New Constructions of Convertible Undeniable Signature Schemes without Random Oracles

Qiong Huang[*] Duncan S. Wong[*]

Abstract

In Undeniable Signature, a signature's validity can only be confirmed or disavowed with the help of an alleged signer via a confirmation or disavowal protocol. A Convertible undeniable signature further allows the signer to release some additional information which can make an undeniable signature become publicly verifiable. In this work we introduce a new kind of attacks, called *claimability attacks*, in which a dishonest/malicious signer both disavows a signature via the disavowal protocol and confirms it via selective conversion. Conventional security requirement does not capture the claimability attacks. We show that some convertible undeniable signature schemes are vulnerable to this kind of attacks.

We then propose a new efficient construction of fully functional convertible undeniable signature, which supports both selective conversion and universal conversion, and is immune to the claimability attacks. To the best of our knowledge, it is the most efficient convertible undeniable signature scheme with provable security in the standard model. A signature is comprised of three elements of a bilinear group. Both the selective converter of a signature and the universal converter consist of one group element only. Besides, the confirmation and disavowal protocols are also very simple and efficient. Furthermore, the scheme can be extended to support additional features which include the delegation of conversion and confirmation/disavowal, threshold conversion and etc.

We also propose an alternative generic construction of convertible undeniable signature schemes. Unlike the conventional sign-then-encrypt paradigm, the signer encrypts its (standard) signature with an identity-based encryption instead of a public key encryption. It enjoys the advantage of short selective converter, which is simply an identity-based user private key, and security against claimability attacks.

Keywords: convertible undeniable signature, standard model, signature scheme, strong Diffie-Hellman assumption, identity-based encryption

[*] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong S.A.R., China. {`csqhuang@student.cityu.edu.hk, duncan@cityu.edu.hk`}.

Contents

1 Introduction 1

2 Related Work 3

3 Convertible Undeniable Signature 4

4 Assumptions 6

5 Our Proposed Scheme 7

5.1 The Scheme 7

5.2 Security Analysis 9

5.3 Efficiency and Comparison 10

6 Extensions 11

7 An Alternative Generic Construction 12

8 Conclusion 14

A Proof of Theorem 5.1 16

B Proof of Theorem 5.2 20

C Proof of Theorem 7.1 22

D Proof of Theorem 7.2 23

E Security of DHSDH Assumption in Generic Bilinear Groups 24

1 Introduction

Digital signature is publicly verifiable but also easy to copy. Anyone can forward a signer's signature to someone else and convince the one about the ownership of the signature. In some scenarios, such as software purchase [11, 6] and e-payment [7], this may not be desirable. In 1989, Chaum and van Antwerpen [13] introduced the notion of Undeniable Signature (US). Unlike conventional digital signature, an undeniable signature is not self-authenticating. If an alleged signer confirms (resp. disavows) the ownership of an undeniable signature, the signer should convince a verifier about the fact via a confirmation (resp. disavowal) protocol. The signer neither can disavow a valid signature nor confirm an invalid one. A US scheme should also be *unforgeable* and *invisible*, that is, no one but the signer can produce valid signatures, and without the help of the signer, no one can tell if a given signature is valid or not, respectively.

Convertible Undeniable Signature (CUS), first proposed by Boyar et al. [6] in 1990, has an additional property introduced to US. After generating an undeniable signature σ, the signer can release an additional piece of information, called *converter*, which makes σ publicly verifiable. There are two types of conversion: *selective conversion* transforms an individual undeniable signature σ to a publicly verifiable one; and *universal conversion* converts all signatures that have been or will be generated by the signer to publicly verifiable ones.

A typical approach of constructing a CUS scheme is based on standard signature and public key encryption (PKE). By this approach, the signer generates a signature, then encrypts it using the PKE, and the ciphertext is treated as the US signature. To confirm/disavow, the signer convinces a verifier that the ciphertext contains a valid/invalid signature. The selective converter of a US signature contains either the non-interactive version of the confirmation protocol, or the signer's standard signature associated with the ciphertext; and the universal converter is the secret key of the PKE. This is known as the '*sign-then-encrypt*' paradigm.

Our Contributions. In this work we introduce a new kind of attacks into the context of CUS, which we call '*claimability attacks*'. The conventional security definition for CUS requires that the signer cannot disavow a valid signature (via the disavowal protocol), nor confirm an invalid signature (via the confirmation protocol). Whereas, this definition does not exclude an issue that a malicious signer deliberately generates an undeniable signature σ which enables the signer to disavow the ownership of σ, while the signer can at the same time produce a selective converter which shows the validity of σ. At the first glance, this attack seems to contradict the conventional security requirements of CUS, however, this is not the case, because the signer does not confirm σ via the confirmation protocol, but via the selective conversion.

To see the practicability of the claimability attacks, we consider the following application. Suppose that a bidding system makes use of undeniable signatures for the sake of privacy, as the bidders do not want others to learn their identities from the signatures. In the bidding phase, each bidder sends their undeniable signature on their bid to the auctioneer. After that, the highest bidder confirms the signature by either executing the confirmation protocol with the auctioneer or sends the selective converter to it. Now Charlie wants to bid some antique online. He prepares a 'special' signature on his bid so that if he succeeds in the bidding but later regrets, he could deny the bid; while in case he still feels that the antique is worth the bid, he could confirm the signature/bid by releasing the selective converter. Clearly, this is unfair to others.

Some CUS schemes suffer from the claimability attacks. For example, consider Damgård and Pedersen's second CUS scheme [15]. A signature on message M is an ElGamal signature (r, s), and the US signature is (r, E) where E is an ElGamal encryption of s. To selectively convert, the signer simply releases s. Due to the lack of proof showing that E is indeed an encryption of s, a malicious signer can produce an ElGamal signature (r, s) and set the US signature to be (r, E') where E' is an encryption of a random s'. Obviously, the signer can disavow (r, E'), and in the meanwhile, the selective converter s validates the US signature, as (r, s) is indeed a valid ElGamal signature on M.

For the schemes in [41], the selective converter of a US signature is the non-interactive version of the confirmation protocol obtained using Groth-Sahai technique [23]. Since the non-interactive zero-knowledge proof works in the common reference string (CRS) model, if we put the CRS into the system parameter, the resulting scheme requires a trusted setup, which is not desired in practice. On the other hand, generally, if we put the CRS into the signer's public key, since the proof is zero-knowledge, there is a simulator which is able to produce a simulated CRS that is indistinguishable from real ones and its corresponding trapdoor, and use the trapdoor to produce indistinguishable proofs even for invalid statements. Therefore, the resulting CUS scheme is not secure under claimability attacks either.

There are two types of CUS schemes in the literature that seem to be invulnerable to claimability attacks. The first type consists of schemes in which the selective converter is the non-interactive version of the confirmation protocol obtained via the Fiat-Shamir heuristic, for example, [35, 21]. The conventional requirement on US schemes says that a US signature which could be disavowed by the signer, could not be confirmed via the confirmation protocol. This also holds even when the confirmation protocol is compressed using Fiat-Shamir transform. The second type consists of schemes in which the signature is (partially) encrypted by a deterministic encryption, for example, [31, 42] and the first scheme in [15] which uses Rabin encryption [43]. Given a US signature and its converter which is the signer's standard signature, anyone checks the validity of the converter by repeating the encryption. If the converter validates the US signature, the signer cannot disavow it again.

On the construction of CUS, we propose a new fully functional (i.e. support both selective and universal conversion) CUS scheme that is secure against claimability attacks. Based on the review given in Sec. 2 below and to the best of our knowledge, this scheme is the most efficient CUS scheme that is proven secure in the standard model. The generation of a signature requires only three exponentiations, and the signature contains merely three elements of a bilinear group $\mathbb{G}$. The scheme also has simple zero-knowledge confirmation/disavowal protocol. Besides, it supports both selective conversion and universal conversion, and both of the conversions involve just the release of one single group element. The unforgeability of the scheme is based on the Hidden Strong Diffie-Hellman (HSDH) assumption which was introduced by Boyen and Waters in [8], and the invisibility is based on a decisional variant of the HSDH assumption, the intractability of which is analyzed in the generic group model [46, 3].

Our scheme also has the advantage that given a selective converter, anyone can check if the converter is correctly generated from the US signature in a quite efficient way, i.e. evaluation of only two bilinear pairings. We emphasize that the simple validity checking is important for two reasons. First, the validity checking of a selective converter provides a way to resist the claimability attacks. Second, for practical issue, the checking should be as efficient as possible.

Like Gennaro-Halevi-Rabin RSA-based US scheme [21], our CUS scheme can be extended to achieve several interesting features as well, thanks to the simple structure of the signature. It supports the delegation of the capability of conversion and that of confirmation/disavowal. It also supports threshold conversion. The capability of conversion can be delegated to multiple delegatees so that at least certain number of them together can convert signatures. Similarly, the ability to confirm/disavow signatures can also be distributed to multiple provers. Furthermore, the scheme can be adapted to support designated verifier proofs [25] and designated confirmer signatures [12]. Readers can refer to Sec. 6 for the details.

As another contribution, we propose an alternative generic construction of CUS, which is similar to but different from the traditional 'sign-then-encrypt' paradigm. The traditional paradigm uses a PKE scheme to hide the signer's standard signature. Usually, the selective converter of a US signature is either a non-interactive proof showing that the ciphertext contains the signer's signature (thus the converter might be long), or simply the signer's standard signature. As discussed above, the resulting scheme might suffer from the claimability attacks, or is only secure in the random oracle model.

In our generic construction we replace the PKE scheme with an identity-based encryption (IBE) scheme [45, 5]. After generating a standard signature on the message, the signer then selects an identity at random and encrypts the signature for the identity under the IBE scheme. To selectively convert a US signature, the signer generates the corresponding secret key of the identity contained in the US signature. The universal converter is simply the master secret key of the IBE scheme. Note that, given a selective converter, anyone can check the validity of the US signature by first decrypting the ciphertext to obtain the signer's standard signature, and then verifying it. Besides, anyone can also check the well-formedness/correctness of the converter by randomly choosing a message, encrypting it under the identity given in the US signature, and then decrypting the ciphertext to see if the obtained message is equal to the chosen message.[1] Therefore, our approach enjoys the advantage of high efficiency in selective conversion, short converters and non-claimability. Moreover, we do not require the signer to store any information used in the signature generation.

Outline. We review some related work in the next section, and describe the formal definition of CUS and its security model in Sec. 3. In Sec. 4 we give the number-theoretic assumptions used in the concrete construction of CUS, which is proposed in Sec. 5. The security of the scheme is also analyzed there. We discuss about several extensions of our scheme in Sec. 6. The alternative generic construction of CUS is proposed in Sec. 7. Finally, the paper is concluded in Sec. 8.

2 Related Work

Since the introduction of US, it has attracted the attention of many researchers, and there has been a lot of work on this notion, such as [11, 12, 16, 37, 19, 18, 33, 27, 26, 32, 30, 29, 36]. Most of the schemes are only secure in the random oracle model. For example, Chaum proposed a US scheme [11] in 1990 and its unforgeability proof has remained open since then until Okamoto and Pointcheval [37] in 2001 considered the security of the full domain hash (FDH) variant of Chaum's scheme in the random oracle model, and Ogata, Kurosawa and Heng [36] in 2006 showed that the security of the FDH variant of Chaum's scheme with non-interactive zero-knowledge confirmation/disavowal protocols is equivalent to the Computational Diffie-Hellman (CDH) problem. The first US scheme in the standard model is due to Laguillaumie and Vergnaud [32], which is based on Boneh-Boyen short signature [3] with the bilinear groups being replaced by an ordinary group.

In the line of CUS, Boyar et al. [6] theoretically constructed a CUS scheme from the one-way function. They also proposed the first practical CUS scheme using the ElGamal signature scheme [17]. The scheme was later broken by Michels, Petersen and Horster [34]. Michels et al. also proposed an improved scheme, but without giving a security proof. In [35], Michels and Stadler proposed a CUS scheme based on Schnorr's signature scheme [44], and proved its security in the random oracle model. Damgård and Pedersen [15] proposed another two CUS schemes based on ElGamal signature. In one scheme the ElGamal signature is encrypted under Rabin encryption [43]; the other one is encrypted under ElGamal encryption [17]. However, it is unknown if these schemes are provably invisible.

Gennaro, Krawczyk and Rabin [21] proposed the first RSA-based convertible undeniable signature scheme, the unforgeability of which is based on the hardness of forging a regular RSA signature. The universal conversion of their scheme is done by releasing the public key of the regular RSA signature scheme and thus is efficient. The selective conversion is a signature of proof of knowledge obtained from a 3-move confirmation protocol by applying the Fiat-Shamir heuristic. Therefore, the security is only retained in the random oracle model. They also proposed several extensions of their scheme, i.e. delegation of confirmation and disavowal, distributed provers and signers, designated verifier and designated confirmer.

Kurosawa and Takagi [31] also presented two efficient RSA-based CUS schemes, KT_0 and KT_1, where KT_0 is secure in the random oracle model, and KT_1 is secure in the standard model. Though both of the schemes have direct selective conversion and short converter, they do not support universal conversion.

[1] This is similar to the transform from IBE scheme to signature scheme observed by Naor [5].

KT_1 was recently shown to be visible by Phong, Kurosawa and Ogata [42]. Phong et al. also proposed three other RSA-based CUS schemes: $\mathbf{SCUS}_0$, $\mathbf{SCUS}_1$ and $\mathbf{SCUS}_2$, where $\mathbf{SCUS}_0$ is secure in the random oracle model, while the other two are secure in the standard model. Both of $\mathbf{SCUS}_1$ and $\mathbf{SCUS}_2$ are instantiaitons of the 'sign-then-encrypt' paradigm. $\mathbf{SCUS}_1$ uses Generic RSA signature [24] and Paillier encryption [38], while $\mathbf{SCUS}_2$ uses Gennaro-Halevi-Rabin signature [20] and Paillier encryption. The signature sizes are four times as big as the one generated by our concrete scheme and the converters are six times that of ours for reaching the same level of security (see Sec. 5.3 for details).

Very recently, Phong, Kurosawa and Ogata [41] proposed another two discrete logarithm based constructions of CUS, SCUS_1 and SCUS_2, which instantiate the 'sign-then-encrypt' paradigm in the standard model with the Generic Bilinear Mapping (GBM) signature [24]/Boneh-Boyen fully secure signature [3] and the linear encryption [4]. The selective converter of a US signature in their schemes is the non-interactive version of the confirmation protocol obtained using Groth-Sahai technique [23], thus the converter is relatively large in size. The signature sizes of their schemes are 13% and 33% larger than that of our scheme respectively. The universal converters and the selective converters are two times and thirteen times that of ours respectively. Moreover, as discussed before, their schemes are vulnerable to the claimability attacks.

3 Convertible Undeniable Signature

Here we give the formal definition of convertible undeniable signature scheme, which consists of five (probabilistic) polynomial-time algorithms and two interactive protocols.

Definition 3.1 (Convertible Undeniable Signature)**.** *A* convertible undeniable signature *(CUS) scheme* $\mathsf{US} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{SConv}, \mathsf{UConv}, \mathsf{Ver}, \mathsf{Confirm}, \mathsf{Disavow})$ *consists of the following algorithms and protocols.*

- Kg*: takes as input 1^k where k is the security parameter, and outputs a public/secret key pair for a signer, i.e.* $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathsf{Kg}(1^k)$.
- Sign*: takes as input the signer's secret key* $\mathtt{sk}$ *and a message M, and outputs a signature σ, i.e.* $\sigma \leftarrow \mathsf{Sign}(\mathtt{sk}, M)$.
- UConv*: takes as input the signer's secret key* $\mathtt{sk}$*, and outputs a universal converter* ucvt*, i.e.* $\mathsf{ucvt} \leftarrow \mathsf{UConv}(\mathtt{sk})$.
- SConv*: takes as input a signer's secret key* $\mathtt{sk}$*, a message M and an alleged signature σ, and outputs a converter* cvt *if σ is a valid signature on M, or $\perp$ otherwise, i.e.* $\mathsf{cvt}/\perp \leftarrow \mathsf{SConv}(\mathtt{sk}, M, \sigma)$.
- Ver*: takes as input the signer's public key* $\mathtt{pk}$*, a message M, an alleged signature σ and a converter* cvt*, and outputs a bit b, which is 1 for acceptance and 0 for rejection, i.e.* $b \leftarrow \mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt})$*. We say that σ is a* valid *signature on M under* $\mathtt{pk}$ *if there exists a converter* cvt *such that the* Ver *algorithm outputs* 1.

– $\mathsf{Confirm}$*: is an interactive protocol run between the signer and a verifier on common input* $(\mathtt{pk}, M, \sigma)$*. The signer with private input* $\mathtt{sk}$ *proves to the verifier that σ is a valid signature on M under* $\mathtt{pk}$*, and the verifier outputs a bit b which is one for acceptance and zero for rejection. We denote it by* $b \leftarrow \mathsf{Confirm}_{S(\mathtt{sk}),V}(\mathtt{pk}, M, \sigma)$.

– $\mathsf{Disavow}$*: is an interactive protocol run between the signer and a verifier on common input* $(\mathtt{pk}, M, \sigma)$*. The signer with private input* $\mathtt{sk}$ *proves to the verifier that σ is an invalid signature on M under* $\mathtt{pk}$*, and the verifier outputs a bit b which is one for acceptance and zero for rejection. We denote it by* $b \leftarrow \mathsf{Disavow}_{S(\mathtt{sk}),V}(\mathtt{pk}, M, \sigma)$.

Remark 1 : The definition of SConv above imposes a check on the validity of the input message-signature pair, and returns $\perp$ if it is invalid. We stress that this requirement is not compulsory, and we do not explicitly do the validity check when describing the SConv algorithms of the proposed schemes. Previous schemes in the literature only focus on the selective conversion of *valid* signatures, i.e. [6, 1, 42], by compressing the confirmation protocol into a non-interactive one. Though some of them also support selective conversion of *invalid* signatures, however, their selective conversion of invalid signature is usually achieved by compressing the disavowal protocol, thus two different verification algorithms are needed. Our scheme supports selective conversion of both valid and invalid signatures in the *same* way, thus resulting in a unified verification of converted signatures. The signer releases a piece of information so that if the signature is valid (resp. invalid), the information confirms its validity (resp. invalidity).

REMARK 2 : The definition of CUS above covers the CUS schemes in which the selective conversion does not require the signer to store any information used in the generation of signatures, as a selective converter can be derived directly from the signer's secret key and an undeniable signature. We note that this definition does not reflect how the universal converter is used for verifying signatures. Alternatively, we can re-define the SConv algorithm so that the selective converter is derived from the universal converter and the signature, i.e. $\mathsf{cvt}/\perp \leftarrow \mathsf{SConv}(\mathsf{ucvt}, M, \sigma)$, though the universal converter is usually a part of the signer's secret key. Our proposed schemes in Sec. 5 and 7 follow this new definition. However, the disadvantage of this new definition is that it cannot cover as many existing CUS schemes as possible, for instance, Gennaro-Krawczyk-Rabin scheme [21] in which the generation of a selective converter requires the knowledge of the entire secret key of the signer. Hence, we choose to use the definition above for the sake of compatibility.

The *correctness* of CUS is defined in a natural way.For $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathsf{Kg}(1^k)$, let M be a message randomly chosen from the space $\mathcal{M}$, σ' be an invalid signature on M that is randomly chosen from the signature space $\mathcal{S}$, for any $\sigma \leftarrow \mathsf{Sign}(\mathtt{sk}, M)$, $\mathsf{cvt} \leftarrow \mathsf{SConv}(\mathtt{sk}, M, \sigma)$, it holds that

$$\begin{aligned}\Pr[1 \leftarrow \mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt})] &= 1\\ \Pr[1 \leftarrow \mathsf{Confirm}_{S(\mathtt{sk}),V}(\mathtt{pk}, M, \sigma)] &= 1\\ \Pr[1 \leftarrow \mathsf{Disavow}_{S(\mathtt{sk}),V}(\mathtt{pk}, M, \sigma')] &= 1\end{aligned}$$

A secure CUS scheme should also satisfy *unforgeability* and *invisibility*, and *non-claimability*, which are defined below.

Unforgeability. The unforgeability of CUS requires that even after obtaining many signatures on messages of its own choices and interacting with the signer for proofs of the validity/invalidity of signatures, the adversary still could not produce a signature on any new message. Formally, we consider the following game, which is played between a challenger C and an adversary $\mathcal{A}$.

1. C initiates the game by preparing a public key $\mathtt{pk}$ and the corresponding universal converter ucvt, and invokes $\mathcal{A}$ on input $(\mathtt{pk}, \mathsf{ucvt})$;
2. $\mathcal{A}$ starts to issue queries for polynomially many times to the following oracles.
   - $\mathcal{O}_{\mathsf{Sign}}$: Given a message M from $\mathcal{A}$, the oracle returns a signature σ.
   - $\mathcal{O}_{\mathsf{Confirm}}$: Given a message M and an alleged signature σ, the oracle starts an execution of the $\mathsf{Confirm}$ protocol with $\mathcal{A}$ if σ is a valid signature on M under $\mathtt{pk}$, and does nothing otherwise.
   - $\mathcal{O}_{\mathsf{Disavow}}$: Given a message M and an alleged signature σ, the oracle starts an execution of the $\mathsf{Disavow}$ protocol with $\mathcal{A}$ if σ is an invalid signature on M under $\mathtt{pk}$, and does nothing otherwise.
3. Finally, $\mathcal{A}$ outputs a pair (M^*, σ^*), and wins the game if (M^*, σ^*) is a valid message-signature pair under $\mathtt{pk}$, and $\mathcal{A}$ did not query $\mathcal{O}_{\mathsf{Sign}}$ on input M^*. The advantage of $\mathcal{A}$ in the game is defined to be its success probability.

Definition 3.2 (Unforgeability)**.** *A CUS scheme is said to be* $(t, q_s, q_c, q_d, \epsilon)$-unforgeable *if there is no adversary $\mathcal{A}$ which runs in time at most t, makes at most q_s signing queries, q_c confirmation queries and q_d disavowal queries, and wins the unforgeability game above with advantage at least ϵ.*

Remark 3 : *Strong Unforgeability* can be defined similarly by changing $\mathcal{A}$'s winning condition to that (M^*, σ^*) should be different from all the message-signature pairs it ever obtained. The adversary could query $\mathcal{O}_{\mathsf{Sign}}$ on M^* provided that σ^* is different from the answer of $\mathcal{O}_{\mathsf{Sign}}$.

Invisibility. This property requires that given a message-signature pair, without any help from the signer, a verifier is not able to tell if it is a valid pair. Below is the formal definition where we consider a game played between a challenger C and an adversary $\mathcal{D}$.

1. C initiates the game, prepares a public key $\mathtt{pk}$, and gives it to $\mathcal{D}$.
2. $\mathcal{D}$ begins to issue queries to the oracles as in the unforgeability game, except that an additional oracle called $\mathcal{O}_{\mathsf{SConv}}$ is given. For this oracle, given a message M and an alleged signature σ, it returns a converter cvt if σ is valid on M under $\mathtt{pk}$, or $\perp$ otherwise.
3. $\mathcal{D}$ submits a challenge message M^*. The challenger C flips a coin b. If $b = 0$, $\mathcal{C}$ prepares a signature σ^* on M^* valid under $\mathtt{pk}$; otherwise, it randomly chooses σ^* from the signature space. In either case, C returns σ^* to $\mathcal{D}$.
4. $\mathcal{D}$ continues to issue queries as in Step 2, with the restriction that it cannot submit (M^*, σ^*) to either of oracles $\mathcal{O}_{\mathsf{SConv}}$, $\mathcal{O}_{\mathsf{Confirm}}$ and $\mathcal{O}_{\mathsf{Disavow}}$.
5. Finally, $\mathcal{D}$ outputs a bit b', and wins the game if $b' = b$. Its advantage in the game is defined to be $|\Pr[b' = b] - \frac{1}{2}|$.

Definition 3.3 (Invisibility)**.** *A CUS scheme is said to be* $(t, q_s, q_{sc}, q_c, q_d, \epsilon)$-invisible *if there is no adversary $\mathcal{D}$ which runs in time at most t, makes at most q_s signing queries, q_{sc} selective conversion queries, q_c confirmation queries and q_d disavowal queries, and wins the unforgeability game above with advantage at least ϵ.*

Remark 4 : In the rest of the paper we sometimes omit the numbers of queries the adversary makes in the games, and simply say that a CUS scheme is (t, ϵ)-(strongly) unforgeable or (t, ϵ)-invisible.

Non-Claimability. This property requires that a malicious signer is unable to produce a signature σ such that the signer can both disavow σ and generate a selective converter to confirm its validity. Formally we consider the game below, in which $\mathcal{A}$ is the malicious signer, and C is the challenger.

1. $\mathcal{A}$ takes as input 1^k and outputs $(\mathtt{pk}, M, \sigma, \mathsf{cvt})$.
2. $\mathcal{A}$ and C start an execution of the $\mathsf{Disavow}$ protocol on common input $(\mathtt{pk}, M, \sigma)$, in which $\mathcal{A}$ acts as the signer/prover and C as the verifier. Let C's output at the end of the protocol be b. $\mathcal{A}$ wins the game if $b = 1$ and $\mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt}) = 1$. The advantage of $\mathcal{A}$ is defined to be its success probability.

Definition 3.4 (Non-Claimability)**.** *A CUS scheme is said to be* (t, ϵ)-non-claimable *if there is no adversary $\mathcal{A}$ which runs in time at most t, and wins the non-claimability game above with advantage at least ϵ.*

4 Assumptions

In this section we review and define some number theoretic assumptions which will be used in our concrete construction of CUS. For simplicity, we define them in symmetric bilinear groups.

Strong Diffie-Hellman Assumption [3]. Let $\mathbb{G}$ be a multiplicative group of prime order p, and g a generator of $\mathbb{G}$. The *Strong Diffie-Hellman* (SDH) assumption is defined as follows.

Definition 4.1 (q-SDH Assumption)**.** *The* q-SDH *assumption* (t, ϵ)*-holds in* $\mathbb{G}$ *if there is no algorithm* $\mathcal{A}$ *which runs in time at most* t*, and satisfies the following condition:*

$$\Pr\left[\mathcal{A}\left(g, g^x, g^{x^2}, \cdots, g^{x^q}\right) = \left(g^{\frac{1}{x+s}}, s\right)\right] \geq \epsilon$$

where $s \in \mathbb{Z}_p$*, and the probability is taken over the random choices of* $x \in \mathbb{Z}_p$ *and the random coins used by* $\mathcal{A}$*.*

Hidden Strong Diffie-Hellman Assumption [8]. Let $\mathbb{G}$ be a multiplicative group of prime order p, and g be its generator. The *Hidden Strong Diffie-Hellman* (HSDH) assumption is defined as below:

Definition 4.2 (q-HSDH Assumption)**.** *The* q-HSDH *assumption* (t, ϵ)*-holds in* $\mathbb{G}$ *if there is no algorithm* $\mathcal{A}$ *which runs in time at most* t*, and satisfies the following condition:*

$$\Pr\left[\mathcal{A}\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}\right) = \left(g^{\frac{1}{x+s}}, g^s, g^{\beta s}\right)\right] \geq \epsilon$$

where $s \in \mathbb{Z}_p$ *and* $s \notin \{s_1, \cdots, s_q\}$*, the probability is taken over the random choices of* $x, \beta, s_1, \cdots, s_q \in \mathbb{Z}_p$ *and the random coins used by* $\mathcal{A}$*.*

We also use a decisional version of the HSDH assumption. Note that for each tuple $(A, B, C) = (g^{1/(x+s)}, g^s, u^s)$ in the HSDH problem where $u = g^\beta$, its well-formedness can be verified in bilinear groups without knowing the secret key x or the value of s, i.e. $\mathsf{e}(A, g^x B) = \mathsf{e}(g, g)$ and $\mathsf{e}(B, u) = \mathsf{e}(g, C)$. However, if we remove B from the tuple, the well-formedness of A and C cannot be checked if one does not know x or s. Below is the formal definition of the decisional HSDH assumption.

Decisional Hidden Strong Diffie-Hellman (DHSDH) Assumption. Let $\mathbb{G}$ be a multiplicative group of prime order p, and g a generator of $\mathbb{G}$. The DHSDH assumption is defined as follows.

Definition 4.3 (q-DHSDH Assumption)**.** *The* q-DHSDH *assumption* (t, ϵ)*-holds in* $\mathbb{G}$ *if there is no algorithm* $\mathcal{A}$ *which runs in time at most* t*, and satisfies the following condition:*

$$\Bigg|\Pr\left[\mathcal{A}\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}, g^{\frac{1}{x+s}}\right) = 1\right] - \Pr\left[\mathcal{A}\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}, Z\right) = 1\right]\Bigg| \geq \epsilon$$

where the probability is taken over the random choices of $x, \beta, s_1, \cdots, s_q, s \in \mathbb{Z}_p$ *and* $Z \in \mathbb{G}$*, and the random coins used by* $\mathcal{A}$*.*

In Appendix E we analyze the intractability of the DHSDH assumption in the generic bilinear group model, where we show that an adversary that solves the q-DHSDH problem with a constant advantage $\epsilon > 0$ in generic groups of order p such that $q < o(\sqrt[3]{p})$, requires $\Omega(\sqrt{\epsilon p/q})$ generic group operations.

5 Our Proposed Scheme

5.1 The Scheme

Our concrete scheme is based on the Generic Bilinear Map (GBM) signature scheme [24]. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative groups of large prime order p, and g be a generator of $\mathbb{G}$. Let $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an admissible pairing. Let $n = n(k)$ and $\eta = \eta(k)$ be two arbitrary positive polynomials. Let $\mathcal{M} := \{0, 1\}^n$ be the message space (otherwise we can use a collision-resistant hash function to map arbitrarily long messages to n-bit strings), and $\mathtt{H} = (\mathsf{PHF.Gen}, \mathsf{PHF.Eval})$ be a programmable hash function from $\mathcal{M}$ to $\mathbb{G}$ [24]. In the following we write $\mathtt{H}_\kappa(M) = \mathsf{PHF.Eval}(\kappa, M)$. A signature in the

```
$\mathsf{Kg}(1^k)$:
    $\kappa \leftarrow_\$ \mathsf{PHF.Gen}(1^k)$
    $x, y \leftarrow_\$ \mathbb{Z}_p$, $u \leftarrow_\$ \mathbb{G}$
    $X \leftarrow g^x$, $Y \leftarrow g^{1/y}$
    return $(\mathtt{pk}, \mathtt{sk}) := ((g, X, Y, u, \kappa), (x, y))$
```

```
$\mathsf{Sign}(\mathtt{sk}, M)$:
    parse $\mathtt{sk}$ as $(x, y)$
    $s \leftarrow_\$ \mathbb{Z}_p$
    $\delta \leftarrow \mathtt{H}_\kappa(M)^{1/(x+s)}$, $\gamma \leftarrow Y^s$, $\theta \leftarrow u^s$
    return $\sigma := (\delta, \gamma, \theta)$
```

```
$\mathsf{UConv}(\mathtt{sk})$:
    parse $\mathtt{sk}$ as $(x, y)$
    return $\mathsf{ucvt} := y$
```

```
$\mathsf{SConv}(\mathtt{sk}, M, \sigma)$:
    parse $\mathtt{sk}$ as $(x, y)$
    $\nu \leftarrow \gamma^y$
    return $\mathsf{cvt} := \nu$
```

```
$\mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt})$:
    parse $\mathtt{pk}$ as $(g, X, Y, u, \kappa)$
    parse $\sigma$ as $(\delta, \gamma, \theta)$, $\mathsf{cvt}$ as $\nu$
    $b_1 \leftarrow \mathsf{e}(\delta, X \cdot \nu) \overset{?}{=} \mathsf{e}(\mathtt{H}_\kappa(M), g)$
    $b_2 \leftarrow \mathsf{e}(\nu, u) \overset{?}{=} \mathsf{e}(g, \theta)$
    return $b_1 \wedge b_2$
```

Figure 1: A Concrete Construction of CUS, US_{GBM}

GBM scheme is of the form $\sigma = (\mathtt{H}_\kappa(M)^{1/(x+s)}, s)$ where $x \in \mathbb{Z}_p^*$ is the secret key and s is a random element of $\{0,1\}^\eta$. The validity of $\sigma = (\sigma_1, \sigma_2)$ can be verified by checking if $\mathsf{e}(\mathtt{H}_\kappa(M), g) = \mathsf{e}(\sigma_1, g^x g^{\sigma_2})$. Based on GBM scheme, we propose a CUS scheme US_{GBM} (Fig. 1), where we assume that all the users in the system share the same system parameter, i.e. $(\mathbb{G}, \mathbb{G}_T, \mathsf{e}, p, g)$.

Note that given the universal converter $\mathsf{ucvt} = y$, anyone can check its validity by $g = Y^y$, and can generate the corresponding converter for any signature, because the selective conversion only requires the knowledge of y. An undeniable signature in US_{GBM} is of the form $(\delta, \gamma, \theta) = (\mathtt{H}_\kappa(M)^{1/(x+s)}, Y^s, u^s)$ and a converted signature is of the form $(\delta, \gamma, \theta, \nu) = ((\mathtt{H}_\kappa(M)^{1/(x+s)}, Y^s, u^s, g^s)$. In fact, one can view (δ, ν, θ) as the signer's self-authenticating signature due to its public verifiability. On the other hand, given a signature $\sigma = (\delta, \gamma, \theta)$ and a converter ν, one can verify the validity of ν by checking if (Y, g, γ, ν) is a DH-tuple, i.e. $\mathsf{e}(Y, \nu) = \mathsf{e}(\gamma, g)$, which serves as an NIZK proof of knowledge of the secret y, and thus shows the correctness of the selective conversion. Suppose ν is a valid converter of σ. If σ is a valid undeniable signature, ν confirms its validity; if it is invalid, ν confirms its invalidity. Therefore, our scheme supports an efficient and unified conversion of both valid and *invalid* signatures.

Signature Space. The signature space $\mathcal{S}$ of US_{GBM} with respect to the public key (g, X, Y, u, κ) is defined as

$$\mathcal{S} := \left\{(\delta, \gamma, \theta) \in \mathbb{G}^3 \ : \ \mathsf{e}(Y, \theta) = \mathsf{e}(\gamma, u)\right\}$$

and the converted signature space $\mathcal{S}'$ is defined as

$$\mathcal{S}' := \left\{(\delta, \gamma, \theta, \nu) \in \mathbb{G}^4 \ : \ (\delta, \gamma, \theta) \in \mathcal{S} \wedge \mathsf{e}(Y, \nu) = \mathsf{e}(\gamma, g)\right\}$$

Confirmation/Disavowal Protocol. Given a message M and a corresponding undeniable signature $\sigma = (\delta, \gamma, \theta)$, both the signer S and the verifier V check if $\sigma \in \mathcal{S}$. If not, they do nothing; otherwise, the signer computes the converter for the signature, i.e. $\mathsf{cvt} := \nu \leftarrow \gamma^y$. Note that from ν, the signature can be verified by checking if

$$\mathsf{e}(\mathtt{H}_\kappa(M), g) = \mathsf{e}(\delta, X \cdot \nu) \tag{1}$$

If equation (1) holds, S and V start an execution of the Confirm protocol; otherwise, they start an execution of the Disavow protocol.

Confirm. Note that equation (1) is equivalent to

$$\mathsf{e}(\delta, \gamma)^y = \mathsf{e}(\mathtt{H}_\kappa(M), g) \cdot \mathsf{e}(\delta, X)^{-1} \tag{2}$$

where only y is unknown to the verifier. Now from the signer's public key, we have that

$$g = Y^y \tag{3}$$

Therefore, to confirm a signature, it is sufficient for the signer to make a proof of equal discrete logarithm, i.e.

$$\log_Y(g) = \log_{\mathsf{e}(\delta,\gamma)} \left(\mathsf{e}(\mathtt{H}_\kappa(M), g) \cdot \mathsf{e}(\delta, X)^{-1}\right) \tag{4}$$

Disavow. If σ is invalid, equation (2) does not hold. However, equation (3) holds no matter if σ is valid or not. Therefore, to disavow a signature, it is sufficient for the signer to make the following proof.

$$\log_Y(g) \neq \log_{\mathsf{e}(\delta,\gamma)} \left(\mathsf{e}(\mathtt{H}_\kappa(M), g) \cdot \mathsf{e}(\delta, X)^{-1}\right) \tag{5}$$

Remark 5 : The left side of equations (4) and (5) works in group $\mathbb{G}$, while the right side works in group $\mathbb{G}_T$. It is easy to resolve this 'incompatibility', say, by changing the left side to $\log_{\mathsf{e}(g,Y)} \mathsf{e}(g, g)$.

Remark 6 : There are standard (3-move) *special honest-verifier zero-knowledge* protocols for the tasks above, e.g. [9, 10], and there are also known ways to transform them into 4-move perfect zero-knowledge proofs of knowledge *in general* with *negligible soundness error*, e.g. [14], so that there exists a probabilistic polynomial-time simulator that produces indistinguishable views of any verifier. In addition, it is easy to see that our scheme has the advantage that the signer does *not* need to remember any signature it ever produced in order to selectively convert, confirm or disavow a signature. This is an important feature for practical use.

5.2 Security Analysis

Theorem 5.1. *Let* $\mathtt{H}$ *be a* $(m, 1, \phi, \varphi)$*-programmable hash function. Let* $\mathcal{F}$ *be a* $(t, q_s, q_c, q_d, \epsilon)$*-forger in the unforgeability game of* US_{GBM}*. Then there exists an adversary* $\mathcal{A}_1$ *that* (t_1, ϵ_1)*-breaks the* q_s*-SDH assumption with*

$$t_1 \approx t \quad \textit{and} \quad \epsilon_1 \geq \frac{\varphi}{q_s}\left(\epsilon - \frac{q_s^{m+1}}{p^m} - \phi\right),$$

or there exists an adversary $\mathcal{A}_2$ *that* (t_2, ϵ_2)*-breaks the* q_s*-HSDH assumption and an adversary* $\mathcal{A}_3$ *that* (t_3, ϵ_3)*-breaks the Discrete Logarithm assumption in* $\mathbb{G}$ *with*

$$t_2, t_3 \approx t \quad \textit{and} \quad \epsilon_2 + \epsilon_3 \geq \epsilon - \phi$$

The proof basically follows that of Theorem 4.2 in [24], except that the component s in a signature is replaced with Y^s and u^s, and that now $\mathcal{A}$ has to handle the confirmation/disavowal requests. Note that all the oracles other than $\mathcal{O}_{\mathsf{Sign}}$ can be perfectly simulated by $\mathcal{A}$ using its knowledge of y, and that since the confirmation and disavowal protocol of US_{GBM} only involve the knowledge of y, which acts as the universal converter, the confirmation oracle and disavowal oracle become useless to the adversary. We defer the proof to Appendix A.

Remark 7 : Theorem 5.1 establishes the existential unforgeability of US_{GBM} under chosen message attacks. Furthermore, we can use the same proof to show that US_{GBM} is *strongly unforgeable.* Note that in the proof of the theorem, we only consider if s collides with any s_j and do not care if M is the same as any M_j. The only place where we need to take care of is in Game 6 of Type 1 in the case that M is equal to M_l for some $1 \leq l \leq q_s$. Since $M = M_l$, by the requirement of winning the game, it must be that $s \neq s_l$. Therefore, in Game 6 of Type 1, the adversary's choice of l must not fall into the set of indices j with $\gamma_j = \gamma_i$ and $\theta_j = \theta_i$ (thus $s_j = s_i$); otherwise, we have that $s \neq s_l = s_i = s$, which is a contradiction. Hence, the probability that we raise the event $\mathsf{abort}_{\mathsf{bad.a}}$ remains unchanged.

Theorem 5.2. *Let* $\mathtt{H}$ *be a* $(m, 1, \phi, \varphi)$*-programmable hash function. Let* $\mathcal{D}$ *be a* $(t, q_s, q_{sc}, q_c, q_d, \epsilon)$*-distinguisher in the invisibility game of* US_{GBM}*. Assume that* US_{GBM} *is* $(t_1, q_s, q_{sc}, q_c, q_d, \epsilon_1)$*-strongly unforgeable, the confirmation (resp. disavowal) protocol is* ϵ_2*-zero-knowledge* [2] *(resp.* ϵ_3*-zero-knowledge).*

[2] We say that a proof system is ϵ-zero-knowledge, if there exists a probabilistic polynomial-time simulator that given oracle access to any (malicious) verifier V^*, outputs a view of V^* such that there is no probabilistic polynomial-time distinguisher which tells the simulated view apart from the view of V^* interacting with a real prover with probability at least $1/2 + \epsilon$. We say that the proof system is *perfect zero-knowledge* if $\epsilon = 0$.

Then there exists an adversary $\mathcal{A}$ which (t', ϵ')-breaks the (q_s+1)-DHSDH assumption and an adversary $\mathcal{A}'$ which (t'', ϵ'')-breaks the Discrete Logarithm assumption with

$$t_1, t', t'' \approx t \quad \textit{and} \quad \epsilon' + \epsilon'' \geq \epsilon - \varphi - \epsilon_1 - q_c \cdot \epsilon_2 - q_d \cdot \epsilon_3$$

The proof is deferred to Appendix B.

Theorem 5.3. *Suppose that* Disavow *Protocol is (t, ϵ)-sound* [3]. *Then* US_{GBM} *is (t', ϵ')-non-claimable, where*

$$t' \approx t \quad \textit{and} \quad \epsilon' \leq \epsilon.$$

Proof. Let $\mathcal{A}$ be an adversary against the non-claimability, and let $(\mathtt{pk}, M, \sigma, \mathsf{cvt})$ be its output in the game, where $\mathtt{pk} = (g, X, Y, u, \kappa)$, $\sigma = (\delta, \gamma, \theta)$ and $\mathsf{cvt} = \nu$. Suppose that $\mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt}) = 1$. We then have $\mathsf{e}(\gamma, u) = \mathsf{e}(Y, \theta)$, $\mathsf{e}(\nu, Y) = \mathsf{e}(g, \gamma)$ and $\mathsf{e}(\delta, X \cdot \nu) = \mathsf{e}(\mathtt{H}_\kappa(M), g)$, which indicates that $\gamma = Y^s$, $\theta = u^s$ and $\nu = g^s$ for some $s \in \mathbb{Z}_p^*$, and $\delta = \mathtt{H}_\kappa(M)^{1/(x+s)}$ for $x = \log_g X$. Therefore, σ is valid on M under $\mathtt{pk}$. By the soundness of the Disavow protocol, we have that with probability at most ϵ the signer can prove to an honest verifier that σ is an invalid signature via Disavow protocol. □

5.3 Efficiency and Comparison

Below we compare our scheme with some existing CUS schemes, in terms of 80-bit security. For schemes based on bilinear pairings, we choose the security parameter $k = 170$, and for those scheme based on RSA, we choose $k = 1024$. For the scheme in [35] we take the values suggested by the authors, i.e. $|p| = 1024$ and $|q| = 256$. All the sizes in Fig. 2 are in bits. By |Sig|, |SConv| and |UConv| we denote the size of a signature, size of a selective converter and size of a universal converter, respectively. 'Non-Clm' means non-claimability. A 'no' in the column of |UConv| indicates that the scheme does not support universal conversion. For the assumptions, by EDL, CDDH, CNR, DNR, dtm-RSA, SRSA, DIV, DLN we denote equal discrete logarithm assumption, composite decision Diffie-Hellman assumption, computational N-th residuosity assumption, decisional N-th residuosity assumption, decisional two moduli RSA assumption, strong RSA assumption, division intractability assumption and decisional linear assumption, respectively.

	|Sig|	|SConv|	|UConv|	Non-Clm	Assumptions	Model
[21]	1024	2048	1024	$\surd$	RSA + EDL	rom
[35]	1280	768	256	$\surd$	CDH + EDL	rom
[18]	2389	2208	1024	$\surd$	Factoring + CDDH	rom
KT$_0$ [31]	1024	1024	no	$\surd$	CNR + DNR	rom
KT$_1$ [31]	3232	1024	no	$\surd$	broken[42]	std
SCUS$_0$ [42]	1024	1024	1024	$\surd$	RSA + dtm-RSA	rom
SCUS$_1$ [42]	2128	1024	1024	$\surd$	SRSA + DNR	std
SCUS$_2$ [42]	2048	1024	1024	$\surd$	SRSA + DIV + DNR	std
SCUS$_1$ [41]	580	2210	340	$\times$	SDH + DLN	std
SCUS$_2$ [41]	680	2210	340	$\times$	SDH + DLN	std
US$_{GBM}$	510	170	170	$\surd$	HSDH + DHSDH	std

Figure 2: Comparison with other CUS schemes

From Fig. 2 we can see that our proposed scheme has the smallest signature size, shortest selective converter and shortest universal converter.

[3] Roughly, a proof system is (t, ϵ)-sound if there is no prover P^* running in time at most t, such that for any statement x outside of the language L, the probability that the verifier outputs 1 after interacting with P^* is at least ϵ.

6 Extensions

In this section we give several extensions of our CUS scheme proposed in the previous section.

Conversion Delegation. In US_{GBM}, the signer's secret key can divided into two parts, i.e. x as the signing key, and y as the conversion key. Since the selective conversion of US_{GBM} only uses y, the signer can delegate its conversion ability to someone that he trusts by sending y to him. Then the delegatee can convert any signature into a publicly verifiable one using y as the universal converter. Besides, the delegatee can confirm/disavow signatures on behalf of the signer without any further help from it, because the confirmation/disavowal protocol requires the knowledge of y only.

Designated Confirmer Signature. Introduced by Chaum [12], *designated confirmer signatures* (DCS) aim to alleviate the burden on the signer in undeniable signatures [13]. A designated party, named *the confirmer*, confirms/disavows signatures on behalf the signer without help from the signer. The discussion in the first extension demonstrates that US_{GBM} can also be slightly modified to be a DCS scheme. Namely, we remove $(Y = g^{1/y}, y)$ from the signer's key pair and set it as the confirmer's key pair. The signing algorithm, conversion algorithm, and confirmation/disavowal protocol simply follow those of US_{GBM}. In this way, we obtain a highly efficient DCS scheme that is provably secure without random oracles. On the other hand, we observe that a DCS scheme can be slightly modified to be a CUS scheme supporting conversion delegation, i.e. by putting the public key of the confirmer into that of the signer, and giving the confirmer's secret key to the delegatee.

Confirmation/Disavowal Delegation. In some applications it may be desired that a party who holds the selective converter of a valid/invalid US signature confirms/disavows the signature on behalf of the signer without releasing the converter to the verifier. Let H be a holder of the selective converter ν of a signature $\sigma = (\delta, \gamma, \theta)$ on message M. Note that the universal converter is unknown to H. To comfirm/disavow σ, H first commits to ν by randomly picking $z \in \mathbb{Z}_p$ and computing $T \leftarrow \nu \cdot \tilde{g}^z$ where $\tilde{g}$ is a random generator of $\mathbb{G}$. Note that T is perfectly hiding. By the validity of ν, we know that

$$\mathsf{e}(\nu, Y) = \mathsf{e}(\gamma, g) \quad \Rightarrow \quad \mathsf{e}(\tilde{g}, Y)^z = \mathsf{e}(T, Y) \cdot \mathsf{e}(\gamma, g)^{-1} \tag{6}$$

<u>Confirm</u>. Now assume that σ is a valid US signature on M. We have

$$\mathsf{e}(\delta, X \cdot T) = \mathsf{e}(\mathtt{H}_\kappa(M), g) \cdot \mathsf{e}(\delta, \tilde{g})^z \quad \Rightarrow \quad \mathsf{e}(\delta, \tilde{g})^z = \mathsf{e}(\delta, X \cdot T) \cdot \mathsf{e}(\mathtt{H}_\kappa(M), g)^{-1} \tag{7}$$

Therefore, by equations (6) and (7), it is sufficient for H to make a proof of equal discrete logarithm using z as the witness, showing that

$$\log_{\mathsf{e}(\tilde{g},Y)} \left(\mathsf{e}(T, Y) \cdot \mathsf{e}(\gamma, g)^{-1} \right) = \log_{\mathsf{e}(\delta,\tilde{g})} \left(\mathsf{e}(\delta, X \cdot T) \cdot \mathsf{e}(\mathtt{H}_\kappa(M), g)^{-1} \right) \tag{8}$$

<u>Disavow</u>. In the other case, i.e. σ is an invalid US signature on M, equation (7) does not hold. However, equation (6) still holds. Hence, it is sufficient for H to make a zero-knowledge proof of non-equal discrete logarithm using z as the witness, showing that equation (8) does not hold.

We stress that the conversion delegation and the confirmation/disavowal delegation are related to but different from DCS [12]. The common ground is that verifiers are sure that someone (the confirmer) can confirm/disavow signatures on behalf of the signers. However, in the conversion delegation and confirmation/disavowal delegation, anyone can act as the confirmer and is not required to have a public/secret key pair; while in DCS, the confirmer is fixed and needs to be equipped with a key pair.

Designated Verifier. The signer S can prove the validity/invalidity of a signature to a verifier via the confirmation/disavowal protocol, however, it cannot choose whom can be convinced of the fact. A verifier V could act as the intermediary between the signer and a set of verifiers. Jakobsson et al. [25] proposed the notion of *designated verifier proofs* to solve this problem, which readily applies to our scenario as well. Now V is equipped with a key pair, and S proves that either the signature is valid/invalid or it knows the secret key of V, so that V is also able to produce indistinguishable proofs.

Distributed Conversion. This is to share the ability of converting signatures to multiple parties. The signer secretly shares the conversion key y among n delegatees so that at least $t+1$ out of them together can selectively convert a US signature using their shares. This can be easily achieved by applying the t-out-of-n verifiable secret sharing scheme in [39, 40] to US_{GBM}.

Distributed Provers. Introduced by Pedersen [39], a distributed provers protocol shares the key among n provers, and only $t+1$ or more provers together can prove to a verifier that the given statement is true. Like Gennaro et al.'s RSA-based US scheme [21], Pedersen's technique [39] also easily extends to our CUS scheme to support distributed provers.

Remark 8 : To the best of our knowledge, only Gennaro et al. mentioned the similar extensions in their work [21]. However, they did not show how to extend their scheme to allow a holder of the selective converter of a signature to conform/disavow the signature. There, the converter of a signature is the non-interactive version of a three-move conformation protocol obtained using the Fiat-Shamir heuristic, thus it is unlikely for their scheme to support this feature. On the other hand, Gennaro et al.'s scheme supports distributed signers, i.e. only certain number of parties who holds a share of the signer's secret key together can sign messages on behalf of the signer, due to the simple structure of RSA signature; while it does not seem like that our scheme enjoys this feature.

As show in Sec. 5.3, the signature size of Gennaro et al.'s scheme is about two times that of ours, and the selective converter and universal converter are twelve and six times that of ours. Besides, the security of their CUS scheme is in the random oracle model, while ours is in the standard model. However, the security of our scheme relies on assumptions that are not studied as well as those of their scheme. We leave the construction of CUS schemes with comparable efficiency (i.e. comparable signature size and converter size) in the standard model based on better studied assumptions and supporting all the aforementioned extensions (including distributed signers), as our future work.

7 An Alternative Generic Construction

In this section we present an alternative generic construction of CUS, which is similar to the traditional 'sign-then-encrypt' paradigm. In our construction the signer encrypts its standard signature on the message with an identity-based encryption (IBE) scheme instead of a public key encryption scheme. Specifically, we use a *separable* IBE scheme, in the sense that the generation of a ciphertext can be divided into two parts, i.e. (C, D), where C is independent of the plaintext, and D is dependent on it. Therefore, C can be generated even before the plaintext is given. Formally, an IBE scheme $\mathsf{IBE} = (\mathsf{Kg}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ is separable if

1. The Enc algorithm is comprised of two sub-algorithms, $\mathsf{EncRand}$ which is probabilistic, and $\mathsf{EncPltx}$ which is deterministic. $\mathsf{EncRand}$ takes as input the master public key and an identity, and outputs C and some state information ω. $\mathsf{EncPltx}$ takes as input ω and the plaintext, and outputs D.
2. C and the message to be encrypted uniquely determine D. That is, given C and the message, there is only one possible D.

To the best of our knowledge, almost all the IBE schemes in the literature are separable, such as [5, 2, 47, 22].

Let $\mathsf{S} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ be a standard signature scheme, $\mathsf{IBE} = (\mathsf{Kg}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be a separable identity-based encryption scheme with (super-polynomially large) identity space $\mathcal{I}$ and $\mathtt{H} = (\mathsf{Kg}, \mathsf{Eval}, \mathsf{Trap})$ be a secure trapdoor hash function [28] with randomness space $\mathcal{R}$. Our generic construction of CUS, named US_{Gen}, is depicted in Fig. 3.

```
$\mathsf{Kg}(1^k)$:
    $(\mathtt{pk}_\mathsf{S}, \mathtt{sk}_\mathsf{S}) \leftarrow \mathsf{S.Kg}(1^k)$
    $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{IBE.Kg}(1^k)$
    $(\mathtt{pk}_\mathtt{H}, \mathtt{sk}_\mathtt{H}) \leftarrow \mathtt{H}.\mathsf{Kg}(1^k)$
    return $(\mathtt{pk}, \mathtt{sk}) := ((\mathtt{pk}_\mathsf{S}, \mathtt{pk}_\mathtt{H}, \mathtt{mpk}), (\mathtt{sk}_\mathsf{S}, \mathtt{msk}))$

$\mathsf{Sign}(\mathtt{sk}, M)$:
    parse $\mathtt{sk}$ as $(\mathtt{sk}_\mathsf{S}, \mathtt{msk})$
    $I \leftarrow_\$ \mathcal{I}$, $(C, \omega) \leftarrow \mathsf{IBE.EncRand}(\mathtt{mpk}, I)$
    $R \leftarrow_\$ \mathcal{R}$, $\overline{C} \leftarrow \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_\mathtt{H}, C, R)$
    $\delta \leftarrow \mathsf{S.Sign}(\mathtt{sk}_\mathsf{S}, M \| I \| \overline{C})$
    $D \leftarrow \mathsf{IBE.EncPltx}(\omega, \delta)$
    return $\sigma := (C, D, I, R)$

$\mathsf{UConv}(\mathtt{sk})$:
    parse $\mathtt{sk}$ as $(\mathtt{sk}_\mathsf{S}, \mathtt{msk})$
    return $\mathsf{ucvt} := \mathtt{msk}$

$\mathsf{SConv}(\mathtt{sk}, M, \sigma)$:
    parse $\mathtt{sk}$ as $(\mathtt{sk}_\mathsf{S}, \mathtt{msk})$
    parse $\sigma \rightarrow (C, D, I, R)$
    $\mathtt{sk}_I \leftarrow \mathsf{IBE.Extract}(\mathtt{msk}, I)$
    return $\mathsf{cvt} := \mathtt{sk}_I$

$\mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt})$:
    parse $\mathtt{pk}$ as $(\mathtt{pk}_\mathsf{S}, \mathtt{pk}_\mathtt{H}, \mathtt{mpk})$
    parse $\sigma$ as $(C, D, I, R)$, $\mathsf{cvt}$ as $\mathtt{sk}_I$
    $\delta \leftarrow \mathsf{IBE.Dec}(\mathtt{sk}_I, \mathtt{mpk}, (C, D))$
    $\overline{C} \leftarrow \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_\mathtt{H}, C, R)$
    return $\mathsf{S.Ver}(\mathtt{pk}_\mathsf{S}, M \| I \| \overline{C}, \delta)$
```

Figure 3: Alternative Generic Construction of Undeniable Signature, US_{Gen}

Remark 9 : One may notice that the trapdoor property of function $\mathtt{H}$ is never used in the scheme US_{Gen}. The trapdoor property is only used in the security proof, i.e. the proof of invisibility, as we shall see later.

Signature Space. Denote by $\mathcal{S}_\mathsf{IBE}$ be the ciphertext space. Then The signature space $\mathcal{S}$ of US_{Gen} is defined to be the set of all tuples of the form (C, D, I, R) where $(C, D) \in \mathcal{S}_\mathsf{IBE}$, $I \in \mathcal{I}$ and $R \in \mathcal{R}$; while the converted signature space $\mathcal{S}'$ is defined to be the set of all tuples of the form $(\sigma, \mathtt{sk}_I)$ where $\sigma \in \mathcal{S}$ and $\mathtt{sk}_I$ is in the space of user private keys in IBE.

Confirmation/Disavowal Protocol. Given a signature $\sigma = (C, D, I, R)$, the signer first computes $\mathtt{sk}_I$ as specified in the scheme using $\mathtt{msk}$, and uses it to recover δ from (C, D). It checks the validity of δ under $\mathtt{pk}_\mathsf{S}$. If it is valid, the signer confirms the validity of σ by starting an execution of a general zero-knowledge proof system showing that $(\delta, \mathtt{sk}_I, \mathtt{msk})$ is in the following NP language:

$$L_Y := \Big\{(\delta, \mathtt{sk}_I, \mathtt{msk}) \ : \ \mathtt{sk}_I = \mathsf{IBE.Extract}(\mathtt{msk}, I) \wedge \delta = \mathsf{IBE.Dec}(\mathtt{sk}_I, \mathtt{mpk}, (C, D)) \wedge \mathsf{S.Ver}(\mathtt{pk}_\mathsf{S}, \overline{M}, \delta) = 1\Big\}$$

where $\overline{M} := M \| I \| \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_\mathtt{H}, C, R)$. Otherwise, it disavows σ by starting an execution of another general zero-knowledge proof system showing that $(\delta, \mathtt{sk}_I, \mathtt{msk})$ is in the following NP language:

$$L_N := \Big\{(\delta, \mathtt{sk}_I, \mathtt{msk}) \ : \ \mathtt{sk}_I = \mathsf{IBE.Extract}(\mathtt{msk}, I) \wedge \delta = \mathsf{IBE.Dec}(\mathtt{sk}_I, \mathtt{mpk}, (C, D)) \wedge \mathsf{S.Ver}(\mathtt{pk}_\mathsf{S}, \overline{M}, \delta) = 0\Big\}$$

Theorem 7.1. *Let $\mathcal{A}$ be an adversary that $(t, q_s, q_c, q_d, \epsilon)$-breaks the strong unforgeability of US_{Gen}. Then there exists another adversary $\mathcal{B}$ that (t', q_s, ϵ')-breaks the strong unforgeability of S and an algorithm $\mathcal{B}'$ that (t'', ϵ'')-breaks the collision resistance of $\mathtt{H}$ with*

$$t'', t' \approx t \quad \textit{and} \quad \epsilon' + \epsilon'' \geq \epsilon$$

Theorem 7.2. *Let $\mathcal{D}$ be a distinguisher that $(t, q_s, q_{sc}, q_c, q_d, \epsilon)$-breaks the invisibility of US_{Gen}. Suppose that the confirmation protocol and the disavowal protocol are ϵ_c-zero-knowledge and ϵ_d-zero-knowledge respectively. Then there exists an algorithm $\mathcal{C}_1$ that (t_1, ϵ_1)-breaks the IND-sID-CPA security of IBE, an algorithm $\mathcal{C}_2$ that $(t_2, q_s, q_{sc}, q_c, q_d, \epsilon_2)$-breaks the strong unforgeability of US_{Gen}, and an algorithm $\mathcal{C}_3$ that (t_3, ϵ_3)-breaks the collision-resistance of the hash function $\mathtt{H}$ with*

$$t_1, t_2, t_3 \approx t \quad \textit{and} \quad \epsilon_1 + \epsilon_2 + \epsilon_3 \geq \epsilon - q_c\epsilon_c - q_d\epsilon_d$$

The proofs of the two theorems above are deferred to Appendix C and D, respectively.

Theorem 7.3. *Suppose that the* $\mathsf{Disavow}$ *protocol is* (t, ϵ)*-sound. Then* US_{Gen} *is* (t', ϵ')*-non-claimable, where*

$$t' \approx t \quad \textit{and} \quad \epsilon' \leq \epsilon.$$

Proof. Let $\mathcal{A}$ be an adversary against the non-claimability of US_{Gen}, and let its output be $(\mathtt{pk}, M, \sigma, \mathsf{cvt})$ where $\mathtt{pk} = (\mathtt{pk}_{\mathsf{S}}, \mathtt{pk}_{\mathtt{H}}, \mathtt{mpk})$, $\sigma = (C, D, I, R)$ and $\mathsf{cvt} = \mathtt{sk}_I$. The validity of $\mathtt{sk}_I$ shows that it is indeed the corresponding secret key of identity I. Now suppose that $\mathsf{Ver}(\mathtt{pk}, M, \sigma, \mathsf{cvt}) = 1$. That is, $\mathsf{S.Ver}(\mathtt{pk}_{\mathsf{S}}, M \| I \| \bar{C}, \delta) = 1$, where $\bar{C} = \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_{\mathtt{H}}, C, R)$ and $\delta = \mathsf{IBE.Dec}(\mathtt{sk}_I, \mathtt{mpk}, (C, D))$. By the consistency of IBE, it indicates that the plaintext encapsulated in (C, D, I, R) is indeed the signer's signature on the message. Then by the soundness of $\mathsf{Disavow}$ protocol, we have that with probability at most ϵ the signer is able to fool the verifier. □

Discussion. We stress that the alternative generic construction of undeniable signature scheme is on the theoretic level. Though the algorithms are efficient, the two protocols involve general zero-knowledge proofs, which are usually complex and inefficient. Unfortunately, it still remains unknown if an instantiation with comparable efficiency to our concrete construction can be built. The main difficulty is in the incompatibility between the signature space of the signature scheme and the plaintext space of the IBE scheme.

8 Conclusion

We introduced the claimability attack into the context of convertible undeniable signature, and showed that some schemes are vulnerable to this attack. We then proposed a new concrete and highly efficient construction of fully functional convertible undeniable signature scheme immune to the new attack, and is provably secure without random oracles. It has short selective converter and universal converter, and admits efficient and simple confirmation and disavowal protocols. Our scheme supports delegation of conversion and confirmation/disavowal, threshold conversion and some other extensions. We also proposed an alternative generic construction of non-claimable convertible undeniable signature scheme, which is immune to claimability attacks as well. It also has short selective converter. The only disadvantage is the inefficient confirmation/disavowal protocol.

References

[1] L. E. Aimani. Toward a generic construction of universally convertible undeniable signatures from pairing-based signatures. In *INDOCRYPT08*, volume 5365 of *LNCS*, pages 145–157. Springer, 2008.

[2] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.

[3] D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT04*, volume 3027 of *LNCS*, pages 56–73. Springer, 2004.

[4] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO04*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[5] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[6] J. Boyar, D. Chaum, I. Damgård, and T. P. Pederson. Convertible undeniable signatures. In *CRYPTO90*, volume 537 of *LNCS*, pages 189–205. Springer, 1990.

[7] C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. In *ASIACRYPT98*, volume 1514 of *LNCS*, pages 271–285. Springer, 1998.

[8] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC07*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.

[9] E. Bresson and J. Stern. Proofs of knowledge for non-monotone discrete-log formulae and applications. In *ISC02*, volume 2433 of *LNCS*, pages 272–288. Springer, 2002.

[10] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *CRYPTO03*, volume 2729 of *LNCS*, pages 126–144. Springer, 2003.

[11] D. Chaum. Zero-knowledge undeniable signatures. In *EUROCRYPT90*, volume 473 of *LNCS*, pages 458–464. Springer, 1990.

[12] D. Chaum. Designated confirmer signatures. In *EUROCRYPT94*, volume 950 of *LNCS*, pages 86–91. Springer, 1995.

[13] D. Chaum and H. van Antwerpen. Undeniable signatures. In *CRYPTO89*, volume 435 of *LNCS*, pages 212–216. Springer, 1989.

[14] R. Cramer, I. Damgård, and P. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *PKC00*, volume 1751 of *LNCS*, pages 354–373. Springer, 2000.

[15] I. Damgård and T. Pedersen. New convertible undeniable signature schemes. In *EUROCRYPT96*, volume 1070 of *LNCS*, pages 372–386. Springer, 1996.

[16] E. v. H. David Chaum and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *CRYPTO91*, volume 576 of *LNCS*, pages 470–484. Springer, 1991.

[17] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, 1985.

[18] S. D. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In *CT-RSA03*, volume 2612 of *LNCS*, pages 80–97. Springer, 2003.

[19] S. D. Galbraith, W. Mao, and K. G. Paterson. RSA-based undeniable signatures for general moduli. In *CT-RSA02*, volume 2271 of *LNCS*, pages 200–217. Springer, 2002.

[20] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *EUROCRYPT99*, volume 1592 of *LNCS*, pages 123–139. Springer, 1999.

[21] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-based undeniable signatures. In *CRYPTO97*, volume 1294 of *LNCS*, pages 132–149. Springer, 1997.

[22] C. Gentry. Practical identity-based encryption without random oracles. In S. Vaudenay, editor, *EUROCRYPT06*, volume 4004 of *LNCS*, pages 445–464. Springer, 2006.

[23] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. Smart, editor, *EUROCRYPT08*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.

[24] D. Hofheinz and E. Kiltz. Programmable hash functions and their applications. In *CRYPTO08*, volume 5157 of *LNCS*, pages 21–38. Springer, 2008.

[25] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT96*, volume 1070 of *LNCS*, pages 143 – 154. Springer, 1996.

[26] S. V. Jean Monnerat. Generic homomorphic undeniable signatures. In *ASIACRYPT04*, volume 3329 of *LNCS*, pages 354–371. Springer, 2004.

[27] S. V. Jean Monnerat. Undeniable signatures based on characters. In *PKC04*, volume 2947 of *LNCS*, pages 69–85. Springer, 2004.

[28] H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS00*. The Internet Society, 2000.

[29] K. Kurasawa and S. H. Heng. 3-move undeniable signature scheme. In *EUROCRYPT05*, volume 3494 of *LNCS*, pages 181–197. Springer, 2005.

[30] K. Kurasawa and S. H. Heng. Relations among security notions for undeniable signature schemes. In *SCN06*, volume 4116 of *LNCS*, pages 34–48. Springer, 2006.

[31] K. Kurasawa and T. Takagi. New approach for selectively convertible undeniable signature schemes. In *ASIACRYPT06*, volume 4284 of *LNCS*, pages 428–443. Springer, 2006.

[32] F. Laguillaumie and D. Vergnaud. Short undeniable signatures without random oracles : The missing link. In *INDOCRYPT05*, volume 3797 of *LNCS*, pages 283–296. Springer, 2005.

[33] B. Libert and J.-J. Quisquater. Identity based undeniable signatures. In *CT-RSA04*, volume 2964 of *LNCS*, pages 112–125. Springer, 2004.

[34] M. Michels, H. Petersen, and P. Horster. Breaking and repairing a convertible undeniable signature scheme. In *CCS*, pages 148–152. ACM, 1996.

[35] M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. In *SAC97*, pages 231–244, 1997.

[36] 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.

[37] T. Okamoto and D. Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In *PKC01*, volume 1992 of *LNCS*, pages 104–118. Springer, 2001.

[38] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

[39] T. P. Pedersen. Distributed provers with applications to undeniable signatures. In *EUROCRYPT91*, volume 547 of *LNCS*, pages 221–242. Springer, 1991.

[40] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO91*, volume 576 of *LNCS*, pages 129–140. Springer, 1992.

[41] L. T. Phong, K. Kurosawa, and W. Ogata. New DLOG-based convertible undeniable signature schemes in the standard model. Cryptology ePrint Archive, Report 2009/394, 2009. `http://eprint.iacr.org/`.

[42] L. T. Phong, K. Kurosawa, and W. Ogata. New RSA-based (selectively) convertible undeniable signature schemes. In *AFRICACRYPT09*, volume 5580 of *LNCS*, pages 116–134. Springer, 2009.

[43] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, MIT, 1979.

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

[45] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO84*, pages 47–53, 1984.

[46] V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT01*, volume 1233 of *LNCS*, pages 256–266. Springer, 1997.

[47] B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *EUROCRYPT05*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

A Proof of Theorem 5.1

Proof. In the unforgeability game, we let M_i be the i-th signing query, $(\delta_i, \gamma_i, \theta_i)$ be the answer, and s_i be the exponent such that $\gamma_i = Y^{s_i}$ (and $\theta_i = u^{s_i}$). We also let (M, σ) be the adversary's forgery, where $\sigma = (\delta, \gamma, \theta) = (\delta, Y^s, u^s)$. Below we distinguish two cases:

Type-1: $\exists 1 \leq i \leq q_s$, $\gamma = \gamma_i$ (and $\theta = \theta_i$), which implies that $s = s_i$.

Type-2: $\forall 1 \leq j \leq q_s$, $\gamma \neq \gamma_j$ (and $\theta \neq \theta_i$), which implies that $s \notin \{s_1, \cdots, s_{q_s}\}$.

We denote by $\mathcal{F}_1$ (resp. $\mathcal{F}_2$) the forger who runs $\mathcal{F}$ but then only outputs the forgery if it is Type-1 (resp. Type-2). We show in the following two lemmas that Type-1 forger can be reduced to the q_s-SDH problem, and Type-2 forger can be reduced to the q_s-HSDH problem (and discrete logarithm problem). Then the theorem follows. □

Lemma A.1. *Suppose that $\mathcal{F}_1$ is a Type-1 forger that $(t_1, q_s, q_{sc}, q_c, q_d, \epsilon_1)$-breaks the existential unforgeability of US_{GBM}. Then there exists an adversary $\mathcal{A}$ that (t', ϵ')-breaks the q_s-SDH assumption with*

$$t' \approx t_1 \quad \textit{and} \quad \epsilon' \geq \frac{\varphi}{q_s}\left(\epsilon_1 - \frac{q_s^{m+1}}{p^m} - \phi\right)$$

Proof. To prove the lemma, we proceed in a series of games. In the following we denote by X_i the event that $\mathcal{F}_1$ wins in the i-th game.

Game 0. This is the original unforgeability game. By definition, we have that

$$\Pr[X_0] = \epsilon_1 \tag{9}$$

Game 1. Now we modify the game so that the key for the hash function is generated by $\mathsf{PHF.TrapGen}$. That is, the key for $\mathtt{H}$ is now chosen via $(\kappa', \tau) \leftarrow_\$ \mathsf{PHF.TrapGen}(1^k, g, h)$ for uniformly selected generators $g, h \in \mathbb{G}$. By the definition of $\mathtt{H}$, we obtain that

$$\Pr[X_1] \geq \Pr[X_0] - \phi \tag{10}$$

Game 2. In this game we choose the s_j used for answering signing queries not upon each signing query, but at the onset of the game. Since the s_j's were selected independently, this change is only conceptual. Let $\mathsf{S} = \bigcup_{j=1}^{q_s}\{s_j\}$ be the set of all s_j's, and let $\mathsf{S}^j = \mathsf{S} \setminus \{s_j\}$. We also change the selection of the elements g, h used during $(\kappa', \tau) \leftarrow \mathsf{PHF.TrapGen}(1^k, g, h)$ as follows. First, we choose at random $i \in \{1, \cdots, q_s\}$ and a generator $\tilde{g} \in \mathbb{G}$. Define

$$\mathsf{p}^i(\eta) = \prod_{t \in \mathsf{S}^i}(\eta + t) \quad \text{and} \quad \mathsf{p}(\eta) = \prod_{t \in \mathsf{S}}(\eta + t)$$

Note that $\deg(\mathsf{p}^i) = q_s - 1$ and $\deg(\mathsf{p}) = q_s$. We then set

$$g := \tilde{g}^{\mathsf{p}^i(x)}, \quad h := \tilde{g}^{\mathsf{p}(x)} \quad \text{and} \quad X := g^x = \tilde{g}^{x \cdot \mathsf{p}^i(x)}$$

all of which can be computed from $\tilde{g}, \tilde{g}^x, \cdots, \tilde{g}^{x^{q_s}}$. Here x is uniformly chosen from $\mathbb{Z}_p$ and is (part of) the secret key of the scheme. Note that we can compute $(x + s_j)$-th roots for $j \neq i$ from g and for all j from h, and that i is independent of the adversary's view. This change is also conceptual. So we have that

$$\Pr[X_2] = \Pr[X_1] \tag{11}$$

Game 3. We then change the way that the signature requests from the adversary are answered. Observe that the way we modified the generation of g and h in Game 2 implies that for any j with $\gamma_j \neq \gamma_i$ and $\theta_j \neq \theta_i$ (thus $s_j \neq s_i$), we have that

$$\begin{aligned} \delta_j = \mathtt{H}_{\kappa'}(M_j)^{\frac{1}{x+s_j}} &= \left(g^{a_{M_j}} h^{b_{M_j}}\right)^{\frac{1}{x+s_j}} \\ &= \left(\tilde{g}^{a_{M_j}\prod_{t\in\mathsf{S}^i}(x+t)} \tilde{g}^{b_{M_j}\prod_{t\in\mathsf{S}}(x+t)}\right)^{\frac{1}{x+s_j}} = \tilde{g}^{a_{M_j}\prod_{t\in\mathsf{S}^{i,j}}(x+t) + b_{M_j}\prod_{t\in\mathsf{S}^j}(x+t)} \end{aligned} \tag{12}$$

for $(a_{M_j}, b_{M_j}) \leftarrow \mathsf{PHF.TrapEval}(\tau, M_j)$. Therefore, for any $j \neq i$, we can generate the signatures $(\delta_j, \gamma_j, \theta_j) = (\delta_j, Y^{s_j}, u^{s_j})$ without explicitly knowing the secret key x, but instead using the right-hand side of (12) for computing δ_j. Note that in this game the game challenger still selects at random $y \in \mathbb{Z}_p$ by itself and computes Y as $Y := g^{1/y}$, and that the oracles of selective/universal conversion, confirmation/disavowal are all simulated by the game challenger using its knowledge of y. Obviously, this change in the game does not bring any difference to the adversary's advantage, and so we have that

$$\Pr[X_3] = \Pr[X_2] \tag{13}$$

Game 4. We now change the game so that if an s_j occurs more than m times, i.e. if there are pairwise distinct indices $j_1, \cdots, j_{m+1}$ with $\gamma_{j_1} = \cdots = \gamma_{j_{m+1}}$ and $\theta_{j_1} = \cdots = \theta_{j_{m+1}}$ (thus $s_{j_1} = \cdots = s_{j_{m+1}}$), we then abort and raise an event $\mathsf{abort}_{\mathsf{col}}$. There are at most $\binom{q_s}{m+1}$ such tuples $(j_1, \cdots, j_{m+1})$. For each tuple, the probability for $s_{j_1} = \cdots = s_{j_{m+1}}$ is $1/p^m$. A union bound shows that an $(m+1)$-wise collision occurs with probability at most

$$\Pr[\mathsf{abort}_{\mathsf{col}}] \leq \binom{q_s}{m+1} \frac{1}{p^m} \leq \frac{q_s^{m+1}}{p^m}$$

Hence, we get that

$$\Pr[X_4] \geq \Pr[X_3] - \Pr[\mathsf{abort}_{\mathsf{col}}] \geq \Pr[X_3] - \frac{q_s^{m+1}}{p^m} \tag{14}$$

Game 5. In this game we abort and raise an event $\mathsf{abort}_{\mathsf{bad.s}}$ if the adversary returns an $s \in \mathsf{S}^i$, i.e. the adversary returns a forgery (δ, γ, θ) with $\gamma = \gamma_j$ and $\theta = \theta_j$ for some j but $\gamma \neq \gamma_i$ and $\theta \neq \theta_i$ (thus $s \neq s_i$). Since i is uniformly chosen from $\{1, \cdots, q_s\}$, and independent from the adversary's view, we have that $\Pr[\mathsf{abort}_{\mathsf{bad.s}}] \leq 1 - 1/q_s$ for any choice of γ_j and θ_j. Hence, we obtain that

$$\Pr[X_5] = \Pr[X_4 \wedge \neg\mathsf{abort}_{\mathsf{bad.s}}] \geq \frac{1}{q_s} \Pr[X_4] \tag{15}$$

Game 6. If there is an index j with $\gamma_j = \gamma_i$ and $\theta_j = \theta_i$ (thus $s_j = s_i$) but $a_{M_j} \neq 0$ or if $a_M = 0$ for the adversary's forgery message, we then abort and raise an event $\mathsf{abort}_{\mathsf{bad.a}}$. That is, we raise $\mathsf{abort}_{\mathsf{bad.a}}$ if and only if we do not have $a_{M_j} = 0$ for all j with $\gamma_j = \gamma_i$, $\theta_j = \theta_i$ and $a_M \neq 0$. Since we have limited the number of such j to m in Game 4, by the programmability of $\mathtt{H}$, we then have that $\Pr[\mathsf{abort}_{\mathsf{bad.a}}] \leq 1 - \varphi$ for any choice of the M_j and s_j. So we get that

$$\Pr[X_6] = \Pr[X_5 \wedge \neg\mathsf{abort}_{\mathsf{bad.a}}] \geq \varphi \cdot \Pr[X_5] \tag{16}$$

Note that in this game, the game challenger never really uses the secret key x to generate signatures: to generate δ_j for $s_j \neq s_i$, we use (12) which does not require x. If $\mathsf{abort}_{\mathsf{bad.a}}$ does not occur, then $a_{M_j} = 0$ whenever $s_j = s_i$, so we can also use (12) to sign without the knowledge of x. On the other hand, if $\mathsf{abort}_{\mathsf{bad.a}}$ does occur, we must abort anyway, so actually no signature is required. Besides, Y in the public key is set according to the scheme, i.e. $Y := g^{1/y}$ for some random $y \in \mathbb{Z}_p$, and the challenger answers the adversary's universal conversion query and confirmation/disavowal queries by using its knowledge of y only. All together means that Game 6 does not use knowledge about the secret key x.

On the other hand, the adversary in Game 6 produces a forgery $(M, (\delta, \gamma, \theta))$ whenever X_6 occurs, which implies $\neg\mathsf{abort}_{\mathsf{bad.s}}$ and $\neg\mathsf{abort}_{\mathsf{bad.a}}$, we have that $\gamma = Y^s = Y^{s_i} = \gamma_i$, $\theta = u^s = u^{s_i} = \theta_i$, and

$$\delta = \mathtt{H}_{\kappa'}(M)^{\frac{1}{x+s}} = \left(\tilde{g}^{a_M \prod_{t \in \mathsf{S}^i}(x+t)} \tilde{g}^{b_M \prod_{t \in \mathsf{S}}(x+t)}\right)^{\frac{1}{x+s}} = \tilde{g}^{\frac{a_M \cdot \mathsf{p}^i(x)}{x+s}} \tilde{g}^{b_M \cdot \mathsf{p}^i(x)} = \tilde{g}^{\frac{a_M \cdot \mathsf{p}^i(x)}{x+s}} g^{b_M}$$

From δ and its knowledge about g and the s_j's, the game challenger can derive

$$\delta' = \left(\frac{\delta}{g^{b_M}}\right)^{\frac{1}{a_M}} = \tilde{g}^{\frac{\mathsf{p}^i(x)}{x+s}}$$

Since $\gcd(\eta + s, \mathsf{p}^i(\eta)) = 1$ (where we interpret $\eta + s$ and $\mathsf{p}^i(\eta)$ as polynomials in η), we can write $\mathsf{p}^i(\eta)/(\eta + s) = \mathsf{p}'(\eta) + q_0/(\eta + s)$ for some polynomial $\mathsf{p}'(\eta)$ of degree at most $q_s - 2$ and some constant

$q_0 \neq 0$. Note that the game challenger knows all s_j's including s, since these were selected by it and $s = s_i$. Again, we can compute $g' := \tilde{g}^{\mathsf{p}'(x)}$. We finally obtain that

$$\delta'' = \left(\frac{\delta'}{g'}\right)^{\frac{1}{q_0}} = \left(\tilde{g}^{\frac{\mathsf{p}^i(x)}{x+s} - \mathsf{p}'(x)}\right)^{\frac{1}{q_0}} = \tilde{g}^{\frac{1}{x+s}}$$

which, together with s, is a solution to the given q_s-SDH problem. This means that from Game 6, we can construct an adversary $\mathcal{A}$ that (t', ϵ')-breaks the q_s-SDH assumption, where the running time t' is approximately t_1, and $\mathcal{A}$'s advantage is $\epsilon' \geq \Pr[X_6]$.

Putting all together, we obtain that

$$\epsilon' \geq \frac{\varphi}{q_s}\left(\epsilon_1 - \frac{q_s^{m+1}}{p^m} - \phi\right)$$

□

Lemma A.2. *Suppose that $\mathcal{F}_2$ is a Type-2 forger that $(t_2, q_s, q_{sc}, q_c, q_d, \epsilon_2)$-breaks the existential unforgeability of US_{GBM}. Then there exists an adversary $\mathcal{A}$ that (t', ϵ')-breaks the q_s-HSDH assumption and an adversary $\mathcal{A}^*$ that (t'', ϵ'')-breaks the Discrete Logarithm assumption in $\mathbb{G}$ such that*

$$t', t'' \approx t_2 \quad \textit{and} \quad \epsilon' + \epsilon'' \geq \epsilon_2 - \phi$$

Proof. Again, we proceed in a series of games and denote by X_i the event that $\mathcal{F}_2$ wins the the i-th game.

Game 0. This is the original game. By definition, we have that

$$\Pr[X_0] = \epsilon_2 \tag{17}$$

Game 1. Now we modify the game so that the key for $\mathtt{H}$ is generated by $\mathsf{PHF.TrapGen}$. That is, we now choose the key for $\mathtt{H}$ via $(\kappa', \tau) \leftarrow \mathsf{PHF.TrapGen}(1^k, g, h)$ for uniformly selected generators $g, h \in \mathbb{G}$. By the programmability of $\mathtt{H}$, we obtain that

$$\Pr[X_1] \geq \Pr[X_0] - \phi \tag{18}$$

Game 2. In this game we change the way that g and h are chosen. Now we set $g := \tilde{g}$, $h := \tilde{g}^c$, $X := \tilde{g}^x$ and $u := \tilde{g}^\beta$, where c is uniformly selected from $\mathbb{Z}_p$, and $\tilde{g}, \tilde{g}^x, \tilde{g}^\beta$ are from an instance of the HSDH problem. Obviously, g, h, u are uniformly distributed in $\mathbb{G}$, and this change is purely conceptual. Then for each signature query M_j, we set

$$\delta_j := \left(\tilde{g}^{\frac{1}{x+s_j}}\right)^{a_{M_j} + c \cdot b_{M_j}}, \quad \gamma_j := (\tilde{g}^{s_j})^{1/y} \quad \text{and} \quad \theta_j := \tilde{u}^{s_j}$$

for $(a_{M_j}, b_{M_j}) \leftarrow \mathsf{PHF.TrapEval}(\tau, M_j)$. Obviously,

$$\delta_j = \tilde{g}^{\frac{a_{M_j} + c \cdot b_{M_j}}{x+s_j}} = \left(g^{a_{M_j}} \cdot h^{b_{M_j}}\right)^{\frac{1}{x+s_j}} = \mathtt{H}_{\kappa'}(M_j)^{\frac{1}{x+s_j}}, \quad \gamma_j = (g^{1/y})^{s_j} = Y^{s_j} \quad \text{and} \quad \theta_j = u^{s_j}$$

So $(\delta_j, \gamma_j, \theta_j)$ is a valid (and uniformly distributed) signature on M_j. Therefore, these changes do not bring any difference to the adversary's advantage, and we have that

$$\Pr[X_2] = \Pr[X_1] \tag{19}$$

Note that in this game, the game challenger need not know the values of the s_j's. On the other hand, the challenger still knows y and sets Y according to the scheme. The selective/universal conversion and confirmation/disavowal protocols are simulated by it using the knowledge of y.

Game 3. We now abort and raise an event $\mathsf{abort}_{\mathsf{log}}$ if $a_M + c \cdot b_M \equiv 0 \bmod p$ for the message in the adversary's forgery or $a_{M_j} + c \cdot b_{M_j} \equiv 0 \bmod p$ for any signature query M_j. Since we chose c in Game 2 as a uniform exponent and only pass g and $h = g^c$ (but no further information about c) to the adversary and $\mathsf{PHF.TrapGen}$, these algorithms break a discrete logarithm problem. We get that

$$\Pr[X_3] \geq \Pr[X_2] - \Pr[\mathsf{abort}_{\mathsf{log}}] \geq \Pr[X_2] - \epsilon'' \tag{20}$$

for a suitable (t'', ϵ'')-attacker $\mathcal{A}^*$ against the discrete logarithm problem in $\mathbb{G}$ with $t'' \approx t_2$.

Now in this game, we can construct an adversary $\mathcal{A}$ against the q_s-HSDH assumption. $\mathcal{A}$ takes inputs $\tilde{g}, \tilde{u}, \tilde{g}^x, \tilde{g}^{1/(x+s_1)}, \tilde{g}^{s_1}, \tilde{u}^{s_1}, \cdots, \tilde{g}^{1/(x+s_{q_s})}, \tilde{g}^{s_{q_s}}, \tilde{u}^{s_{q_s}}$ and simulates Game 3 with adversary $\mathcal{F}_2$. $\mathcal{A}$ uses its inputs as if it was selected by the experiment. Note that in Game 3, the secret key x is never used. Now, whenever $\mathcal{F}_2$ outputs a forgery $(M, (\delta, \gamma, \theta))$ with $\gamma \notin \{(\tilde{g}^{s_1})^y, \cdots, (\tilde{g}^{s_{q_s}})^y\}$ and $\theta \notin \{\tilde{u}^{s_1}, \cdots, \tilde{u}^{s_{q_s}}\}$, and

$$\delta = \left(g^{a_M} h^{b_M}\right)^{\frac{1}{x+s}} = \left(\tilde{g}^{a_M + c \cdot b_M}\right)^{\frac{1}{x+s}}$$

Since $a_M + c \cdot b_M \not\equiv 0 \bmod p$, we can compute a nontrivial $(x+s)$-th root of $\tilde{g}$. Therefore, we have

$$\delta' = \delta^{\frac{1}{a_M + c \cdot b_M}} = \tilde{g}^{\frac{1}{x+s}}$$

which, together with $\tilde{g}^s = g^s = (Y^s)^y = \gamma^y$ and $\tilde{u}^s = u^s = \theta$, forms a solution to the given q_s-HSDH problem.

Putting everything together, we obtain that $\epsilon' + \epsilon'' \geq \epsilon_2 - \phi$. □

B Proof of Theorem 5.2

Proof. Again, to prove the theorem, we proceed in a series of games. We denote by X_i the event that $\mathcal{D}$ wins the i-th game. In these games, we let M_j be the j-th signature query, $(\delta_j, \gamma_j, \theta_j)$ be the corresponding answer, and s_j be the exponent such that $\gamma_j = Y^{s_j}$ and $\theta_j = u^{s_j}$. We also let M be the challenge message chosen by the adversary and $\sigma = (\delta, \gamma, \theta)$ be the corresponding challenge signature.

Game 0. This is the original invisibility game. By definition, we have that

$$\Pr[X_0] = \epsilon \tag{21}$$

Game 1. We modify the game so that now the key for the hash function $\mathtt{H}$ is generated using $\mathsf{PHF.TrapGen}$. Namely, we use the trapdoor key generation $(\kappa', \tau) \leftarrow \mathsf{PHF.TrapGen}(1^k, g, h)$ for uniformly selected generators $g, h \in \mathbb{G}$. By the programmability of $\mathtt{H}$, we have that

$$\Pr[X_1] \geq \Pr[X_0] - \varphi \tag{22}$$

Game 2. For any message/signature pair (M_l, σ_l) submitted by the adversary to the selective conversion oracle or the confirmation oracle, if the adversary never queried the signing oracle on M_l, or it requested a signature on M_l but the answer returned by the oracle is different from σ_l, we abort and raise an event $\mathsf{abort}_{\mathsf{suf}}$. Besides, for a disavowal query (M_l, σ_l), if the adversary ever queried the signing oracle on M_l and obtained σ_l from it, the disavowal oracle simply returns $\perp$. Obviously, by the strong unforgeability of US_{GBM}, we have that

$$\Pr[X_2] \geq \Pr[X_1] - \Pr[\mathsf{abort}_{\mathsf{suf}}] \geq \Pr[X_1] - \epsilon_1 \tag{23}$$

Game 3. We change confirmation oracle so that given a message/signature pair (M_l, σ_l), the oracle runs the simulator of the confirmation protocol to produce an indistinguishable proof. By the zero knowledge property of the confirmation protocol and the union bound, we have that

$$\Pr[X_3] \geq \Pr[X_2] - q_c \cdot \epsilon_2 \tag{24}$$

Game 4. Similarly, we now change disavowal oracle so that given a message/signature pair (M_l, σ_l), the oracle runs the simulator of the disavowal protocol to produce an indistinguishable proof. By the zero knowledge property of the disavowal protocol and the union bound, we have that

$$\Pr[X_4] \geq \Pr[X_3] - q_d \cdot \epsilon_3 \tag{25}$$

Game 5. Now we change the selection of g, h and Y. We now set $g := \tilde{g}$, $h := \tilde{g}^c$, $u := \tilde{g}^\beta$, $X := g^x = \tilde{g}^x$ and $Y := (\tilde{g}^\beta)^d$, where $\tilde{g}, \tilde{g}^x, \tilde{g}^\beta$ are from a random instance of the DHSDH problem, and c, d are uniformly chosen from $\mathbb{Z}_p$. Note that the secret key y is implicitly defined to be $y = (d \cdot \log_{\tilde{g}} \tilde{u})^{-1}$. Obviously, this change is purely conceptual. Then for each signature query M_j, the game challenger computes

$$\delta_j := \mathtt{H}_{\kappa'}(M_j)^{\frac{1}{x+s_j}} = \left(g^{a_{M_j}} h^{b_{M_j}}\right)^{\frac{1}{x+s_j}} = \left(\tilde{g}^{\frac{1}{x+s_j}}\right)^{a_{M_j}+c\cdot b_{M_j}}, \quad \gamma_j := Y^{s_j} = (\tilde{u}^{s_j})^d \quad \text{and} \quad \theta_j := u^{s_j} = \tilde{u}^{s_j}$$

for $(a_{M_j}, b_{M_j}) \leftarrow \mathsf{PHF.TrapEval}(\tau, M_j)$. To selectively convert (δ_j, γ_j), the oracle returns

$$\nu_j := \gamma_j^y = \left((\tilde{u}^{s_j})^d\right)^{\left(d \cdot \log_{\tilde{g}} \tilde{u}\right)^{-1}} = \tilde{g}^{s_j}$$

Note that all of the signature queries and selective conversion queries can be answered using the tuples $(\tilde{g}^{1/(x+s_j)}, \tilde{g}^{s_j}, \tilde{u}^{s_j})$ given in the DHSDH problem instance. Clearly, this change does not bring any difference to the adversary's advantage. Therefore, we have that

$$\Pr[X_5] = \Pr[X_4] \tag{26}$$

Note that in Game 5, only the generation of the challenge signature requires the knowledge of the secret key x.

Game 6. In this game if for the challenge message M we have that $a_M + c \cdot b_M \equiv 0 \bmod p$ for $(a_M, b_M) \leftarrow \mathsf{PHF.TrapEval}(\tau, M)$, we then abort and raise an event $\mathsf{abort_{log}}$. Since we chose c as a uniform exponent and only pass g and $h = g^c$ (but no further information about c) to the adversary and $\mathsf{PHF.TrapGen}$, these algorithms break a discrete logarithm. Hence we have that

$$\Pr[X_6] = \Pr[X_5] - \Pr[\mathsf{abort_{log}}] \geq \Pr[X_5] - \epsilon'' \tag{27}$$

for a suitable (t'', ϵ'')-attacker $\mathcal{A}'$ on the discrete logarithm problem in $\mathbb{G}$ with $t'' \approx t$.

Game 7. In this game we change the generation of the challenge signature. Given the challenge message M from the adversary, the challenger computes

$$\delta = Z_b^{a_M + c \cdot b_M}, \quad \gamma = Y^s = (\tilde{u}^s)^d \quad \text{and} \quad \theta = \tilde{u}^s$$

where Z_b and $\tilde{u}^s$ are from the given instance of the DHSDH problem. If the bit in the DHSDH assumption is $b = 0$, we have that

$$\delta = Z_0^{a_M + c \cdot b_M} = \left(\tilde{g}^{\frac{1}{x+s}}\right)^{a_M + c \cdot b_M} = \left(g^{a_M + c \cdot b_M}\right)^{\frac{1}{x+s}} = \mathtt{H}_{\kappa'}(M)^{\frac{1}{x+s}}$$

So (δ, γ, θ) is a valid signature on M. On the other hand, if the bit is $b = 1$, we have that Z_b is a random element of $\mathbb{G}$, and so is δ. So (δ, γ, θ) is a random element from the signature space. The challenger returns $\sigma := (\delta, \gamma, \theta)$ to the adversary. It is readily seen that the challenge signature is identically distributed as a real one. So we have that

$$\Pr[X_7] = \Pr[X_6] \tag{28}$$

Note that in Game 7, no knowledge of the secret key x is required. We then can build another algorithm for breaking the q_s-DHSDH assumption using the adversary in this game, whose running time is approximately the same as t. Therefore, we have that

$$\epsilon' \geq \Pr[X_7] \tag{29}$$

Putting everything together, we obtain that $\epsilon' \geq \epsilon - \varphi - \epsilon_1 - q_c \cdot \epsilon_2 - q_d \cdot \epsilon_3 - \epsilon''$. □

C Proof of Theorem 7.1

Proof. We proceed in a series of games.

Game 0. This is the original unforgeability game. By definition, we have that

$$\Pr[X_0] = \epsilon$$

Game 1. Consider the signatures returned by the signature oracle and those submitted by the adversary to the selective conversion, confirmation and disavowal oracles, if there exist two signatures say (C_i, D_i, I_i, R_i) and (C_j, D_j, I_j, R_j) with $\mathtt{H}.\mathsf{Eval}(\mathtt{pk_H}, C_i, R_i) = \mathtt{H}.\mathsf{Eval}(\mathtt{pk_H}, C_j, R_j)$ but $(C_i, R_i) \neq (C_j, R_j)$, we abort and raise an event $\mathsf{abort_{col}}$. If this event happens, these algorithms break the collision resistance of the hash function. We get that

$$\Pr[X_1] \geq \Pr[X_0] - \epsilon''$$

for a suitable (t'', ϵ'')-attacker $\mathcal{B}'$ against the collision resistance of $\mathtt{H}$ with $t'' \approx t$. Next we show that $\Pr[X_1]$ is upper bounded by ϵ' by constructing an algorithm $\mathcal{B}$ against the strong unforgeability of S.

Algorithm $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine. Given a public key $\mathtt{pk_S}$ of signature scheme S, $\mathcal{B}$ runs $\mathsf{IBE}.\mathsf{Kg}(1^k)$ and $\mathtt{H}.\mathsf{Kg}(1^k)$ to generate key pairs for IBE and $\mathtt{H}$ respectively, say $(\mathtt{mpk}, \mathtt{msk})$ and $(\mathtt{pk_H}, \mathtt{sk_H})$, and invokes $\mathcal{A}$ on input $((\mathtt{pk} = (\mathtt{pk_S}, \mathtt{pk_H}, \mathtt{mpk}), \mathtt{msk})$. It then answers queries issued by $\mathcal{A}$ as below.

Signature Query. Given a message M, $\mathcal{B}$ first randomly selects an identity $I \in \mathcal{I}$ and a random number $R \in \mathcal{R}$, and calls $\mathsf{IBE}.\mathsf{EncRand}$ on input $(\mathtt{mpk}, I)$ to generate (C, ω). It then computes the hash value $\overline{C}$ of C and R, and asks its own signature oracle to produce a signature δ on $M \| I \| \overline{C}$. After that, $\mathcal{B}$ runs $\mathsf{IBE}.\mathsf{EncPltx}$ on input (ω, δ) to generate D. It returns (C, D, I, R) back to $\mathcal{A}$.

Confirmation/Disavowal Query. Given a message-signature pair, i.e. $(M, \sigma = (C, D, I, R))$, $\mathcal{B}$ first checks the validity of σ as in handling selective conversion queries. If valid, it starts an execution of the confirmation protocol with $\mathcal{A}$; otherwise, it starts an execution of the disavowal protocol with $\mathcal{A}$. In either case, $\mathcal{B}$ uses $(\delta, \mathtt{sk}_I, \mathtt{msk})$ as the witness, where $\delta, \mathtt{sk}_I$ are derived from $\mathtt{msk}$ and σ as specified in the scheme.

At the end of the game, $\mathcal{A}$ outputs its forgery $(M^*, \sigma^* = (C^*, D^*, I^*, R^*))$. Suppose that $\mathcal{A}$ succeeds, and thus σ^* is a valid signature on M^* under $\mathtt{pk}$. Let $\mathtt{sk}_{I^*}$ be the secret key of I^* in IBE and let δ^* be the plaintext recovered from (C^*, D^*) using $\mathtt{sk}_{I^*}$, both of which can be computed by $\mathcal{B}$. We have that $\mathsf{S}.\mathsf{Ver}(\mathtt{pk_S}, M^* \| I^* \| \overline{C}^*, \delta^*) = 1$, where $\overline{C}^* = \mathtt{H}.\mathsf{Eval}(\mathtt{pk_H}, C^*, R^*)$. So $\mathcal{B}$ outputs $(M^* \| I^* \| \overline{C}^*, \delta^*)$ as its forgery for the signature scheme S.

Now we assume that $(M^*\|I^*\|\overline{C}^*, \delta^*)$ is the same as $(M\|I\|\overline{C}, \delta)$ that $\mathcal{B}$ ever obtained from its signature oracle. Since $I^* = I$ and $\overline{C}^* = \overline{C}$, according to the game specification, i.e. event $\mathsf{abort_{col}}$ did not happen, we have that $C^* = C$ and $R^* = R$. By the separability of IBE, it turns out that $D^* = D$ as well. Therefore, we obtain that $(M^*, \sigma^*) = (M, \sigma)$, which contradicts the success of $\mathcal{A}$. Consequently, $(M^*\|I^*\|\overline{C}^*, \delta^*)$ is a valid forgery for S, and $\mathcal{B}$ breaks the strong unforgeability of S with probability at least the same as that of $\mathcal{A}$ in breaking the strong unforgeability of US_{Gen}.

Putting everything together, we then obtain that $\epsilon' + \epsilon'' \geq \epsilon$. □

D Proof of Theorem 7.2

Proof. We proceed in a series of invisibility games.

Game 0. This is the original game. By the definition, we have that

$$\Pr[X_0] = \epsilon$$

Game 1. Consider the signatures returned by the signature oracle and those submitted by the adversary to the selective conversion, confirmation and disavowal oracles, if there exist two signatures say (C_i, D_i, I_i, R_i) and (C_j, D_j, I_j, R_j) with $\mathtt{H}.\mathsf{Eval}(\mathtt{pk_H}, C_i, R_i) = \mathtt{H}.\mathsf{Eval}(\mathtt{pk_H}, C_j, R_j)$ but $(C_i, R_i) \neq (C_j, R_j)$, we abort and raise an event $\mathsf{abort_{col}}$. If this event happens, these algorithms break the collision resistance of the hash function. We get that

$$\Pr[X_1] \geq \Pr[X_0] - \epsilon_3$$

for a suitable (t_3, ϵ_3)-attacker $\mathcal{C}_3$ against the collision resistance of $\mathtt{H}$ with $t_3 \approx t$.

Game 2. Now consider the query (M, σ) that $\mathcal{D}$ submits to the selective conversion oracle, confirmation oracle or disavowal oracle. If σ is a valid signature on M but (M, σ) was not a pair that the adversary obtained from its signature oracle, we abort and raise an event $\mathsf{abort_{suf}}$. If this event happens, these algorithms break the strong unforgeability of US_{Gen}. We have that

$$\Pr[X_2] \geq \Pr[X_1] - \epsilon_2$$

for a suitable (t_2, ϵ_2)-attacker $\mathcal{C}_2$ against the strong unforgeability of US_{Gen} with $t_2 \approx t$.

Game 3. In this game all confirmation queries are handled by calling the simulator of the confirmation protocol instead of using $\mathtt{msk}$, which may rewind the adversary. Since the protocol is zero-knowledge, this change brings a difference of at most $q_c\epsilon_c$ to the adversary's success probability. So we have that

$$\Pr[X_3] \geq \Pr[X_2] - q_c\epsilon_c$$

where q_c is the number of confirmation queries.

Game 4. Similar to Game 3, now we answer all the disavowal queries using the simulator of the disavowal protocol. We obtain that

$$\Pr[X_4] \geq \Pr[X_3] - q_d\epsilon_d$$

where q_d is the number of disavowal queries.

Game 5. We change the game so that the identity I^* in the challenge signature σ^* is now chosen at the very onset of the game, even before the generation of the public key of US_{Gen}. This change is purely conceptual. So we have

$$\Pr[X_5] = \Pr[X_4]$$

Next we show that $\Pr[X_5]$ is upper bounded by ϵ_1 by constructing an algorithm $\mathcal{C}_1$ for breaking the IND-sID-CPA security of IBE, which runs the adversary $\mathcal{D}$ as a subroutine.

Algorithm $\mathcal{C}_1$ selects at random an identity $I^* \in \mathcal{I}$, submits it to its challenger in the IND-sID-CPA game, and is returned a master public key $\mathtt{mpk}$. It then generates a key pair $(\mathtt{pk}_{\mathsf{S}}, \mathtt{sk}_{\mathsf{S}})$ for the signature scheme S and a key pair $(\mathtt{pk}_{\mathtt{H}}, \mathtt{sk}_{\mathtt{H}})$ for the hash function $\mathtt{H}$, and invokes $\mathcal{D}$ on input $(\mathtt{pk}_{\mathsf{S}}, \mathtt{pk}_{\mathtt{H}}, \mathtt{mpk})$. $\mathcal{C}_1$ then answers $\mathcal{D}$'s queries as below.

Signature Query. Given a message M, $\mathcal{C}_1$ selects an identity $I \in \mathcal{I}\backslash\{I^*\}$ at random, and computes a signature σ using $\mathtt{sk}_{\mathsf{S}}$ by following the Sign algorithm of US_{Gen}.

Selective Conversion Query. Given $(M, \sigma = (C, D, I, R))$, $\mathcal{C}_1$ submits I to its extraction oracle and obtains $\mathtt{sk}_I$. It returns $\mathtt{sk}_I$ to $\mathcal{D}$.

Confirmation/Disavowal Query. These queries are handled by $\mathcal{C}_1$ using the corresponding simulator, as specified by the game.

At some time, $\mathcal{D}$ submits a message M^*. $\mathcal{C}_1$ first runs $\mathsf{IBE.EncRand}(\mathtt{mpk}, I^*)$ and obtains C'. It selects at random $R' \in \mathcal{R}$, and computes $\overline{C}^* = \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_{\mathtt{H}}, C', R')$. Then it signs $M^* \| I^* \| \overline{C}^*$ using $\mathtt{sk}_{\mathsf{S}}$. $\mathcal{C}_1$ and obtains δ_0. $\mathcal{C}_1$ also selects at random another signature σ_1 from the signature space of S. It then submits (δ_0, δ_1) to its challenger of the IND-sID-CPA game, which chooses one of them at random and encrypts. After receiving the ciphertext (C^*, D^*) from the challenger, $\mathcal{C}_1$ uses $\mathtt{sk}_{\mathtt{H}}$ to trapdoor invert $\overline{C}^*$ and finds R^* such that $\overline{C}^* = \mathtt{H}.\mathsf{Eval}(\mathtt{pk}_{\mathtt{H}}, C^*, R^*)$, and returns $\sigma^* = (C^*, D^*, I^*, R^*)$ back to $\mathcal{D}$. Note that if (C^*, D^*) is a ciphertext of δ_0, σ^* is also a well distributed and valid signature. If (C^*, D^*) is a ciphertext of δ_1 which is randomly chosen from the signature space of S, σ^* is also a random signature uniformly distributed in the signature space of US_{Gen}.

$\mathcal{C}_1$ continues to answer $\mathcal{D}$'s queries as above. Finally, $\mathcal{D}$ outputs a bit b'. $\mathcal{C}_1$ then outputs b' and halts. Clearly, all the queries submitted by $\mathcal{D}$ were perfectly answered, and the challenge signature was also perfectly generated. If $\mathcal{D}$ succeeds in outputting the correct bit, so does $\mathcal{C}_1$. Thus, we have that

$$\epsilon_1 \geq \Pr[X_5]$$

Putting everything together, we then obtain that $\epsilon_1 + \epsilon_2 + \epsilon_3 \geq \epsilon - q_c\epsilon_c - q_d\epsilon_d$. □

E Security of DHSDH Assumption in Generic Bilinear Groups

To give more confidence in the DHSDH assumption, we prove a lower bound of computational complexity of q-DHSDH problem in the generic group model [46, 3]. In this model, the adversary can only perform group operations in $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear pairing $\mathsf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, by interacting with an oracle $\mathcal{O}$ so that it only sees group elements encoded as unique random strings. This is modeled using two encoding functions, ξ and ξ' for $\mathbb{G}$ and $\mathbb{G}_T$ respectively. A group element $g^t \in \mathbb{G}$ is represented as the string $\xi(t)$. Elements of $\mathbb{G}_T$ are represented similarly using ξ'. For convenience, we re-state the DHSDH assumption briefly below.

The q-DHSDH assumption states that for any adversary $\mathcal{D}$, for $x, \beta, s_1, \cdots, s_q, s \leftarrow_{\$} \mathbb{Z}_p$ and $Z \leftarrow_{\$} \mathbb{G}$, the following is negligible.

$$\left|\Pr\left[\mathcal{D}\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}, g^{\frac{1}{x+s}}\right) = 1\right] - \Pr\left[\mathcal{D}\left(g, g^x, g^\beta, \left\{g^{\frac{1}{x+s_i}}, g^{s_i}, g^{\beta s_i}\right\}_{i=1}^{q}, g^{\beta s}, Z\right) = 1\right]\right|$$

Theorem E.1. *Let $\mathcal{D}$ be an algorithm that solves the q-DHSDH problem in the generic group model, making at most ℓ queries to the oracles computing the group action in $\mathbb{G}, \mathbb{G}_T$, and the oracle computing the bilinear pairing e. Suppose that $x, \beta, s_1, \cdots, s_q, s, r \leftarrow_{\$} \mathbb{Z}_p$, $b \leftarrow_{\$} \{0,1\}$, and ξ, ξ' are chosen at random. Set $w_b = 1/(x+s)$ and $w_{1-b} = r$. Then $\mathcal{D}$'s advantage*

$$\epsilon := \left|\Pr\Big[\mathcal{D}\Big(\xi(1), \xi(x), \xi(\beta), \Big\{\xi(\frac{1}{x+s_i}), \xi(s_i), \xi(\beta s_i)\Big\}_{i=1}^{q}, \xi(\beta s), \xi(w_0), \xi(w_1)\Big) = b\Big] - \frac{1}{2}\right|$$

is bounded by

$$\epsilon \leq \frac{2(\ell + 3q + 6)^2 q + q + 1}{p} = O\left(\frac{(\ell + q)^2 q}{p}\right)$$

Proof. We construct a simulator S that simulates the generic group oracles without committing to values for $x, \beta, s_1, \cdots, s_q, s, r$. S keeps track of the group elements by their discrete logarithms to the generator $g \in \mathbb{G}$ and $\mathsf{e}(g, g) \in G_T$. Since the variables $x, \beta, s_1, \cdots, s_q, s, r$ are undetermined, these discrete logarithms are polynomials in $\mathbb{F}_p[x, \beta, s_1, \cdots, s_q, s, r]$, which we denote by ρ_i for expressions in $\mathbb{G}$ and ρ'_i for expressions in $\mathbb{G}_T$. S then maps the corresponding group elements to random strings it gives to $\mathcal{D}$, i.e. in group $\mathbb{G}$ it associates ρ_i to $\xi_i = \xi(\rho_i)$, and in $\mathbb{G}_T$ it associates ρ'_i to $\xi'_i = \xi'(\rho_i)$.

At the beginning of the game, S creates the following strings to the adversary, which corresponds to an instance of the DHSDH problem.

- three strings, ξ_0, ξ_1, ξ_2, which binds to $\rho_0 = 1$, $\rho_1 = x$ and $\rho_2 = \beta$ respectively;
- $3q$ strings, $(\xi_{3i}, \xi_{3i+1}, \xi_{3i+2})$ for $i = 1, \cdots, q$, which binds to $\rho_{3i} = \frac{1}{x+s_i}$, $\rho_{3i+1} = s_i$, and $\rho_{3i+2} = \beta s_i$ respectively;
- three strings, $\xi_{3q+3}, \xi_{3q+4}, \xi_{3q+5}$, which binds to βs, w_0 and w_1 respectively, where $w_b = \frac{1}{x+s}$ and $w_{1-b} = r$.

For simplicity and to avoid dealing with ratios, we reduce all the expressions to the common denominator $\Delta = (x + s) \prod_{i=1}^{q} (x + s_i)$, and for all i, we define $\pi_i = \rho_i \Delta$ and $\pi'_k = \rho'_i \Delta$. Note that all these π_i are polynomials in $\mathbb{F}_p[x, \beta, s_1, \cdots, s_q, s, r]$ of degree at most $q + 3$.

S maintains two lists, L which contains all the $3q + 6$ polynomial-string pairs created above i.e. (π_i, ξ_i), and L' which is initially empty, and initiates two counters $\tau = 3q + 6$ and $\tau' = 0$. It gives all the strings created above to $\mathcal{D}$, and then simulates the oracles for $\mathcal{D}$ as below, where without loss of generality, we assume that $\mathcal{D}$ only queries S on legitimate strings that were previously revealed.

Group Actions. To compute the product/division of two operands in the group $\mathbb{G}$ represented as ξ_i and ξ_j, where $0 \leq i, j < \tau$, S computes $\pi_\tau \leftarrow \pi_i \pm \pi_j$ depending on whether a multiplication or a division is requested. If $\pi_\tau = \pi_l$ for some l with $0 \leq l < \tau$, S sets $\xi_\tau = \xi_l$; otherwise, it sets ξ_τ to a random string in $\{0, 1\}^*$ distinct from the strings in L. S then appends the new pair (π_τ, ξ_τ) to L, gives ξ_τ to $\mathcal{D}$, and increases τ by one. Group action queries in $\mathbb{G}_T$ are treated similarly.

Pairings. Given two operands ξ_i and ξ_j with $0 \leq i, j < \tau$, S computes the product $\pi'_{\tau'} \leftarrow \pi_i \cdot \pi_j$. If $\pi'_{\tau'} = \pi'_l$ for some l with $0 \leq l < \tau'$, S sets $\xi'_{\tau'} = \xi'_l$; otherwise, it sets $\xi'_{\tau'}$ to a random string in $\{0, 1\}^*$ distinct from those in the list L'. S then appends the new pair $(\pi'_{\tau'}, \xi'_{\tau'})$ to L', gives $\xi'_{\tau'}$ to $\mathcal{D}$, and increases τ' by one.

Note that at any time in the game, all the polynomials used by S to represent an element in $\mathbb{G}$ have degree at most $q + 3$, and the polynomials to represent elements in $\mathbb{G}_T$ have degree at most $2q + 6$.

When $\mathcal{D}$ terminates after making at most ℓ queries, it outputs a bit b' for the guess of b. S chooses a random assignment, i.e. $x = x^*$, $\beta = \beta^*$, $s_1 = s_1^*$, $\cdots$, $s_q = s_q^*$, $s = s_q^*$ and $r = r^*$. The simulation provided by S is perfect and reveals nothing to $\mathcal{D}$ unless the chosen random values for the variables results in a non-trivial equality relation between the simulated group elements that was not revealed to $\mathcal{D}$. This happens if either of the following holds:

1. $\pi_i(x^*, \beta^*, s_1^*, \cdots, s_q^*, s^*, r^*) - \pi_j(x^*, \beta^*, s_1^*, \cdots, s_q^*, s^*, r^*) = 0$ but $\pi_i \neq \pi_j$ for some $0 \leq i \neq j < \tau$;
2. $\pi'_i(x^*, \beta^*, s_1^*, \cdots, s_q^*, s^*, r^*) - \pi'_j(x^*, \beta^*, s_1^*, \cdots, s_q^*, s^*, r^*) = 0$ but $\pi'_i \neq \pi'_j$ for some $0 \leq i \neq j < \tau'$;
3. any relation similar to the above in which $1/(x + s)$ and r have been exchanged;
4. $\Delta(x^*, \beta^*, s_1^*, \cdots, s_q^*, s^*, r^*) = 0$.

Because the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ are implemented by the addition/subtraction between polynomials in L and L' respectively, and the pairing operations are implemented by the multiplication of polynomials in L, it is unable for the adversary to trivially obtain the polynomial $(x+s)\Delta$ via these operations.

Since $\pi_i - \pi_j$ for fixed i and j is of degree at most $q+3$, it equals zero for a random assignment of the variables in $\mathbb{Z}_p$ with probability at most $(q+3)/p$. Similarly, for fixed i and j, $\pi'_i - \pi'_j$ becomes zero with probability $(2q+6)/p$. The same probabilities can be found in the third case. Regarding the fourth case, we have that the probability that it occurs is at most $(q+1)/p$.

Conditioned on that the events above do not happen, the distribution of the bit b in $\mathcal{D}$'s view is independent and $\mathcal{D}$'s probability of making a correct guess is exactly $1/2$. Therefore, we have that $\mathcal{D}$ makes a correct guess with advantage bounded by

$$\epsilon \leq 2\left(\binom{\tau}{2}\frac{q+3}{p} + \binom{\tau'}{2}\frac{2q+6}{p}\right) + \frac{q+1}{p}$$

Since $\tau + \tau' \leq \ell + 3q + 6$, we then obtain that

$$\epsilon \leq \frac{2(\ell+3q+6)^2 q + q + 1}{p}$$

This completes the proof. □