A Generic Construction for Universally-Convertible Undeniable Signatures

Xinyi Huang, Yi Mu, Willy Susilo, and Wei Wu

Centre for Computer and Information Security Research
School of Computer Science & Software Engineering
University of Wollongong, Australia
{xh068,ymu,wsusilo,wei}@uow.edu.au

Abstract. Undeniable signatures are classic digital signatures which are not universally verifiable and can only be verified with the help of the signer. Its extended version, convertible undeniable signatures, equips the signer with the additional ability to make his undeniable signatures universally verifiable whenever required. A selectively-convertible undeniable signature scheme allows the signer to convert a single signature into a universally verifiable signature by releasing a selective proof in a later time, while "universally-convertible" refers to the case where the signer has the additional ability to generate a universal proof which can finally convert *all* his undeniable signatures into universally verifiable signatures. In this paper, we propose a *generic* construction for universally-convertible undeniable signatures. Our construction is based on three building blocks: a strongly unforgeable classic signature scheme, a selectively-convertible undeniable signature scheme and a collision-resistant hash function. Formal proofs guarantee that our construction has a tight security reduction to the underlying security assumptions. As one of the applications of our generic construction, one can obtain the *first* provable secure universally-convertible undeniable signature scheme in the standard model.

Keywords: Undeniable Signature, Universally-Convertible, Generic Construction, Provable Security.

1 Introduction

Universal verifiability is one of the most important properties in classic digital signatures. This property allows everybody to check the correctness of a signature. However, for some personally or commercially sensitive applications, universal verifiability is not required or even undesirable during certain periods. Therefore, the concept of undeniable signature was introduced by Chaum and van Antwerpen in Cypto'89 [6].

Undeniable signatures are like classic digital signatures, with the only difference that they are not universally verifiable. Instead, the validity or invalidity of an undeniable signature can only be verified via the Confirmation/Disavowal

F. Bao et al. (Eds.): CANS 2007, LNCS 4856, pp. 15–33, 2007.

protocol with the help of the signer. Undeniable signatures have found various applications in cryptography such as in licensing software [6], electronic cash [43], electronic voting and auctions. The first undeniable signature was proposed by Chaum and van Antwerpen [6] and it was further improved by Chaum in [7]. However, the unforgeability of the FDH (Full Domain Hash) variant of Chaum's scheme remains as an open problem and was recently proven formally in the random oracle model [41]. There have been a wide range of research covering a variety of different schemes for undeniable signatures in the literature [4,3,9,11,14,15,16,17,20,24,29,30,31,33,34,36,48,47,49,50].

The concept of convertible undeniable signatures was introduced by Boyar, Chaum, Damgård and Pedersen [4], where the convertibility refers to the ability of the signer to convert one or more his undeniable signatures into universally verifiable. "Convert" in the undeniable signatures has two types: **Selectively-Convert** and **Universally-Convert**. A selectively-convertible undeniable signature scheme allows the signer to convert an undeniable signature into a universally verifiable signature by releasing a **Selective Proof** in a later time. Then, one can check the validity of this signature using the selective proof and signer's public key. However, the validity of other undeniable signatures remains unknown and can only be verified via the confirmation/disavowal protocol with the help of the signer. Universally convertible refers that the signer has the additional ability to generate a universal proof which can finally convert *all* his undeniable signatures into universally verifiable signatures. Thus, one can check the validity of any undeniable signature without requiring any help from the signer.

1.1 Previous Works

The first convertible undeniable signature scheme proposed in [4] has been broken by Michels, Petersen and Horster [34] who proposed a repaired version with heuristic security. In Eurocrypt'96, Damgård and Pedersen [9] proposed two convertible undeniable signature schemes, in which forging signatures is provably equivalent to forging El Gamal signature. An efficient convertible undeniable signature based on Schnorr signature was proposed by Michels and Stadler in [35]. The new scheme can be used as a basis of an efficient extension to threshold signature. Other constructions in RSA systems were also introduced. The first RSA based (convertible) undeniable signature was proposed by Gennaro, Rabin and Krawczyk in CRYPTO'97 [16], which was later improved by Miyazaki [33]. Very recently, Kurosawa and Takagi [26] proposed a new approach for constructing selectively-convertible undeniable signature schemes, and presented two schemes based on RSA related assumptions. Furthermore, Kurosawa and Takagi's second scheme is the first selectively-convertible scheme whose security can be proven without random oracles. Based on the computation of characters, Monnerat and Vaudenay proposed a novel construction of undeniable signature which offers the advantage of having an arbitrarily short signature (depending on the required security level) [36]. Monnerat and Vaudenay also generalized and optimized their scheme in [37] and [38], respectively, and claimed that their scheme proposed in [37] can achieve the selective convertibility, without providing a formal security proof to support this claim. Laguillaumie and Vergnaud proposed a new (time-selective) convertible undeniable signature scheme from pairing [31] which a short signature length. Very recently, Huang *et al.* [18] presented a short convertible undeniable proxy signature from pairings. The first construction of identity based selectively-convertible undeniable signature was proposed by Libert and Quisquater. Fig. 1 summarizes the known convertible undeniable signatures.

Scheme	**Selectively-Convert**	**Universally-Convert**
Boyar-Chaum-Damgård-Pedersen's [4]	✓	✓
Damgård-Pedersen's [9]	✓	✓
Michels-Petersen-Horster's [34]	✓	✓
Michels-Stadler's [35]	✓	✓
Gennaro-Rabin-Krawczyk's [17]	✓	✓
Miyazaki's [33]	✓	✓
Libert and Quisquater's (ID-based) [29]	✓	
Monnerat-Vaudenay's [37]	✓	
Laguillaumie-Vergnaud's [31]	✓	✓
Kurosawa-Takagi's [26]	✓	
Huang *et al.*'s [18]	✓	✓

Fig. 1. Convertible Undeniable Signature Schemes in the Literature

There are two main challenges in the construction of universally-convertible undeniable signatures. The first one is how to generate the universal proof which can convert all undeniable signatures to be universally verifiable. As shown in the above table, some of the convertible undeniable signatures are not universally-convertible, and only selectively-convertible. It seems that "selectively-convertible" is *relatively easier* to achieve. Very recently, Kurosawa and Takagi showed the first example of selectively-convertible undeniable signature scheme [26], which is provably secure in the standard model. However, there is no universally-convertible undeniable signatures which is provably secure in the standard model. Therefore, it is worthwhile to find an efficient way to construct a universally-convertible undeniable signature scheme.

The other challenge is how to ensure the security of the universally-convertible undeniable signatures. From information theory aspect, a universal proof contains much more information than a selective proof, which might help the adversary to break the scheme. For example, Boyar-Chaum-Damgård-Pedersen's scheme [4] is unforgeable when the universal proof of their scheme is not published. However, it turns out to be insecure after the signer releases the universal proof. An adversary can generate a valid signature for any message after obtaining the universal proof. We can see there are several constructions of universally-convertible undeniable signatures with formal security analysis in the literature. However, most of them only consider the security of the basic undeniable signatures. That is, universal proofs of those schemes are not given to the adversaries, which might weaken their security claims.

1.2 Our Contributions

In this paper, we propose a *generic* construction for universally-convertible undeniable signatures which is based on the following three building blocks: (1) A strongly existentially unforgeable classic signature scheme, (2) A selectively-convertible undeniable signature scheme and (3) A collision-resistant hash function.

We provide a formal proof to show that our construction is strongly unforgeable against the adversary who even has the knowledge of the universal proof of our construction, assuming that the underlying classic signature scheme is strongly unforgeable and the hash function is collision resistant. We also prove that the resulting signatures of our construction are invisible if the underlying classic signature scheme is strongly unforgeable and the selectively-convertible undeniable signature scheme is invisible as well.

As one of the applications of our generic construction, we can obtain the *first* universally-convertible undeniable signature scheme in the *standard* model when certain building blocks are used. In addition, one can also fix and improve some known convertible undeniable signature schemes by applying our generic construction. We believe that the generic construction proposed in this paper is a useful tool for constructing other variants of undeniable signatures with universal convertibility, such as designated confirmer signatures, directed signatures and etc.

Organizations of the Paper
In the next section, we will review some preliminaries required throughout the paper. The outlines and security models of (universally) convertible undeniable signature are proposed in Section 3. In Section 4, we describe our generic construction of the universally-convertible undeniable signatures and its security analysis. Finally, Section 6 concludes this paper.

2 Preliminaries

2.1 Outline of Classic Signatures

A classic signature scheme **Classic-Signature** consists of the following algorithms:

CS-Setup: Given the system security number ℓ, this algorithm outputs the parameter $CS\text{-}Params$ which is shared by all the users in the system.

CS-KeyGen: Given the system parameters $CS\text{-}Params$, this algorithm outputs a public-secret key pair (PK_{CS}, SK_{CS}).

CS-Sign: Given a secret key SK_{CS}, $CS\text{-}Params$ and a message M to be signed, this algorithm outputs a publicly verifiable signature σ_{CS}.

CS-Verify: Given a message-signature pair (M, σ_{CS}), a public key PK_{CS} and $CS\text{-}Params$, this algorithm will check whether (M, σ_{CS}) is valid under the public key PK_{CS}. If it is, outputs Acc. Otherwise, Rej.

2.2 Strong Unforgeability of Classic Signatures

The strong existential unforgeability of **Classic-Signature** under an adaptive chosen-message attack is defined using the game in Fig. 2:

Setup: The challenger runs **CS-Setup** and **CS-KeyGen**. It gives the forger $\mathcal{F}_{CS}$ the resulting public key PK_{CS} and the parameters $CS\text{-}Params$. The challenger keeps the private key SK_{CS} for itself.

CS-Sign Queries: The forger $\mathcal{F}_{CS}$ can issue signature queries $\{M_1, M_2, \cdots, M_{q_S}\}$. To each message M_i, the challenger responds by running **CS-Sign** to generate a valid signature σ_{CS}^i and sending it to $\mathcal{F}_{CS}$ as the answer. These queries might be asked adaptively such that each message M_i may depend on the previously message-signature pairs $\{(M_1, \sigma_{CS}^1), (M_2, \sigma_{CS}^2), \cdots, (M_{i-1}, \sigma_{CS}^{i-1})\}$.

Output: Finally $\mathcal{F}_{CS}$ outputs a pair (M^*, σ_{CS}^*). The forger $\mathcal{F}_{CS}$ wins the game if

1. $Acc \leftarrow$ **CS-Verify**$(M^*, \sigma_{CS}^*, PK_{CS}, CS{-}Params)$ and
2. $(M^*, \sigma_{CS}^*) \notin \{(M_1, \sigma_{CS}^1), (M_2, \sigma_{CS}^2), \cdots, (M_{q_S}, \sigma_{CS}^{q_S})\}$.

Fig. 2. Strong Unforgeability of **Classic-Signature**

We define the advantage of an adversary $\mathcal{F}_{CS}$ in attacking the classic signature scheme **Classic-Signature** as the probability that $\mathcal{F}_{CS}$ wins the game in Fig. 2, taken over the random bits of the challenger and the adversary.

Definition 1. *A classic signature scheme* **Classic-Signature** *is (t, q_S, ϵ)-strongly existentially unforgeable under an adaptive chosen-message attack if no t-time forger $\mathcal{F}_{CS}$ making at most q_S signature queries has advantage at least ϵ in the game in Fig. 2.*

Remark: The adversary can also have access to the random oracles if necessary. It is also the same for the remaining security definitions.

Please refer to [12,2,45,46,27] for how to obtain a strongly existentially unforgeable classic signature scheme.

2.3 Collision-Resistant Hashing

Let $\mathcal{H} = \{H_k\}$ be a keyed hash family of functions $H_k : \{0,1\}^* \rightarrow \{0,1\}^n$ indexed by $k \in \mathcal{K}$. We say that algorithm $\mathcal{A}$ has advantage ϵ in breaking the collision-resistant of function $\mathcal{H}$ if:

$$\Pr[\mathcal{F}(k) = (m_0, m_1) : m_0 \neq m_1, H_k(m_0) = H_k(m_1)] \geq \epsilon,$$

where the probability is over the random choice of $k \in \mathcal{K}$ and the random bits of $\mathcal{A}$.

Definition 2. *A hash family $\mathcal{H}$ is (t, ϵ)-collision-resistant if no t-time adversary has advantage at least ϵ in breaking the collision-resistance of $\mathcal{H}$.*

3 Definitions of Undeniable Signatures

In this section, we will describe the definitions of the universally-convertible undeniable signatures and selectively undeniable signatures, which are denoted by **UC-Undeniable-Signature** and **SC-Undeniable-Signature** respectively.

3.1 Outline of Universally-Convertible Undeniable Signatures

A universally-convertible undeniable signature scheme **UC-Undeniable-Signature** consists of the following algorithms:

UC-US-Setup: Given the system security number ℓ, this algorithm outputs the parameter *UC-US-Params* which is shared by all the users in the system.

US-KeyGen: Given the system parameters *UC-US-Params*, this algorithm outputs a public-secret key pair (PK_{UC}, SK_{UC}).

UC-US-Sign: Given a secret key SK_{UC}, *UC-US-Params* and a message M to be signed, this algorithm outputs an undeniable signature σ_{UC} such that the validity of the pair (M, σ) is not publicly verifiable.

UC-US-Verify: Given a message-signature pair (M, σ_{UC}), the signer's public-secret key (PK_{UC}, SK_{UC}) and *UC-US-Params*, this algorithm will check whether (M, σ_{UC}) is a qualified pair. If it is not a qualified one, a symbol $\perp$ will be returned. Otherwise, it will further check its validity using the secret key SK_{UC}. If it is, outputs *Valid*. Otherwise, *Invalid*.

UC-US-Confirmation: A protocol between the signer and verifier such that given a message-signature pair (M, σ_{UC}), a public key PK_{UC} and *UC-US-Params*, this protocol allows the signer to convince the verifier that the given message-signature pair is valid, with the knowledge of the corresponding secret key SK_{UC}.

UC-US-Disavowal: A protocol between the signer and verifier such that given a message-signature pair (M, σ_{UC}), a public key PK_{UC} and *UC-US-Params*, this protocol allows the signer to convince the verifier that the given message-signature pair is invalid, with the knowledge of the corresponding secret key SK_{UC}.

UC-US-SConvert: Given a qualified message-signature pair (M, σ_{UC}), the signer's public-secret key $(PK_{UC},\ SK_{UC})$ and *UC-US-Params*, this algorithm outputs a selective proof **SelectiveProof**$\{M, \sigma_{UC}, PK_{UC}\}$.

UC-US-SVerify: Given a message-signature pair (M, σ_{UC}) , a pubic key PK_{UC}, **SelectiveProof**$\{M, \sigma_{UC},\ PK_{UC}\}$ and *UC-US-Params*, this algorithm will check whether (M, σ_{UC}) is valid under the public key PK_{UC}. If it is, outputs *Acc*. Otherwise, *Rej*.

UC-US-UConvert: Given the signer's public-secret key (PK_{UC}, SK_{UC}) and *UC-US-Params*, this algorithm outputs a universal proof **UniversalProof**$\{PK_{UC}\}$.

UC-US-UVerify: Given *any* message-signature pair (M, σ_{UC}), a public key PK_{UC}, **UniversalProof**$\{PK_{UC}\}$ and *UC-US-Params*, this algorithm will check whether (M, σ_{UC}) is valid under the public key PK_{UC}. If it is, outputs *Acc*. Otherwise, *Rej*.

The above algorithms should satisfy the following three properties:

1. **Completeness and Soundness:** the **UC-US-Confirmation** and **UC-US-Disavowal** protocols and all the verify algorithms are complete and sound, where completeness means that valid (invalid) signatures can always proven to be valid (invalid), and soundness means that no valid (invalid) signature can proven to be invalid (valid).
2. **Non-Transferable:** a verifier participating in an execution of the **UC-US-Confirmation** and **UC-US-Disavowal** protocols does not obtain information that could be used to convince a third party about the validity/invalidity of a signature.
3. **Impersonation:** only the signer can execute the **UC-US-Confirmation** and **UC-US-Disavowal** protocols. Anyone else who does not have the knowledge of the secret key can not impersonate the signer to carry out these protocols.

3.2 Strong Unforgeability of UC-Undeniable-Signature

The strong existential unforgeability of **UC-Undeniable-Signature** under an adaptive chosen message attack is defined using the game which is similar in Fig. 2. The difference is that the forger is allowed to have the knowledge of the universal proof, which will help the forger to verify the validity of any message-signature pair. In addition, The forger can also obtain some selective proofs of certain message-signature pairs chosen by himself. It is formally defined using the game described in Fig. 3.

We define the advantage of an adversary $\mathcal{F}_{US}$ in attacking **UC-Undeniable-Signature** as the probability that $\mathcal{F}_{US}$ wins the above game, taken over the random bits of the challenger and the adversary.

Setup: The challenger runs **UC-US-Setup** and **UC-US-KeyGen**. It gives the forger $\mathcal{F}_{US}$ the resulting public key PK_{UC} and the parameters $UC\text{-}US\text{-}Params$. The challenger also generates the universal proof **UniversalProof**$\{PK_{UC}\}$ and sends it $\mathcal{F}_{US}$ as well.

US-Sign Queries: The forger $\mathcal{F}_{US}$ can adaptively issue up to q_S signature queries $\{M_1, M_2, \cdots, M_{q_S}\}$. To each message M_i, the challenger responds by running **UC-US-Sign** to generate a valid signature σ_{UC}^i and sending it to $\mathcal{F}_{CS}$ as the answer.

Selective-Conversion Queries: The forger $\mathcal{F}_{US}$ can issue up to q_{SC} selective-conversion queries $\{(M_1, \sigma_{UC}^1), (M_2, \sigma_{UC}^2), \cdots, (M_{q_{SC}}, \sigma_{UC}^{q_{SC}})\}$ which are adaptively chosen by himself. To each pair (M_i, σ_{UC}^i),

1. If it is a qualified message-signature pair, then the challenger responds by generating a valid **SelectiveProof**$\{M_i, \sigma_{UC}^i, PK_{SC}\}$ and sending it to $\mathcal{F}_{US}$ as the answer.
2. Otherwise, the symbol $\perp$ is returned which means (M_i, σ_{UC}^i) is not a qualified message-signature pair.

Output: Finally $\mathcal{F}_{US}$ outputs a pair (M^*, σ_{UC}^*). The forger $\mathcal{F}_{US}$ wins the game if

1. $Valid \leftarrow$ **UC-US-Verify**$(M^*, \sigma_{UC}^*, PK_{UC}, UC{-}US{-}Params)$ and
2. $(M^*, \sigma^*) \notin \{(M_1, \sigma_{UC}^1), (M_2, \sigma_{UC}^2), \cdots, (M_{q_S}, \sigma_{UC}^{q_S})\}$.

Fig. 3. Strong Unforgeability of **UC-Undeniable-Signature**

Definition 3. *A universally-convertible undeniable signature scheme* **UC-Undeniable-Signature** *is $(t, q_S, q_{SC}, \epsilon)$-strongly existentially unforgeable under an adaptive chosen-message attack if no t-time forger $\mathcal{F}_{US}$ making at most q_S signature queries, q_{SC} selective-conversion queries and has advantage at least ϵ in the game in Fig. 3.*

3.3 Invisibility of UC-Undeniable-Signature

Roughly speaking, the invisibility property requires that a valid message-signature pair is indistinguishable from other qualified pairs, without the help of the signer. It will be defined using the similar game in the Fig. 3. The only difference is that the signer's universal proof is not returned to the distinguisher.

Setup: The challenger runs **UC-US-Setup** and **UC-US-KeyGen**. It gives the distinguisher $\mathcal{D}$ the resulting public key PK_{UC} and the parameters $UC\text{-}US\text{-}Params$. The challenger keeps the private key SK_{UC} to itself.

Phase 1: In this phase, $\mathcal{D}$ can adaptively issue the following queries :

US-Sign Queries and **Selective-Conversion Queries**: The challenger responds the same as defined in Fig. 3.

Verify Queries: The distinguisher $\mathcal{D}$ can issue up to q_V verify queries $\{(M_1, \sigma_{UC}^1), (M_2, \sigma_{UC}^2), \cdots, (M_{q_V}, \sigma_{UC}^{q_V})\}$ where (M_i, σ_{UC}^i) can either be the message-signature pair returned as the answer to one of **US-Sign Queries**, or adaptively chosen by the distinguisher himself. To each message-signature pair (M_i, σ_{UC}^i), the challenger responds by first running the **UC-US-Verify** algorithm. If it is not a qualified message-signature pair, the symbol $\perp$ is returned. Otherwise, the challenger then responds based on whether a passive attack or an active/concurrent attack is mounted.

1. Active/Concurrent attack: The challenger executes the **UC-US-Confirmation** (**UC-US-Disavowal**) protocol with adversary (acting as a cheating verifier) if the verification result is $Valid$ ($Invalid$).
2. Passive attack: The challenger returns a transcript of **UC-US-Confirmation** protocol if the verification result is $Valid$. Otherwise, a transcript of **UC-US-Disavowal** protocol is returned.

Challenge: At the end of Phase 1, $\mathcal{D}$ will choose a message M^* with the restriction that M^* has not been issued as one of the **US-Sign** queries. The challenger responds by selecting a random coin $\gamma \in \{0, 1\}$. If $\gamma = 1$, the challenger runs the algorithm **UC-US-Sign** to generate a valid universally-convertible undeniable signature σ_{UC}^* of message M^*. Otherwise, σ_{UC}^* is randomly chosen such that (M^*, σ^*) is a qualified message-signature pair. In both cases, σ_{UC}^* is returned to $\mathcal{D}$ as the challenging signature.

Phase 2: In this phase, $\mathcal{D}$ can adaptively issue **US-Sign Queries**, **Selective-Conversion Queries** and **Verify Queries** with the restrictions that:

1. If **UC-US-Sign** is a deterministic algorithm, M^* cannot be issued as one of the **US-Sign Queries**.
2. (M^*, σ_{UC}^*) can not be issued as one of the **Verify Queries** or **Selective-Conversion Queries**.

The challenger will respond these queries as it does in Phase 1.

Output: Finally $\mathcal{D}$ outputs its guess γ'. The distinguisher $\mathcal{D}$ wins the game if $\gamma = \gamma'$.

Fig. 4. Invisibility of **UC-Undeniable-Signature**

It is formally defined in Fig. 4. The success probability that $\mathcal{D}$ outputs a correct guess is defined as $Succ_{\mathcal{D}}$. We define the advantage of an distinguisher $\mathcal{D}$ in attacking **UC-Undeniable-Signature** as $|Succ_{\mathcal{D}} - \frac{1}{2}|$, taken over the random bits of the challenger and the adversary.

Definition 4. *A universally-convertible undeniable signature scheme* **UC-Undeniable-Signature** *is $(t, q_S, q_{SC}, q_V, \epsilon)$-invisible under an adaptive chosen-message attack if no t-time distinguisher $\mathcal{D}$ making at most q_S signature queries, q_{SC} selective-conversion queries, q_V verify queries and has advantage at least ϵ in the game defined in Fig. 4.*

3.4 Definitions of Selectively-Convertible Undeniable Signatures

A selectively-convertible undeniable signature scheme **SC-Undeniable-Signature** consists of 8 algorithms: **SC-US-Setup, SC-US-KeyGen, SC-US-Sign, SC-US-Verify, SC-US-Confirmation, SC-US-Disavowal, SC-US-SConvert** and **SC-US-SVerify**. All these algorithms are basically similar to the corresponding ones in **UC-Undeniable-Signature** defined in Section 3.1, the only difference is that we add "**SC**" to distinguish it from the latter. Therefore, the system's parameters in **SC-Undeniable-Signature** is denoted by $SC\text{-}US\text{-}Params$, user's public-secret key pair is (PK_{SC}, SK_{SC}), a selectively-convertible undeniable signature is denoted by (M, σ_{SC}) and etc..

SC-Undeniable-Signature should also satisfy the three properties: Completeness and Soundness, Non-Transferable and Impersonation which are the same as defined in Section 3.1. The security notions Strongly Unforgeable and Invisibility can be defined similarly with some minor difference. Here, we only give the definition of the invisibility in **SC-Undeniable-Signature**.

Definition 5. *A selectively-convertible undeniable signature scheme* **SC-Undeniable-Signature** *is $(t, q_S, q_{SC}, q_V, \epsilon)$-invisible under an adaptive chosen-message distinguisher if no t-time distinguisher $\mathcal{D}$ making at most q_S signature queries, q_{SC} selective-conversion queries, q_V verify queries and has advantage at least ϵ.*

4 A Generic Construction of UC-Undeniable-Signature

In this section, we will describe our generic construction of the universally-convertible undeniable signature scheme **UC-Undeniable-Signature**. Our construction is based on the following three building blocks: a classic signature scheme **Classic-Signature** which is strongly unforgeable as defined in Definition 1, a hash function which is collision-resistant as defined in Definition 2 and a selectively undeniable signature scheme **SC-Undeniable-Signature** which is invisible as defined in Definition 5. Each algorithm of our generic construction is described as below:

UC-US-Setup: Given the system security number ℓ, this algorithm generates the system parameter $UC\text{-}US\text{-}Params = \{CS\text{-}Params, SC\text{-}US\text{-}Params,$

$H_k\}$, where $CS\text{-}Params$ is the parameters in the classic signature scheme **Classic-Signature** which is the output of **CS-Setup**(ℓ), $SC\text{-}US\text{-}Params$ is the parameters in the selectively-convertible undeniable signature scheme **SC-Undeniable-Signature** which is the output of **SC-US-Setup**(ℓ) and H_k is a random function in the collision-resistant keyed hash family $\mathcal{H}$.

UC-US-KeyGen: Each signer of a universally-convertible undeniable signature has two public-secret key pairs: (PK_{CS}, SK_{CS}) and (PK_{SC}, SK_{SC}) where

1. (PK_{CS}, SK_{CS}) is the public-secret key pair in the classic signature scheme **Classic-Signature** which is generated by the algorithm **CS-KeyGen**.
2. (PK_{SC}, SK_{SC}) is the public-secret key pair in the selective undeniable signature scheme **SC-Undeniable-Signature** which is generated by the algorithm **SC-US-KeyGen**.

The public key PK_{UC} is set as (PK_{CS}, PK_{SC}) and the secret key SK_{UC} is set as (SK_{CS}, SK_{SC}).

UC-US-Sign: The universally-convertible undeniable signature of the message M is $\sigma_{UC} = (\sigma_{SC}, \sigma_{CS})$ where

1. σ_{SC} is a selectively-convertible undeniable signature on the message M which is generated by the algorithm **SC-US-USign**:
   $\sigma_{SC} \leftarrow$ **SC-US-USign**$(M, SK_{SC}, SC{-}US{-}Params)$.
2. σ_{CS} is a classic signature on the message $H_k(M \| \sigma_{SC} \| Undeniable)$ which is generated by the algorithm **CS-Sign**:
   $\sigma_{CS} \leftarrow$ **CS-Sign**$(H_k(M \| \sigma_{SC} \| Undeniable), SK_{CS}, CS{-}Params)$[1].
   Here, the world *Undeniable* indicates that this signature is generated in the scenario of undeniable signature.

UC-US-Verify: Given a message-signature pair $(M, \sigma_{SC}, \sigma_{CS})$, this algorithm first checks whether σ_{CS} is a valid classic signature on $H_k(M \| \sigma_{SC} \| Undeniable)$.

1. If $Rej \leftarrow$**CS-Verify**$(H_k(M \| \sigma_{SC} \| Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$, then $(M, \sigma_{SC}, \sigma_{CS})$ is regarded as a non-qualified pair and the symbol $\perp$ is output.

   Here the definition of the "qualified pair" is different from the previous one. In most undeniable signature schemes, it refers to the message-signature pairs where the signature could be any element in the signature space. In this sense, the invisibility of the proposed construction is a little weaker than the traditional one, since we require that σ_{CS} must be a valid signature.
2. Otherwise, it further runs the algorithm **SC-US-Verify**$(M, \sigma_{SC}, SK_{SC}, SC{-}US{-}Params)$ and forwards its output.

[1] We note that σ_{CS} is *not* a classic (or, publicly verifiable) signature on the message M. Instead, it is a signature on the string "$\xi = M \| \sigma_{SC} \| Undeniable$". Since $\|$ denotes the concatenation of the bit strings, ξ corresponds to many different pairs (M^i, σ_{SC}^i) provided that $\xi = M^i \| \sigma_{SC}^i \| Undeniable$. Therefore, given the signature σ_{CS}, one cannot decide if the signer has actually signed the message M. In the proof of Theorem 1, we also discuss how to remove the message from the input of the hash function.

UC-US-Confirmation: Given a message-signature pair $(M, \sigma_{SC}, \sigma_{CS})$, the verifier first runs the algorithm **CS-Verify**$(H_k(M\|\sigma_{SC}\|Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$.

1. If it outputs Rej, nothing is to be carried out between the verifier and the signer.
2. Otherwise, the verifier will execute the **SC-US-Confirmation** protocol with the signer.

UC-US-Disavowal: Given a message-signature pair $(M, \sigma_{SC}, \sigma_{CS})$, the verifier first runs the algorithm **CS-Verify**$(H_k(M\| \ \sigma_{SC}\|Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$.

1. If it outputs Rej, nothing is to be carried out between the verifier and signer.
2. Otherwise, the verifier will execute the **SC-US-Disavowal** protocol with the signer.

UC-US-SConvert: Given a pair $(M, \sigma_{SC}, \sigma_{CS})$, it runs the algorithm **CS-Verify** $(H_k(M\|\sigma_{SC}\|Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$.

1. If it outputs Rej, the symbol $\perp$ is output, which means $(M, \sigma_{SC}, \sigma_{CS})$ is not a qualified pair.
2. Otherwise, it runs the algorithm **SC-US-SConvert**$(M, \sigma_{SC}, PK_{SC}, SK_{SC}, SC{-}US{-}Params)$ to generate **SelectiveProof**$\{M, \sigma_{SC}, PK_{SC}\}$.

UC-US-SVerify: Given a pair $(M, \sigma_{SC}, \sigma_{CS})$, and its selective proof **SelectiveProof**$\{M, \sigma_{SC}, PK_{SC}\}$, this algorithm outputs Acc if $Acc \leftarrow$ **CS-Verify**$(H_k(M\|\sigma_{SC}\|Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$ and $Acc \leftarrow$ **SC-US-SVerify**$(M, \sigma_{SC},$ **SelectiveProof**$\{M, \sigma_{SC}, PK_{SC}\}, PK_{SC}, SC{-}US{-}Params)$.

Otherwise, outputs Rej.

UC-US-UConvert: This algorithm outputs SK_{SC} as the universal proof **Universal**$\{PK_{SC}\}$.

UC-US-UVerify: Given a pair $(M, \sigma_{SC}, \sigma_{CS})$, and the universal proof SK_{SC}, this algorithm outputs Acc if

$Acc \leftarrow$ **CS-Verify**$(H_k(M\|\sigma_{SC}\|Undeniable), \sigma_{CS}, PK_{CS}, CS{-}Params)$ and

$$Acc \leftarrow \textsf{\textbf{SC-US-Verify}}(M, \sigma_{SC}, SK_{SC}, SC{-}US{-}Params).$$

Otherwise, outputs Rej.

5 Security Analysis

In this section, we will give a security analysis of our generic construction. Our generic construction will directly satisfy the properties: **Completeness and Soundness**, **Non-Transferable** and **Impersonation** if the underlying building blocks satisfy those properties as well. Due to the page limitation, we will skip the analysis of those properties and focus on the the unforgeability and invisibility of our construction.

5.1 Strong Unforgeability of Our Generic Construction

Theorem 1. *Our proposed universally-convertible undeniable signature scheme* **UC-Undeniable-Signature** *is $(t, q_S, q_{SC}, \epsilon)$-strongly existentially unforgeable assuming the underlying classic signature scheme* **Classic-Signature** *is $(t, q_S, \epsilon/2)$-strongly existentially unforgeable and $\mathcal{H}$ is $(t, \epsilon/2)$-collision-resistant.*

Proof. Suppose there is a forger $\mathcal{F}_{US}$ that $(t, q_S, q_{SC}, \epsilon)$ breaks strong unforgeability of our generic construction proposed in Section 4, then we will show there exists an algorithm $\mathcal{A}$ who can either $(t, q_S, \epsilon/2)$-break the strong unforgeability of the underlying **Classic-Signature** or $(t, \epsilon/2)$-break the collision-resistance of $\mathcal{H}$. Our proof will use the similar techniques in [2].

As defined in Fig 3, $\mathcal{F}_{US}$ can obtain the target public key (PK_{CS}, PK_{SC}), the parameters (CS-$Params$, SC-US-$Params$, H_k) and the universal proof SK_{SC}.

$\mathcal{F}_{US}$ can adaptively choose message M_i and is given corresponding signature $(M_i, \sigma^i_{SC}, \sigma^i_{CS})$. Let $\mathbb{S} = \{(M_i, \sigma^i_{SC}, \sigma^i_{CS})\}$ be the set of message-signature pairs generated during the **US-Sign** queries. In our construction, the selective proof of a message-signature pair is generated by using SK_{SC} which has been already sent to $\mathcal{F}_{US}$. Therefore, $\mathcal{F}_{US}$ himself can generate the selective proof of any message-signature pair and does not need to issue the **Selective-Conversion Queries** any more. After all the queries, $\mathcal{F}_{US}$ will output a forgery $(M^*, \sigma^*_{SC}, \sigma^*_{CS}) \notin \mathbb{S}$. This forgery must fall into one of the following two types:

Type I: For $\forall(M_i, \sigma^i_{SC}, \sigma^i_{CS}) \in \mathbb{S}$, $(H_k(M_i \| \sigma^i_{SC} \| Undeniable), \sigma^i_{CS}) \neq (H_k(M^* \| \sigma^*_{SC} \| Undeniable), \sigma^*_{CS})$.

Type II: There exists at least one tuple $(M_i, \sigma^i_{SC}, \sigma^i_{CS}) \in \mathbb{S}$ such that $(H_k(M_i \| \sigma^i_{SC} \| Undeniable), \sigma^i_{CS}) = (H_k(M^* \| \sigma^*_{SC} \| Undeniable), \sigma^*_{CS})$.

We will show later that the Type I forgery can be used to break the strong unforgeability of the underlying classic signature scheme **Classic-Signature** and Type II forgery can be used to find a collision of $\mathcal{H}$. The simulation will be different due to different forgeries considered. At the beginning, the algorithm $\mathcal{A}$ will flip a *coin* in $\{1, 2\}$. If $coin = 1$, $\mathcal{A}$ will guess that Type I forgery will be the output of $\mathcal{F}_{US}$. Otherwise, Type II forgery will be produced.

Type I: Suppose $\mathcal{F}_{US}$ is a Type I forger who can $(t, q_S, q_{SC}, \epsilon)$-break strong unforgeability of our generic construction. We will construct an algorithm $\mathcal{A}$ that can (t, q_S, ϵ)-break the strong unforgeability of the underlying **Classic-Signature**. At the beginning, $\mathcal{A}$ is given a public key PK_{CS} and the parameter CS-$Params$. $\mathcal{A}$ will answer $\mathcal{F}_{US}$'s queries as described below:

Setup: $\mathcal{A}$ generates SC-US-$Params$ by running the algorithm **SC-US-Setup**(ℓ). Then, it runs the algorithm **SC-US-KeyGen** to generate the public-secret key pair (PK_{SC}, SK_{SC}). It also chooses a random hash function H_k in the collision-resistant keyed hash family $\mathcal{H}$. At last, $\mathcal{A}$ returns $(PK_{CS}, PK_{SC}, SK_{SC})$, CS-$Params$, SC-US-$Params$ and H_k to $\mathcal{F}_{US}$.

US-Sign Queries: For a sign query M_i from $\mathcal{F}_{US}$, $\mathcal{A}$ responds as followings:

1. $\mathcal{A}$ first runs the algorithm **SC-US-Sign** using the secret key SK_{SC} to generate σ^i_{SC}.

2. $\mathcal{A}$ then sets $H_k(M_i \| \sigma_{SC}^i \| Undeniable)$ as his own **CS-Sign** query. As the model defined in Fig 2, a valid signature σ_{CS}^i will be returned to $\mathcal{A}$.

At last, $\mathcal{A}$ will return $(\sigma_{SC}^i, \sigma_{CS}^i)$ as the answer.

Selective-Conversion Queries: As we have explained earlier, $\mathcal{F}_{US}$ does not need to issue these queries since the knowledge SK_{SC} enables him to generate the selective proof of our generic construction.

After all the queries, $\mathcal{F}_{US}$ will output a Type I forgery $(M^*, \sigma_{SC}^*, \sigma_{CS}^*) \notin \mathbb{S}$ such that $(H_k(M^* \| \sigma_{SC}^* \| Undeniable), \sigma_{CS}^*) \neq (H_k(M_i \| \sigma_{SC}^i \| Undeniable), \sigma_{CS}^i)$ for $\forall (M_i, \sigma_{SC}^i, \sigma_{CS}^i) \in \mathbb{S}$.

With probability at least ϵ, it is a valid message-signature pair of our proposed construction. Thus, $Acc \leftarrow$ **CS-Verify**$(H_k(M^* \| \sigma_{SC}^* \| Undeniable), \sigma_{CS}^*, PK_{CS}, CS{-}Params)$. Note that the pair $(H_k(M^* \| \sigma_{SC}^* \| Undeniable), \sigma_{CS}^*)$ is not generated during $\mathcal{A}$'s **CS-Sign Queries**. Thus, $(H_k(M^* \| \sigma_{SC}^* \| Undeniable), \sigma_{CS}^*)$ is a valid forgery of the underlying **Classic-Signature** as defined in Fig 2.

Type II: Suppose $\mathcal{F}_{US}$ is a Type II forger who can $(t, q_S, q_{SC}, \epsilon)$-break strong unforgeability of our generic construction. We will construct an algorithm $\mathcal{A}$ that can (t, ϵ)-break the collision-resistance of $\mathcal{H}$. Algorithm $\mathcal{A}$ is given a random key $k \in \mathcal{K}$. Its goal is to output a pair of messages (m_1, m_2) such that $m_1 \neq m_2$ and $H_k(m_1) = H_k(m_2)$. $\mathcal{A}$ will answer $\mathcal{F}_{US}$'s queries as described below:

Setup: $\mathcal{A}$ generates $CS\text{-}Params$, $SC\text{-}US\text{-}Params$, (PK_{SC}, SK_{SC}) and (PK_{CS}, SK_{CS}) by running the corresponding algorithms defined in Section 4. It then returns $(PK_{SC}, PK_{CS}, SK_{SC}, CS\text{-}Params, SC\text{-}US\text{-}Params, H_k)$ to $\mathcal{F}_{US}$. $\mathcal{A}$ keeps SK_{CS} as secret to himself.

US-Sign Queries: To each sign query, $\mathcal{A}$ runs the algorithm **UC-US-Sign** using the secret keys SK_{SC} and SK_{CS}.

After all the queries, $\mathcal{F}_{US}$ will output a Type II forgery $(M^*, \sigma_{SC}^*, \sigma_{CS}^*) \notin \mathbb{S}$ and there exists at least one tuple $(M_i, \sigma_{SC}^i, \sigma_{CS}^i) \in \mathbb{S}$ such that $(H_k(M_i \| \sigma_{SC}^i \| Undeniable), \sigma_{CS}^i) = (H_k(M^* \| \sigma_{SC}^* \| Undeniable), \sigma_{CS}^*)$. Thus, $(M_i, \sigma_{SC}^i) \neq (M^*, \sigma_{SC}^*)$ due to the requirement that $(M^*, \sigma_{SC}^*, \sigma_{CS}^*) \notin \mathbb{S}$. As the assumption in [2], we require that any selectively undeniable signature σ_{SC} has a unique encoding. Therefore, $\mathcal{A}$ successfully find the collision $(M_i \| \sigma_{SC}^i \| Undeniable, M^* \| \sigma_{SC}^* \| Undeniable)$ of $\mathcal{H}$[2].

In summary, we have showed how to use $\mathcal{F}_{US}$ to find a new message-signature pair of the underlying classic signature scheme **Classic-Signature** or a collision of $\mathcal{H}$. □

[2] This explains why σ_{CS} must be a classic signature on $H_k(M \| \sigma_{SC} \| Undeniable)$. If we remove the message M from the input of hash function H_k, then the unforgeability of our construction relies on a stronger assumption: Given the signing key of the **SC-Undeniable-Signature** scheme, it is impossible for an adversary to find two different messages which share the same selectively convertible undeniable signature. There is no evidence shows that all **SC-Undeniable-Signature** schemes satisfy this requirement.

Remark: As one can see from the above analysis, the unforgeability of the proposed construction does not rely on the unforgeability of the underlying undeniable signature scheme. This is due to the fact the signer could publish his secret key of the **SC-Undeniable-Signature** as the universal proof.

5.2 Invisibility of Our Generic Construction

Theorem 2. *Our proposed universally-convertible undeniable signature scheme* **UC-Undeniable-Signature** *is* $(t, q_S, q_{SC}, q_V, \epsilon)$*-invisible assuming the underlying classic signature scheme* **Classic-Signature** *is* (t, q_S, ϵ')*-strongly existentially unforgeable and the selectively-convertible undeniable signature scheme* **SC-Undeniable-Signature** *is* $(t, q_S, q_{SC}, q_V, \epsilon \cdot (1-\epsilon')^{q_V+q_{SC}})$*-invisible.*

Proof. Suppose there is a distinguisher $\mathcal{D}_{UC}$ that $(t, q_S, q_{SC}, q_V, \epsilon)$-breaks the invisibility of our generic construction proposed in Section 4, then we will show there exists an algorithm $\mathcal{D}_{SC}$ who can $(t, q_{SC}, q_{SC}, q_V, (1-\epsilon')^{q_V+q_{SC}})$-break the invisibility of **SC-Undeniable-Signature** if **Classic-Signature** is (t, q_S, ϵ')-strongly existentially unforgeable.

At the beginning, $\mathcal{D}_{SC}$ receives the public key PK_{SC} and $SC\text{-}US\text{-}Params$ of **SC-Undeniable-Signature**. $\mathcal{D}_{SC}$ will answer $\mathcal{D}_{UC}$'s queries as described below:

Setup: $\mathcal{D}_{SC}$ generates $CS\text{-}Params$ by running the algorithm **CS-Setup**(ℓ). Then, he runs the algorithm **CS-KeyGen** to obtain the key pair (PK_{CS}, SK_{CS}). He also chooses a random hash function $H_k \in \mathcal{H}$. At last, $\mathcal{D}_{SC}$ returns $(PK_{CS}, PK_{SC}$, $CS\text{-}Params$, $SC\text{-}US\text{-}Params$, $H_k)$ to $\mathcal{D}_{UC}$.

US-Sign Queries: For a sign query M_i from $\mathcal{D}_{UC}$, $\mathcal{D}_{SC}$ responds as following:

1. $\mathcal{D}_{SC}$ first issues M_i as one of the **US-Sign Queries** to his own challenger and obtains the selectively-convertible undeniable signature σ^i_{SC}.
2. $\mathcal{D}_{SC}$ generates the signature σ^i_{CS} for $H_k(M_i \| \sigma^i_{SC} \| Undeniable)$ by running the algorithm **CS-Sign** with the knowledge SK_{CS}.

At last, $\mathcal{D}_{SC}$ returns $(\sigma^i_{SC}, \sigma^i_{UC})$ to $\mathcal{D}_{UC}$ as the answer.

Selective-Conversion Queries: For a selective-conversion query $(M_i, \sigma^i_{SC}, \sigma^i_{CS})$, $\mathcal{D}_{SC}$ firstly runs the algorithm **CS-Verify**$(H_k(M_i \| \sigma^i_{SC} \| Undeniable)$, σ^i_{CS}, $PK_{CS}, CS{-}Params)$.

1. If it outputs Rej, the symbol $\perp$ is returned which means $(M_i, \sigma^i_{SC}, \sigma^i_{CS})$ is not a qualified pair.
2. Otherwise, $\mathcal{D}_{SC}$ sets (M_i, σ^i_{SC}) as his own selective-conversion query and issues it to his challenger. $\mathcal{D}_{SC}$ will obtain **SelectiveProof**$\{M_i, \sigma^i_{SC}, PK_{SC}\}$ from its own challenger. Then, he returns it to $\mathcal{D}_{UC}$ as the answer.

Verify Queries: For each verify query $(M_i, \sigma^i_{SC}, \sigma^i_{CS})$, $\mathcal{D}_{SC}$ firstly runs the algorithm **CS-Verify** $(H_k(M_i \| \sigma^i_{SC} \| Undeniable), \sigma^i_{CS}, PK_{CS}, CS{-}Params)$. If it outputs Rej, the symbol $\perp$ is returned which means $(M_i, \sigma^i_{SC}, \sigma^i_{CS})$ is not a qualified pair. Otherwise, $\mathcal{D}_{SC}$ will respond as following:

1. For an active/concurrent attack, $\mathcal{D}_{SC}$ must execute the Confirmation (Disavowal) protocol with $\mathcal{D}_{UC}$. It will act as the middle-man in the sense that $\mathcal{D}_{SC}$ will forward each $\mathcal{D}_{UC}$'s query in the protocol as his own query and return each response from his challenger to $\mathcal{D}_{UC}$.

2. For a passive attack, $\mathcal{D}_{SC}$ will issue (M_i, σ^i_{SC}) as one of his **Verify Queries** to his challenger. $\mathcal{D}_{SC}$ will obtain a transcript of the Confirmation/Disavowal protocol. Then, he returns that transcript to $\mathcal{D}_{UC}$.

Challenging: At the end of Phase 1, $\mathcal{D}_{UC}$ will output a challenging message M^*. $\mathcal{D}_{SC}$ will forward M^* as his own challenging message and obtain the challenging signature σ^*_{SC}. Then, $\mathcal{D}_{SC}$ runs the algorithm **CS-Sign** with SK_{CS} and generates the signature σ^*_{CS}. At last, $\mathcal{D}_{SC}$ returns the challenging signature $(\sigma^*_{SC}, \sigma^*_{CS})$ to $\mathcal{D}_{UC}$.

Phase 2: $\mathcal{D}_{UC}$ can continue to issue queries as defined in Fig. 4 and $\mathcal{D}_{SC}$ can answer these queries as described previously. In addition, There might be some special queries $(M^*, \sigma^*_{SC}, \sigma^\dagger_{CS})$ during Phase 2. In these queries, the first two parts M^* and σ^*_{SC} are the same as those in the challenging signature, but $\sigma^\dagger_{CS} \neq \sigma^*_{CS}$. We say these queries are special since $\mathcal{D}_{UC}$ is allowed to issue these queries as one of the **Verify Queries** or **Selective-Conversion Queries**, but $\mathcal{D}_{UC}$ is not allowed to issue (M^*, σ^*_{SC}) as his own query. So, $\mathcal{D}_{SC}$ can not use his own challenger to respond these queries. For each special query, $\mathcal{D}_{SC}$ will act as described below. When $(M^*, \sigma^*_{SC}, \sigma^\dagger_{CS})$ is issued by $\mathcal{D}_{UC}$, $\mathcal{D}_{SC}$ firstly runs the algorithm **CS-Verify**$(H_k(M^* \| \sigma^*_{SC} \| Undeniable), \sigma^\dagger_{CS}, PK_{CS}, CS{-}Params)$.

1. It outputs Rej, the symbol $\perp$ is returned because $(M^*, \sigma^*_{SC}, \sigma^\dagger_{CS})$ is not a qualified pair.
2. Otherwise, it outputs Acc and $\mathcal{D}_{SC}$ will abort. However, if the algorithm **CS-Verify** outputs Acc, then $\sigma^\dagger_{CS}$ and σ^*_{CS} will be two different valid signatures of the same message $H_k(M^* \| \sigma^*_{SC} \| Undeniable)$. Due to the strong unforgeability of **Classic-Signature**, the probability that $\mathcal{D}_{UC}$ can find out the new pair $(H_k(M^* \| \sigma^*_{SC} \| Undeniable), \sigma^\dagger_{CS})$ is at most ϵ'.

If $\mathcal{D}_{SC}$ does not abort during the simulation, then $\mathcal{D}_{UC}$ will output his guess γ' which is correct with advantage ϵ. $\mathcal{D}_{SC}$ will forward γ' as his own guess. It is obvious that if $(M^*, \sigma^*_{SC}, \sigma^*_{CS})$ is a valid message-signature pair of our generic scheme, then (M^*, σ^*_{SC}) will be valid of **SC-Undeniable-Signature** as well. Thus, If $\mathcal{D}_{SC}$ does not abort during the simulation, $\mathcal{D}_{SC}$ can also output a correct guess with the same advantage ϵ. We now go to compute the probability that $\mathcal{D}_{SC}$ does not abort during the simulation. If the underlying **Classic-Signature** is (t, q_S, ϵ')-strong unforgeable, then $\mathcal{D}_{SC}$ could abort with probability at most ϵ' for each verify query or selective-conversion query. Therefore, the probability that $\mathcal{D}_{SC}$ does not abort during the simulation is at least $(1-\epsilon')^{q_V+q_{SC}}$. Thus, the advantage that $\mathcal{D}_{SC}$ can break the invisibility of the underlying **SC-Undeniable-Signature** scheme with advantage at least $\epsilon \cdot (1-\epsilon')^{q_V+q_{SC}}$ which contradicts the assumption that **SC-Undeniable-Signature** is $(t, q_S, q_{SC}, q_V, \epsilon \cdot (1-\epsilon')^{q_V+q_{SC}})$-invisible. □

5.3 Applications

A direct application of our generic construction is the first provably secure universally-convertible undeniable signature scheme in the standard model. It can be constructed by a strongly existentially unforgeable **Classic-Signature** in the standard model (e.g. BB's scheme [1]) and an invisible selectively-convertible undeniable signature scheme **SC-Undeniable-Signature** [26] in the standard model. In addition, we can fix Boyar-Chaum-Damgård-Pedersen's scheme [4] by applying a strongly unforgeable **Classic-Signature**. We also believe that the ideas in our generic construction can be used for other variants of undeniable signatures with universal convertibility, such as designated confirmer signatures [5], directed signatures [28] and etc. Due to the page limitation, we cannot show the details to these constructions.

6 Conclusion

We introduced a generic construction for universally-convertible undeniable signatures. Our construction uses a strongly existentially unforgeable classic signature scheme, an invisible selectively undeniable signature scheme and a collision-resistant hash function as the building blocks. The security of the proposed construction is formally analyzed, which is tightly related to the security of underlying build blocks. When applying this construction to certain specific schemes, we can obtain some useful results. One of these applications is the first universally-convertible undeniable signature scheme in the standard model.

References

1. Boneh, D., Boyen, X.: Short Signatures without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 382–400. Springer, Heidelberg (2004)
2. Boneh, D., Shen, E., Waters, B.: Strongly Unforgeable Signatures based on Computational Diffie-Hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer, Heidelberg (2006)
3. Biehl, I., Paulus, S., Takagi, T.: Efficient Undeniable Signature Schemes Based on Ideal Arithmetic in Quadratic Orders. In: Designs, Codes and Cryptography, vol. 31(2), pp. 99–123. Springer, Netherlands (2004)
4. Boyar, J., Chaum, D., Damgård, I.B., Pedersen, T.P.: Convertible Undeniable Signatures. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 189–205. Springer, Heidelberg (1991)
5. Chaum, D.: Designated Confirmer Signatures. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 86–91. Springer, Heidelberg (1995)
6. Chaum, D., van Antwerpen, H.: Undeniable Signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 212–216. Springer, Heidelberg (1990)
7. Chaum, D.: Zero-Knowledge Undeniable Signatures (Extended Abstract). In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 458–464. Springer, Heidelberg (1991)
8. Diffie, W., Hellman, M.: New directions in cryptography. IEEE IT 22, 644–654 (1976)
9. Damgård, I.B., Pedersen, T.P.: New Convertible Undeniable Signature Schemes. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 372–386. Springer, Heidelberg (1996)

10. Desmedt, Y., Yung, M.: Weaknesses of Undeniable Signature Schemes (Extended Abstract). In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 205–220. Springer, Heidelberg (1991)
11. Fujioka, A., Okamotoa, T., Ohta, K.: Interactive Bi-Proof Systems and Undeniable Signature Schemes. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 243–256. Springer, Heidelberg (1991)
12. Goldreich, O.: Foundations of Cryptography, Basic Applications, vol. II. Cambridge University Press, Cambridge (2004)
13. Goldwasser, S., Micali, S., Rivest, R.: A Digital signature scheme secure against adaptively chosen message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
14. Galbraith, S.D., Mao, W., Paterson, K.G.: RSA-Based Undeniable Signatures for General Moduli. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 200–217. Springer, Heidelberg (2002)
15. Galbraith, S.D., Mao, W.: Invisibility and Anonymity of Undeniable and Confirmer Signatures. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 80–97. Springer, Heidelberg (2003)
16. Gennaro, R., Krawczyk, H., Rabin, T.: RSA-Based Undeniable Signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 132–149. Springer, Heidelberg (1997)
17. Gennaro, R., Rabin, T., Krawczyk, H.: RSA-Based Undeniable Signatures. Journal of Cryptology 13(4), 397–416 (2000)
18. Huang, X., Mu, Y., Susilo, W., Wu, W.: Provably Secure Pairing-based Convertible Undeniable Signature with Short Signature Length. In: Pairing 2007. LNCS, vol. 4575, pp. 367–391. Springer, Heidelberg (2007)
19. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated Verifier Proofs and Their Applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
20. Jongkook, L., Shiryong, R., Jeungseop, K., Keeyoung, Y.: A New Undeniable Signature Scheme Using Smart Cards. In: Honary, B. (ed.) Cryptography and Coding. LNCS, vol. 2260, pp. 387–394. Springer, Heidelberg (2001)
21. Jakobsson, M.: Blackmailing Using Undeniable Signatures. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 425–427. Springer, Heidelberg (1995)
22. Furukawa, J., Kurosawa, K., Imai, H.: An Efficient Compiler from Σ-Protocol to 2-Move Deniable Zero-Knowledge. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 46–57. Springer, Heidelberg (2006)
23. Kudla, C., Paterson, K.G.: Non-interactive Designated Verifier Proofs and Undeniable Signatures. In: Smart, N.P. (ed.) Cryptography and Coding. LNCS, vol. 3796, pp. 136–154. Springer, Heidelberg (2005)
24. Kim, S., Won, D.: Threshold Entrusted Undeniable Signature. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 195–203. Springer, Heidelberg (2005)
25. Kurosawa, K., Heng, S-H.: 3-Move Undeniable Signature Scheme. In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 181–197. Springer, Heidelberg (2005)
26. Kurosawa, K., Takagi, T.: New Approach for Selectively Convertible Undeniable Signature Schemes. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 428–443. Springer, Heidelberg (2006)
27. Huang, Q., Wong, D.S., Zhao, Y.: Generic Transformation to Strongly Unforgeable Signatures. ACNS 2007, Available online `http://eprint.iacr.org/2006/346`

28. Laguillaumie, F., Paillier, P., Vergnaud, D.: Universally Convertible Directed Signatures. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 682–701. Springer, Heidelberg (2005)
29. Libert, B., Quisquater, J.-J.: Identity Based Undeniable Signatures. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 112–125. Springer, Heidelberg (2004)
30. Lyuu, Y.-D., Wu, M.-L.: Convertible Group Undeniable Signatures. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 48–61. Springer, Heidelberg (2003)
31. Laguillaumie, F., Vergnaud, D.: Time-Selective Convertible Undeniable Signatures. In: Menezes, A.J. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 154–171. Springer, Heidelberg (2005)
32. Laguillaumie, F., Vergnaud, D.: Short Undeniable Signatures Without Random Oracles: The Missing Link. In: Maitra, S., Madhavan, C.E.V., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 283–296. Springer, Heidelberg (2005)
33. Miyazaki, T.: An Improved Scheme of the Gennaro-Krawczyk-Rabin Undeniable Signature System Based on RSA. In: Won, D. (ed.) ICISC 2000. LNCS, vol. 2015, pp. 135–149. Springer, Heidelberg (2001)
34. Michels, M., Petersen, H., Horster, P.: Breaking and Repairing a Convertible Undeniable Signature Scheme. In: Third ACM Conference on Computer and Communications Security, pp. 148–152. ACM Press, New York (1996)
35. Michels, M., Stadler, M.: Efficient Convertible Undeniable Signature Schemes. In: SAC 1997. The 4th International Workshop on Selected Areas in Cryptography, pp. 231–244 (1997)
36. Monnerat, J., Vaudenay, S.: Undeniable Signatures Based on Characters: How to Sign with One Bit. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 69–85. Springer, Heidelberg (2004)
37. Monnerat, J., Vaudenay, S.: Generic Homomorphic Undeniable Signatures. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 354–371. Springer, Heidelberg (2004)
38. Monnerat, J., Vaudenay, S.: Optimization of the MOVA Undeniable Signature Scheme. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 196–209. Springer, Heidelberg (2005)
39. Monnerat, J., Vaudenay, S.: Short 2-Move Undeniable Signatures. In: Nguyen, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 19–36. Springer, Heidelberg (2006)
40. National Institute of Standards and Technology (NIST). Digital Signature Standard (DSS). Federal Information Processing Standards Publication 186-2 (January 2000)
41. Ogata, W., Kurosawa, K., Heng, S.-H.: The Security of the FDH Variant of Chaum's Undeniable Signature Scheme. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 328–345. Springer, Heidelberg (2005)
42. Okamoto, T., Pointcheval, D.: The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
43. Pointcheval, D.: Self-Scrambling Anonymizers. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 259–275. Springer, Heidelberg (2001)
44. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)
45. Steinfeld, R., Pieprzyk, J., Wang, H.: How to Strengthen Any Weakly Unforgeable Signature into a Strongly Unforgeable Signature. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 357–371. Springer, Heidelberg (2006)

46. Teranishi, I., Oyama, T., Ogata, W.: General Conversion for Obtaining Strongly Existentially Unforgeable Signatures. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 191–205. Springer, Heidelberg (2006)
47. Wang, G.: An Attack on Not-interactive Designated Verifier Proofs for Undeniable Signatures, Available online http://eprint.iacr.org/2003/243
48. Wang, G., Qing, S., Wang, M., Zhou, Z.: Threshold Undeniable RSA Signature Scheme. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001. LNCS, vol. 2229, pp. 221–232. Springer, Heidelberg (2001)
49. Wang, G., Zhou, J., Deng, R.H.: On the Security of the Lee-Hwang Group-Oriented Undeniable Signature schemes. In: Katsikas, S.K., Lopez, J., Pernul, G. (eds.) TrustBus 2004. LNCS, vol. 3184, pp. 289–298. Springer, Heidelberg (2004), Available online http://eprint.iacr.org/2002/150
50. Zhang, F., Safavi-Naini, R., Susilo, W.: Attack on Han et al.'s ID-based Confirmer (Undeniable) Signature at ACM-EC 2003, Avalibale online http://eprint.iacr.org/2003/129
51. Zhang, F., Safavi-Naini, R., Susilo, W.: An Efficient Signature Scheme from Bilinear Pairings and Its Application. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004)