equations (1). Then $\mathcal{A}_{dlin}$

gives $pk = (pk_1, g_1, g_2, crs)$ 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)) produced by using the trapdoor t. 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 [20] 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 [2] 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 (standard) 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 [2] 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 D, 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

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

The first three equations show $a = x + ys$, $b = -x_1\left(x + H\left(m \parallel u_1 \parallel u_2\right) + ys\right)$, and $c = -x_2\left(x + H\left(m \parallel u_1 \parallel u_2\right) + ys\right)$, where $x_1 = \mathrm{dlog}_{g_1} g$ and $x_2 = \mathrm{dlog}_{g_2} g$. 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 proofs 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

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

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 $\mathsf{SCUS}_{1,2}$ as anonymous signature schemes

The security notions for an anonymous signature scheme are unforgeability, anonymity, and unambiguity. The former two notions are met by SCUS_1 and SCUS_2, as seen in the previous sections. The last notion, unambiguity, intuitively ensures that if one signer releases a converter to convert a signature, then nobody else can convert that signature. We formalize the notion as follows.

Definition 7 (Unambiguity) *A scheme SCUS satisfies unambiguity if for any poly-time adversary $\mathcal{A}$,*

$$\mathbf{Adv}^{unamb}_{\mathsf{SCUS}}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr\left[\begin{array}{c}(pk_A, sk_A) \stackrel{\$}{\leftarrow} \mathsf{KeyGen}(1^\kappa), (pk_B, sk_B) \stackrel{\$}{\leftarrow} \mathsf{KeyGen}(1^\kappa) \\ (m_A, m_B, \sigma, cvt_A, cvt_B) \stackrel{\$}{\leftarrow} \mathcal{A}(pk_A, sk_A, pk_B, sk_B) \\ \mathsf{Verify}(pk_A, m_A, \sigma, cvt_A) = \mathsf{Verify}(pk_B, m_B, \sigma, cvt_B) = 1\end{array}\right]$$

is negligible in the parameter κ.

If the adversary chooses cvt_A randomly and lets $m_A = m_B$, the above definition essentially becomes that of Saraswat and Yun [36]. On the other hand, the difference with Bellare and Duan [1] is that we require the users indeed hold secret keys corresponding to their public keys (which can be done via efficient zero-knowledge proofs of knowledge). Ours is stronger than [36], weaker than [1]. It is however worth noting that since our schemes are also undeniable signature ones, requiring knowledge of valid secret keys is normal; since otherwise a signer creates a fake pair (sk', pk) (e.g., unrelated values), then all signatures become invalid with respect to pk, so the signer obviously can deny signatures he himself produced.

We now consider the schemes SCUS_1 and SCUS_2, and let the converters of the schemes be the randomness of the LE scheme.

Theorem 8 *The schemes SCUS_1 and SCUS_2 (releasing randomness for selective conversion) satisfy unambiguity, under the discrete-log assumption. In particular, for any adversary $\mathcal{A}$, there is an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{unamb}_{\mathsf{SCUS}_{1,2}}(\mathcal{A}) \leq \mathbf{Adv}^{dlog}_{G}(\mathcal{B}),$$
$$\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}).$$

The full proof is given in Appendix E, but the intuition is as follows. From the input g, h of $\mathcal{B}$, we set up the keys (pk_A, sk_A) in base g, and (pk_B, sk_B) in base h and run $\mathcal{A}$. Any ambiguity will lead to the value $\mathrm{dlog}_g(h)$, against the dlog assumption.

Acknowledgements

We thank Dennis Hofheinz for communicating on the strong uf-cma security of the GBM scheme. Many thanks also go to Laila El Aimani, Jacob Schuldt, and

Ryo Kikuchi for fruitful discussions, which sharpened the knowledge of the first author on the topic. We are indebted to the anonymous reviewers for comprehensive comments. Parts of this work was done while the first author was at Tokyo Institute of Technology with a MEXT scholarship.

References

1. M. Bellare and S. Duan. New definitions and designs for anonymous signatures. Cryptology ePrint Archive, Report 2009/336, 2009. `http://eprint.iacr.org/`.
2. D. Boneh and X. Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.
3. 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.
4. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. D. Chaum. Zero-knowledge undeniable signatures. In *EUROCRYPT*, pages 458–464, 1990.
11. 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.
12. D. Chaum and T. P. Pedersen. Wallet databases with observers. In Brickell [7], pages 89–105.
13. I. Damgård and T. P. Pedersen. New convertible undeniable signature schemes. In *EUROCRYPT*, pages 372–386, 1996.
14. L. El Aimani. Toward a generic construction of universally convertible undeniable signatures from pairing-based signatures. In D. R. Chowdhury, V. Rijmen, and A. Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pages 145–157. Springer, 2008.
15. L. El Aimani. Efficient confirmer signatures from the "signature of a commitment" paradigm. Cryptology ePrint Archive, Report 2009/435, 2009. `http://eprint.iacr.org/`.

16. L. El Aimani. On generic constructions of designated confirmer signatures. In B. K. Roy and N. Sendrier, editors, *INDOCRYPT*, volume 5922 of *Lecture Notes in Computer Science*, pages 343–362. Springer, 2009. Full version available at `http://eprint.iacr.org/2009/403`.
17. L. El Aimani. Toward a generic construction of convertible undeniable signatures from pairing-based signatures. Cryptology ePrint Archive, Report 2009/362, 2009. `http://eprint.iacr.org/`.
18. M. Fischlin. Anonymous signatures made easy. In T. Okamoto and X. Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2007.
19. 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.
20. O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.
21. 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.
22. 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.
23. Q. Huang and D. S. Wong. New constructions of convertible undeniable signature schemes without random oracles. Cryptology ePrint Archive, Report 2009/517, 2009. `http://eprint.iacr.org/`.
24. 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.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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.

32. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Brickell [7], pages 31–53.
33. L. T. Phong, K. Kurosawa, and W. Ogata. New rsa-based (selectively) convertible undeniable signature schemes. In B. Preneel, editor, *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 116–134. Springer, 2009.
34. L. T. Phong, K. Kurosawa, and W. Ogata. Provably secure convertible undeniable signatures with unambiguity. Cryptology ePrint Archive, Report 2009/394, 2009. `http://eprint.iacr.org/`. Full version of this paper.
35. D. Pointcheval. Self-scrambling anonymizers. In Y. Frankel, editor, *Financial Cryptography*, volume 1962 of *Lecture Notes in Computer Science*, pages 259–275. Springer, 2000.
36. V. Saraswat and A. Yun. Anonymous signatures revisited. In J. Pieprzyk and F. Zhang, editors, *ProvSec*, volume 5848 of *Lecture Notes in Computer Science*, pages 140–153. Springer, 2009.
37. C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.
38. J. C. N. Schuldt and K. Matsuura. An efficient convertible undeniable signature scheme with delegatable verification. Cryptology ePrint Archive, Report 2009/454, 2009. `http://eprint.iacr.org/`.
39. G. Yang, D. S. Wong, X. Deng, and H. Wang. Anonymous signature schemes. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, *Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 347–363. Springer, 2006.
40. 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.
41. R. Zhang and H. Imai. Strong anonymous signatures. In M. Yung, P. Liu, and D. Lin, editors, *Inscrypt*, volume 5487 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2008.

A A flaw in [40]

We first show that the scheme of Yuen et al [40] 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 [40] does not satisfy invisibility in the sense of Definition 3.

However, Yuen et al [40] 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 [40]) 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 unfortunately 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 [40] is totally insecure; while if the weaker definition is used, then the invisibility proof provided in [40] is incorrect.

In the full version of [40], Yuen et al have totally revised their scheme, which is based on the CDH and DLIN assumptions. However, the scheme is not as efficient as ours, let alone seems hard to meet unambiguity.

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 [21], but here we follows the exposition of Camenisch, Chandran and Shoup [8] (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 must be honestly generated, and can be kept in the public key of the signer, is generated as follows: $\gamma_1, \gamma_2, \gamma_3 \xleftarrow{\$} G$ and $\boldsymbol{\gamma} = (\gamma_0, \gamma_0', \gamma_0'') \xleftarrow{\$} G^3$. Let the common reference string be $crs = (\gamma_1, \gamma_2, \gamma_3, \boldsymbol{\gamma})$, and define vectors $\boldsymbol{\gamma_1} = (\gamma_1, 1, \gamma_3)$, $\boldsymbol{\gamma_2} = (1, \gamma_2, \gamma_3)$.

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

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

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

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

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

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

Derived from [8], the NIZK proof has perfect completeness, statistical soundness, and computational zero-knowledge (based on the decision linear assumption). The zero-knowledge is computational since a simulated crs is needed, and is created as follows: $\boldsymbol{\gamma_1}$ and $\boldsymbol{\gamma_2}$ are generated as above, but $\boldsymbol{\gamma} = \boldsymbol{\gamma_1}^{t_1}\boldsymbol{\gamma_2}^{t_2}$ for trapdoor $t = (t_1, t_2)$.

C Proof of Theorem 5

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 [20] 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.

D Security of SCUS_2

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

Theorem 9 (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

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

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 10 (Strong invisibility) *The SCUS_2 scheme satisfies strong 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.

E Unambiguity of $\mathsf{SCUS}_{1,2}$

We begin to show unambiguity for the scheme SCUS_2 (choosing to release LE randomness as converter) by proving

$$\mathbf{Adv}^{unamb}_{\mathsf{SCUS}_2}(\mathcal{A}) \leq \mathbf{Adv}^{dlog}_{G,\mathbb{PG}}(\mathcal{B}),$$
$$\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}).$$

Given $\mathcal{A}$ against unambiguity of SCUS_2, we build $\mathcal{B}$ against the dlog assumption on G of $\mathbb{PG}$. The adversary $\mathcal{B}$ gets $(g, h) \in G^2$ and the description of the pairing group $\mathbb{PG}$ as input, and needs to output $\mathrm{dlog}_g(h)$. Using the generator g and $\mathbb{PG}$, $\mathcal{B}$ sets up (pk_A, sk_A) for user A where the value g_0 of the Boneh-Boyen signature scheme is set to g^a for $a \xleftarrow{\$} Z_q$. It does the same for (pk_B, sk_B) except that the value g_0 of the Boneh-Boyen signature scheme is set to h.

The adversary $\mathcal{B}$ runs $\mathcal{A}$ on input $(pk_A, sk_A, pk_B, sk_B, \mathbb{PG})$. $\mathcal{A}$ returns the tuple $(m_A, m_B, \sigma, cvt_A, cvt_B)$, where $\sigma = (s, u_1, u_2, u_3)$, the converters $cvt_A = (r_{1A}, r_{2A})$ and $cvt_B = (r_{1B}, r_{2B})$ satisfying

$$u_3 = g^{\frac{a}{x_A + H_A(m_A \| u_1 \| u_2) + y_A s}} \cdot g^{r_{1A} + r_{2A}}$$
$$u_3 = h^{\frac{1}{x_B + H_B(m_B \| u_1 \| u_2) + y_B s}} \cdot g^{r_{1B} + r_{2B}}$$

The values (x_A, y_A) and (x_B, y_B) are respectively in sk_A and sk_B, set up by $\mathcal{B}$. The above equations are thanks to $\mathsf{Verify}(pk_A, m_A, \sigma, cvt_A) = \mathsf{Verify}(pk_A, m_A, \sigma, cvt_A) = 1$. Note that we have the Boneh-Boyen signatures in base g in the first equation and h in the second one. From the above equations, it is clear that $\mathcal{B}$ can compute $\mathrm{dlog}_g(h)$, ending the proof for SCUS_2.

We proceed with unambiguity of SCUS_1. Similarly with the above, we have the equations

$$u_3 = H_A(\overline{m})^{\frac{1}{x_A + s}} \cdot g^{r_{1A} + r_{2A}}$$
$$u_3 = H_B(\overline{m})^{\frac{1}{x_B + s}} \cdot g^{r_{1B} + r_{2B}}$$

Note that now H_A, H_B are not arbitrary, but specific hash functions, given as $H_Y(X) = h_0 \Pi_{i=1}^{160} h_i^{\mathsf{hash}(X)[i]}$ for $Y \in \{A, B\}$, $h_0, \ldots, h_{160} \in G$ and collision-resistant $\mathsf{hash} : \{0,1\}^* \to \{0,1\}^{160}$, where $\mathsf{hash}(X)[i]$ denotes the i-th bit of the hash value. Again, the idea is to set up the base g for H_A and the base h for H_B, which can be easily done by the adversary $\mathcal{B}$. We omit further details.

It is interesting to ask whether our schemes with NIZK converters satisfy unambiguity or not. They seem to meet the notion, but we unfortunately cannot prove, so leaving it as an open problem.