Universally Composable Undeniable Signature

Kaoru Kurosawa[1] and Jun Furukawa[2]
[1] Ibaraki University, Japan, kurosawa@mx.ibaraki.ac.jp
[2] NEC Corporation, Japan, j-furukawa@ay.jp.nec.com

Abstract. How to define the security of undeniable signature schemes is a challenging task. This paper presents two security definitions of undeniable signature schemes which are more useful or natural than the existing definition. It then proves their equivalence.
We first define the UC-security, where UC means universal composability. We next show that there exists a UC-secure undeniable signature scheme which does not satisfy the standard definition of security that has been believed to be adequate so far. More precisely, it does note satisfy the invisibility defined by [19]. We then show a more adequate definition of invisibility which captures a wider class of (naturally secure) undeniable signature schemes.
We finally prove that the UC-security against non-adaptive adversaries is equivalent to this definition of invisibility and the strong unforgeability in $\mathcal{F}_{ZK}$-hybrid model, where $\mathcal{F}_{ZK}$ is the ideal ZK functionality. Our result of equivalence implies that all the known proven secure undeniable signature schemes (including Chaum's scheme) are UC-secure if the confirmation/disavowal protocols are both UC zero-knowledge.

Keywords: Universal composability, undeniable signature scheme

1 Introduction

The concept of undeniable signature schemes was introduced by Chaum and van Antwerpen [15]. In an undeniable signature scheme, the signer issues an undeniable signature σ which is not publicly verifiable. She then proves the validity or invalidity of σ to the verifier in zero-knowledge (ZK) by running a confirmation protocol or disavowal protocol. Undeniable signature schemes have found various applications in cryptography such as in licensing software [15], electronic cash [16, 2, 34], electronic voting and auction. Then there have been a wide range of research covering a variety of different schemes for undeniable signatures over the past 15 years [12, 1, 14, 13, 24, 19, 23, 30, 9, 22, 21, 27, 3, 31, 32].

Recently, the security of Chaum's undeniable signature scheme is proved formally in the random oracle model under the decisional Diffie-Hellman (DDH) assumption by [33]. In the standard model, Laguillaumie and Vergnaud showed an undeniable signature scheme which is secure

under a decisional variant of the strong Diffie-Hellman (DH) assumption [28]. Kurosawa and Takagi showed an undeniable signature scheme which is secure under the strong RSA assumption and the decisional Nth residuosity assumption [26].

However, how to define the security of undeniable signature schemes is a challenging task. For example, it is not known if the security of these schemes is maintained under a general protocol composition. This concern is serious because undeniable signatures are often used as a building block in a more complicated protocol as shown above.

This paper presents two security definitions of undeniable signature schemes which are more useful or natural than the existing definition. It then proves their equivalence.

We first present an ideal functionality of undeniable signature schemes Σ in the universally composable (UC) framework [4, 5]. We next show that there exists a UC-secure undeniable signature scheme which does not satisfy the standard definition of security that has been believed to be adequate so far. More precisely, it does not satisfy the invisibility defined by [19]. The invisibility means that, for a message m, the receiver cannot tell if σ is a valid signature or a simulated signature. We then show a more adequate definition of invisibility which captures a wider class of (naturally secure) undeniable signature schemes.

We finally prove that the UC-security against non-adaptive adversaries is equivalent to this definition of invisibility and the strong unforgeability in $\mathcal{F}_{ZK}$-hybrid model where $\mathcal{F}_{ZK}$ is the ideal ZK functionality. For adaptive adversaries, we show that it is impossible to construct a UC-secure undeniable signature scheme even in the $\mathcal{F}_{ZK}$-hybrid model.

Our result of equivalence implies that all the known proven secure undeniable signature schemes (including Chaum's scheme) [33, 28, 26] are UC-secure against non-adaptive adversaries if the confirmation protocol and the disavowal protocol are UC zero-knowledge. Hence the security of these schemes is maintained under a general protocol composition against non-adaptive adversaries.

2 Preliminaries

2.1 Undeniable Signature Scheme

According to [19], an undeniable signature scheme is denoted by

$$\Sigma = (\mathsf{G}_{sign}, \mathsf{Sign}, \mathsf{Check}, \mathsf{Sim}, \pi_{con}, \pi_{dis}).$$

It consists of a key generation algorithm G_{sign}, a signing algorithm Sign, a validity check algorithm Check, a signature simulator Sim, a confirmation protocol π_{con} and a disavowal protocol π_{dis}.

The key generation algorithm G_{sign} is a PPT (probabilistic polynomial-time) algorithm which outputs (vk, sk), where vk is a verification key and sk is the signing key. [1] The message space $\mathcal{M}$ is specified by vk.

The signing algorithm Sign is a PPT algorithm which generates a signature σ on input a message $m \in \mathcal{M}$ and the signing key sk.

We say that (m, σ) is valid if σ is an output of $\mathsf{Sign}(sk, m)$ for some random string r. Otherwise, we say that (m, σ) is invalid. The validity check algorithm Check is a deterministic polynomial time algorithm such that

$$\mathsf{Check}((vk, m, \sigma), sk) = \begin{cases} 1 \; if \; (m, \sigma) = \mathsf{valid} \\ 0 \; if \; (m, \sigma) = \mathsf{invalid} \end{cases}$$

The signature simulator Sim is a PPT algorithm which outputs a simulated signature such that $\sigma' = \mathsf{Sim}(vk, m)$.

An undeniable signature scheme must satisfy unforgeability and invisibility. Invisibility means that for a message m, the receiver cannot tell if σ is a valid signature or a simulated signature.

This implies that the receiver cannot verify the validity of (m, σ) by himself. Instead, the cooperation of the signer is needed to verify the validity and invalidity of (m, σ) by running a confirmation protocol π_{con} and a disavowal protocol π_{dis} with the receiver respectively. π_{con} is a zero-knowledge interactive proof system (ZKIP) on a language $L_0 = \{(vk, m, \sigma) \mid (m, \sigma) \text{ is valid}\}$, and π_{dis} is a ZKIP on a language $L_1 = \{(vk, m, \sigma) \mid (m, \sigma) \text{ is invalid}\}$. Each ZKIP must satisfy completeness, soundness and zero-knowledgeness.

2.2 Security of Undeniable Signature

Unforgeability The unforgeability is defined as follows. Consider the following game between a challenger and an adversary A.

1. The challenger generates a key pair (vk, sk) randomly, and gives the verification key vk to A.
2. For $i = 1, 2, \ldots, q_s$ for some q_s, A queries a message m_i to the signing oracle adaptively and receives a signature σ_i.
3. Eventually, A outputs a forgery (m^*, σ^*).

[1] We assume that sk is uniquely determined by vk.

We allow the adversary A to query (m_j, σ_j) to the confirmation/disavowal oracle adaptively at step 2, where the confirmation/disavowal oracle responds as follows.

- If (m_j, σ_j) is a valid pair, then the oracle returns a bit $\mu = 1$ and proceeds with the execution of the confirmation protocol π_{con} with A.
- Otherwise, the oracle returns a bit $\mu = 0$ and executes the disavowal protocol π_{dis} with A accordingly.

We say that A succeeds in strong forgery if (m^*, σ^*) is valid and (m^*, σ^*) is not among the pairs (m_i, σ_i) generated during the signing oracle queries. [2]

Definition 1. *We say that Σ is strongly unforgeable if* $\Pr[A$ *succeeds in strong forgery* $]$ *is negligible for any PPT adversary A in the above game.*

Invisibility Damgård and Pedersen defined the invisibility by using the following game between a challenger and an adversary A [19].

1. The challenger generates a key pair (vk, sk) randomly, and gives the verification key vk to A.
2. A is permitted to issue a series of signing queries m_i to the signing oracle adaptively and receives a signature σ_i.
3. At some point, A chooses a message m^* and sends it to the challenger.
4. The challenger chooses a random bit b.
   If $b = 1$, then he computes a real signature $\sigma^* = \mathsf{Sign}(sk, m^*)$.
   Otherwise, he computes a fake signature $\sigma^* = \mathsf{Sim}(vk, m^*)$.
   He then returns σ^* to A.
5. A performs some signing queries again
6. At the end of this attack game, A outputs a guess b'.

We allow the adversary A to query (m_j, σ_j) to the confirmation/disavowal oracle adaptively at step 2 and at step 5.

However, A is not allowed to query the challenge (m^*, σ^*) to the confirmation/disavowal oracle at step 5. Also A is not allowed to query m^* to the signing oracle.

Definition 2. *We say that Σ is invisible if for any PPT adversary A,* $|\Pr[b = b'] - 1/2|$ *is negligible in the above game.*

[2] We say that A succeeds in weak forgery if (m^*, σ^*) is valid and m^* has never been queried to the signing oracle. Weak unforgeability and strong one are equivalent if the signing algorithm is deterministic, and there exists a unique signature for each message that is verified correctly.

2.3 Universal Composability

The security of a protocol $\pi = (P_1, \cdots, P_n)$ is maintained under a general protocol composition if π is secure in the universally composable (UC) security framework [4, 5]. See Appendix A.

3 UC Undeniable Signature

3.1 Ideal Functionality

Suppose that there exists a trusted third party (TTP) who has magical ink such that anything written by it is not visible. Only TTP can see it by using a special pair of glasses. Then the ideal functionality of undeniable signature schemes can be illustrated as follows.

1. A signer, Alice, first receives a registered number vk from TTP.
2. Upon signing request on a message m from Alice, TTP makes a signature σ on m (on behalf of vk) by using the magical ink.
3. Upon verification request on (m, σ, Bob) from Alice, TTP checks if σ is a correct signature (on behalf of vk) by using the special pair of glasses. Then it tells Bob if (m, σ) is valid or not.

We now present the ideal functionality $\mathcal{F}_{usig}$ of undeniable signature schemes in the UC framework.

Key Generation: 1. Upon receiving a value $(\texttt{KeyGen}, sid)$ from some party P, verify that $sid = (P, sid')$ for some sid'. If not, then ignore the request. Else, hand $(\texttt{KeyGen}, sid)$ to the adversary.
2. Upon receiving $(\texttt{Keys}, sid, vk, \mathsf{Sim})$ from the adversary, output $(\texttt{VerifyKey}, sid, vk, \mathsf{Sim})$ to P, where vk is a verification key and Sim is a PPT algorithm.

Signature Generation: Upon receiving a value $(\texttt{Sign}, sid, m)$ from P, verify that $sid = (P, sid')$ for some sid'. If not, then ignore the request. Else do:

1. If $(m, \sigma, 1)$ is recorded, then output $(\texttt{Signature}, sid, m, \sigma)$ to P. [3]
2. Else, if P is not corrupted, generate $\sigma = \mathsf{Sim}(vk, m)$ randomly such that no entry $(m, \sigma, 0)$ is recorded. Then output $(\texttt{Signature}, sid, m, \sigma)$ to P and the adversary.

[3] Ignore this step if the signing algorithm is probabilistic.

3. Else send $(\texttt{Sign}, sid, m)$ to the adversary.
   Upon receiving $(\texttt{Signature}, sid, m, \sigma)$ from the adversary, verify that no entry $(m, \sigma, vk, 0)$ is recorded. If it is, then output an error message to P and halt. Else, output $(\texttt{Signature}, sid, m, \sigma)$ to P, and record the entry $(m, \sigma, vk, 1)$.

Verification: Upon receiving a value $(\texttt{Verify}, sid, m, \sigma, V)$ from P, where V is a verifier, verify that $sid = (P, sid')$ for some sid'. If not, then ignore the request. Else do:

1. If $(m, \sigma, flag')$ is recorded, then set $flag = flag'$.
2. Else, if P is not corrupted, then set $flag = 0$ and record $(m, \sigma, 0)$. (This condition guarantees strong unforgeability: if the signer is not corrupted, and never signed m, then the verification fails.)
3. Else, hand $(\texttt{Verify}, sid, m, \sigma, V)$ to the adversary.
   Upon receiving $(\texttt{AdVerified}, sid, m, \sigma, \phi)$ from the adversary, let $flag = \phi$ and record (m, σ, ϕ).

Finally output $(\texttt{Verified}, sid, (m, \sigma), flag)$ to V and the adversary.

3.2 Remarks

The main differences between $\mathcal{F}_{usig}$ and $\mathcal{F}_{sig}$ are as follows, where $\mathcal{F}_{sig}$ is defined in Appendix A.1.

- At key generation, $\mathcal{F}_{sig}$ receives vk from the adversary, and hands it to P. On the other hand, $\mathcal{F}_{usig}$ receives (vk, Sim) from the adversary, and hands it to P.
- At signature generation, $\mathcal{F}_{sig}$ receives σ from the adversary, and hands it to P. On the other hand, $\mathcal{F}_{usig}$ computes $\sigma = \mathsf{Sim}(vk, m)$, and hands it to P. This is because σ must be invisible in undeniable signature schemes.
- The *signer* (P) issues $\texttt{Verify}$ command to $\mathcal{F}_{usig}$ while the *verifier* (V) issues it to $\mathcal{F}_{sig}$. This is because V should not be able to verify the validity of (m, σ) without the cooperation of P in undeniable signature schemes.

The adversary returns $(\texttt{Keys}, sid, vk, \mathsf{Sim})$ to $\mathcal{F}_{usig}$ at key generation. Hence Sim depends on vk. This means that we can write $\sigma = \mathsf{Sim}(m)$ instead of $\sigma = \mathsf{Sim}(vk, m)$ at signature generation.

4 Subtlety on Invisibility and New definition

4.1 Problem of Previous Definition

The standard definition of invisibility (Def. 2) was given by Damgård and Pedersen [19], where Sim is a part of Σ. However, we show that there

exists a UC-secure (and naturally secure) undeniable signature scheme which does not satisfy this definition of invisibility (Def. 2).

Let Σ be an undeniable signature scheme which satisfies the strong unforgeability and the invisibility defined by Def. 1 and Def. 2. Let Σ' be a strongly unforgeable (usual) signature scheme. Then consider an undeniable signature scheme Ω based on Σ and Σ' as follows.

- The public-key of Ω is (vk, vk'), where vk is a public-key of Σ, and vk' is a public-key of Σ'.
- The undeniable signature $\tilde{\sigma}$ on a message m is (σ, sk', σ'), where σ is an undeniable signature of Σ on m, sk' is a secret-key of Σ' and σ' is a (usual) signature of Σ' on m.

This undeniable signature scheme Ω does not satisfy the invisibility defined by Def. 2 because any PPT $\mathsf{Sim}()$ cannot compute sk'.

However, we can show that Ω is UC-secure. Intuitively, it is strongly unforgeable because Σ is strongly unforgeable. It is *naturally* invisible because σ is invisible, and everyone can compute σ' for any message by using sk' once he obtains sk' (for example, by known message attack). Indeed, our ideal process adversary S has only to return Sim which includes sk' at Key Generation.

4.2 New Definition of Invisibility

The above difference comes from the fact that Sim is independent of vk in the previous definition while it is not in the UC framework. Indeed, the adversary returns (vk, Sim) to $\mathcal{F}_{usig}$ in the UC framework.

We now show a new definition of invisibility. We delete Sim from Σ, and let Sim be a part of a public-key. That is, we define an undeniable signature scheme as

$$\Sigma = (\mathsf{G}_{sign}, \mathsf{Sign}, \mathsf{Check}, \pi_{con}, \pi_{dis})$$

such that

- the key generation algorithm G_{sign} outputs (vk, sk) and Sim. The signer makes (vk, Sim) public, and keeps sk secret.

The other parts of Σ remain the same. Accordingly, we need to modify step 1 of the attack game of invisibility shown in Sec.2.2 as follows.

1. The challenger generates (vk, sk) and Sim by running G_{sign}, and gives (vk, Sim) to A.

We then define invisibility as follows.

Definition 3. *We say that Σ is invisible if for any PPT adversary A, $|\Pr[b = b'] - 1/2|$ is negligible in the modified attack game.*

Then Ω is invisible under our new definition. More generally, it is easy to see that our new definition captures a wider class of (naturally secure) undeniable signature schemes.

4.3 New Definition of Unforgeability

We also need to modify step 1 of the attack game of unforgeability shown in Sec.2.2 as follows.

1. The challenger generates (vk, sk) and Sim by running G_{sign}, and gives (vk, Sim) to A.

We then define strong unforgeability as follows.

Definition 4. *We say that Σ is strongly unforgeable if $\Pr[A$ succeeds in strong forgery $]$ is negligible for any PPT adversary A in the modified attack game.*

4.4 Translation to Protocol

Under our new definition of Sec.4.2 and Sec.4.3, we show how to translate an undeniable signature scheme $\Sigma = (\mathsf{G}_{sign}, \mathsf{Sign}, \mathsf{Check}, \pi_{con}, \pi_{dis})$ into a protocol π_Σ in $\mathcal{F}_{ZK}$-hybrid model, where $\mathcal{F}_{ZK}$ is the ZK functionality on the binary relation Check.

1. When party P receives an input $(\texttt{KeyGen}, sid)$, it verifies that $sid = (P, sid')$ for some sid'. If not, it ignores the input. Else it generates (vk, sk) and Sim by running G_{sign}, and outputs $(\texttt{VerifyKey}, sid, vk, \mathsf{Sim})$.
2. When P receives an input $(\texttt{Sign}, sid, m)$ with $sid = (P, sid')$, it sets $\sigma = \mathsf{Sign}(sk, m)$ and outputs $(\texttt{Signature}, sid, m, \sigma)$.
3. When P receives an input $(\texttt{Verify}, sid, m, \sigma, V)$, do:
   (a) P sends $((vk, m, \sigma), sk)$ to $\mathcal{F}_{ZK}$.
   (b) $\mathcal{F}_{ZK}$ then sends $(\texttt{Verified}, sid, P, (vk, m, \sigma), f)$ to V and the adversary, where $f = \mathsf{Check}((vk, m, \sigma), sk)$.
   (c) Finally V outputs $(\texttt{Verified}, sid, (m, \sigma), f)$.

When a party is corrupted, it reveals its internal state, which includes all past signing and verification requests and answers, and for P also the state of the signing algorithm, including the signing key and the randomness used to sign past messages.

Definition 5. *We say that an undeniable signature scheme Σ is UC-secure if π_Σ securely realizes $\mathcal{F}_{usig}$.*

5 Equivalence

In this section, we prove that our UC-security notion of undeniable signature schemes is equivalent to the strong unforgeability (given by Def. 1) and our new definition of invisibility (see Sec.4.2).

Theorem 1. *Σ satisfies strong unforgeability and invisibility if Σ is UC-secure against non-adaptive adversaries in the $\mathcal{F}_{ZK}$-hybrid model.*

Proof. Assume that Σ does not satisfy strong unforgeability or invisibility. We show that π_Σ does not securely realize $\mathcal{F}_{usig}$.

This is done by constructing an environment $\mathcal{Z}$ and an adversary A such that for any ideal process adversary $\mathcal{S}$, $\mathcal{Z}$ can tell whether it is interacting with A and π_Σ, or with $\mathcal{S}$ in the ideal process for $\mathcal{F}_{usig}$. Our $\mathcal{Z}$ corrupts no parties and gives no inputs to A.

(I) Assume that Σ is not strongly unforgeable, i.e. there exists an adversary G which succeeds in strong forgery with nonnegligible probability ϵ. $\mathcal{Z}$ first sets $sid = (P, 0)$, and activates some party P with input $(\texttt{KeyGen}, sid)$. $\mathcal{Z}$ then obtains (vk, Sim) [4] and internally runs an instance of G. From now on, whenever G asks the sign oracle to sign a message m, $\mathcal{Z}$ activates P with input $(\texttt{Sign}, sid = (P, 0), m)$, and reports the output to G. Whenever G makes a confirmation/disavowal query (m, σ), $\mathcal{Z}$ activates P with input $(\texttt{Verify}, sid, m, \sigma, V)$ for some party V, and reports the output of V to G.

Finally G outputs (m^*, σ^*). Then $\mathcal{Z}$ activates P with input $(\texttt{Verify}, sid, m^*, \sigma^*, V)$ for some party V. Finally $\mathcal{Z}$ outputs 1 if and only if the output of V is valid[5] and (m^*, σ^*) is not among the pairs (m_i, σ_i) generated during the signing oracle queries.

Now suppose that $\mathcal{Z}$ interacts with π_Σ. It is easy to see that $\mathcal{Z}$ simulates the attack game perfectly for G. Further $\mathcal{Z}$ outputs 1 if and only if G succeeds in forgery. Therefore, $\mathcal{Z}$ outputs 1 with probability ϵ.

[4] In the ideal world, see step 2 of key generation. In the real world, see step 1 of π_Σ.
[5] where valid means that the output of V is $(\texttt{Verified}, sid, P, (vk, m, \sigma), 1)$.

On the other hand, if $\mathcal{Z}$ interacts with $\mathcal{S}$ in the ideal process for $\mathcal{F}_{usig}$, then $\mathcal{Z}$ never outputs 1. This means that π_Σ does not securely realize $\mathcal{F}_{usig}$.

(II) Assume that Σ is not invisible. Then there exists an adversary A which breaks the invisibility with nonnegligible probability ϵ.

$\mathcal{Z}$ proceeds as above, i.e. $\mathcal{Z}$ first sets $sid = (P, 0)$, and activates some party P with input $(\mathtt{KeyGen}, sid)$. $\mathcal{Z}$ then obtains (vk, Sim) and internally runs an instance of A. From now on, whenever A asks the sign oracle to sign a message m, $\mathcal{Z}$ activates P with input $(\mathtt{Sign}, sid = (P, 0), m)$, and reports the output to A. Whenever A makes a confirmation/disavowal query (m, σ), $\mathcal{Z}$ activates P with input $(\mathtt{Verify}, sid, m, \sigma, V)$ for some party V, and reports the output of V to A.

At some point, A chooses a message m^* (which has never been queried if the signing algorithm is deterministic), and sends it to $\mathcal{Z}$. Then $\mathcal{Z}$ chooses a random bit b and does the following:

- If $b = 1$, $\mathcal{Z}$ activates P with input $(\mathtt{Sign}, sid = (P, 0), m)$, and reports the output of P to A.
- Otherwise, $\mathcal{Z}$ computes $\sigma' = \mathsf{Sim}(vk, m)$ randomly and reports σ' to A.

A finally outputs a bit b'. $\mathcal{Z}$ then outputs 1 if and only if $b = b'$.

Suppose that $\mathcal{Z}$ interacts with π_Σ. Then A sees exactly the attack game on invisibility on Σ. Therefore,

$$|\Pr(\mathcal{Z} \text{ outputs } 1) - 1/2| \geq \epsilon.$$

On the other hand, if $\mathcal{Z}$ interacts with $\mathcal{S}$ in the ideal process for $\mathcal{F}_{usig}$. then it is easy to see that

$$\Pr(\mathcal{Z} \text{ outputs } 1) - 1/2 = 0$$

because both $\mathcal{F}_{usig}$ and $\mathcal{Z}$ compute $\sigma' = \mathsf{Sim}(vk, m)$ randomly. This means that π_Σ does not securely realize $\mathcal{F}_{usig}$. This completes our proof. □

Corollary 1. *Σ satisfies weak unforgeability and invisibility if Σ is UC-secure against non-adaptive adversaries.*

Theorem 2. *Σ is UC-secure against non-adaptive adversaries if Σ satisfies strong unforgeability and invisibility in the $\mathcal{F}_{ZK}$-hybrid model.*

5.1 Proof of Theorem 2

Assume that π_Σ does not securely realize $\mathcal{F}_{usig}$ against non-adaptive adversaries. We show that Σ does not satisfy strong unforgeability or invisibility. Assume that Σ is invisible (otherwise the theorem is proven). Then there exists a PPT algorithm Sim which satisfies the definition of the invisibility. Our goal is to construct a forger G.

Using the equivalent notion of security against the (non-adaptive) dummy adversary D, [6] we have that for any ideal process adversary $\mathcal{S}$, there exists an environment $\mathcal{Z}$ that can tell whether it is interacting with $\mathcal{F}_{usig}$ and $\mathcal{S}$, or with π_Σ and the non-adaptive dummy adversary D. (Remember that non-adaptive adversaries corrupt parties at the beginning of executions only.)

We consider a particular $\mathcal{S}$ as shown below. For this particular $\mathcal{S}$, there exists an environment $\mathcal{Z}_S$ that can distinguish the real world and the ideal world. We will use this $\mathcal{Z}_S$ to construct a forger G on Σ.

First our particular $\mathcal{S}$ behaves as follows.

- Suppose that there are no party corruption instructions by $\mathcal{Z}$. In this case, $\mathcal{S}$ provides $\mathcal{F}_{usig}$ with vk and Sim at key generation. $\mathcal{S}$ outputs nothing other than this.

- Suppose that $\mathcal{Z}$ instructs $\mathcal{S}$ to corrupt P at the beginning. In this case, $\mathcal{F}_{usig}$ forwards all commands of $\mathcal{Z}$ (to P) to $\mathcal{S}$. Then $\mathcal{S}$ behaves in the same way as the real signer of π_Σ does. That is:

1. At key generation, $\mathcal{S}$ generates (vk, sk) randomly and returns vk and Sim to $\mathcal{F}_{usig}$.
2. At signature generation, $\mathcal{S}$ computes $\sigma = \mathsf{Sign}(sk, m)$ and returns σ to $\mathcal{F}_{usig}$.
3. At signature verification, $\mathcal{S}$ computes $\phi = \mathsf{Check}((vk, m, \sigma), sk))$ and returns $(\texttt{AdVerified}, sid, m, \sigma, \phi)$ to $\mathcal{F}_{usig}$.

Lemma 1. *$\mathcal{Z}_S$ does not corrupt P with nonnegligible probability.*

Proof. If $\mathcal{Z}_S$ always corrupts P (at the beginning), then such $\mathcal{Z}_S$ cannot distinguish the real world and the ideal world because our $\mathcal{S}$ behaves

[6] The dummy adversary D only delivers to parties messages generated by the environment $\mathcal{Z}$, and delivers to $\mathcal{Z}$ all messages generated by the parties. Instead of quantifying over all possible adversary A, it suffices to require that the ideal protocol adversary $\mathcal{S}$ be able to simulate, for any environment $\mathcal{Z}$, the behavior of the dummy adversary D. [6]

in the same way as the real signer. Hence $\mathcal{Z}_S$ does not corrupt P with nonnegligible probability.

□

Fig. 1. Forger G

1. G is given (vk, Sim) as an input. G then runs $\mathcal{Z}_S$.
2. If $\mathcal{Z}_S$ corrupts some party P at the beginning, then G outputs failure.
   If $\mathcal{Z}_S$ activates P with input $(\mathtt{KeyGen}, sid)$ with $sid = (P, sid')$ for some sid', then G returns (vk, Sim) to $\mathcal{Z}_S$.
3. When $\mathcal{Z}_S$ activates P with input $(\mathtt{Sign}, sid, m)$, then G asks its signing oracle for a signature σ on m, and returns σ to $\mathcal{Z}_S$.
4. When $\mathcal{Z}_S$ activates P with input $(\mathtt{Verify}, sid, m, \sigma, V)$ for some party V, then G queries (m, σ) to its confirmation/disavowal oracle, and returns the answer to $\mathcal{Z}_S$ through V.
5. If the answer is valid, and (m, σ) is not a pair generated at step 3, then G outputs (m, σ) as a strong forgery and stops.

Next let **FORGE** denote the event that $\mathcal{Z}_S$ activates P with input $(\mathtt{Verify}, sid, m, \sigma, V)$ such that (m, σ) is a strong forgery.

Lemma 2. *Suppose that Σ satisfies the invisibility. Also suppose that $\mathcal{Z}_S$ does not corrupt P, and can distinguish the real world from the ideal world. Then* **FORGE** *happens in the real world with nonnegligible probability.*

Now we present our forger G in Fig.1. Suppose that $\mathcal{Z}_S$ does not corrupt P. Then G simulates the real world for $\mathcal{Z}_S$ until step 5. Thereofore the view of $\mathcal{Z}_S$ of Fig.1 is the same as the view of $\mathcal{Z}_S$ in the real world until step 5. Hence from Lemma 1 and Lemma 2, it is clear that G succeeds in strong forgery with nonnegligible probability if Σ is not UC-secure and satisfies the invisibility. This completes the proof of Theorem 2.

(Proof of Lemma 2)

It is clear that **FORGE** never happens in the ideal world. We prove that $\mathcal{Z}_S$ cannot distinguish the real world and the ideal world if **FORGE** never happens in the real world.

Suppose that **FORGE** never happens in the real world. Then the view of $\mathcal{Z}_S$ in the real world is identical to the view of $\mathcal{Z}_S$ shown in Fig.2.

We consider a series of games on $\mathcal{Z}_S$ as follows. $\mathbf{Game}_0$ is the same as Fig.2 except for that σ_i are all simulated signatures. Assume that $\mathcal{Z}_S$

Fig. 2. FORGE never happens

1. When party P receives an input $(\texttt{KeyGen}, sid)$,
   it verifies that $sid = (P, sid')$ for some sid'. If not, it ignores the input.
   Else it generates (vk, sk) and Sim by running G_{sign},
   and outputs $(\texttt{VerifyKey}, sid, vk, \mathsf{Sim})$.
2. When P receives an input $(\texttt{Sign}, sid, m)$ with $sid = (P, sid')$,
   it sets $\sigma = \mathsf{Sign}(sk, m)$ and outputs $(\texttt{Signature}, sid, m, \sigma)$.
   P records (m, σ).
3. When P receives an input $(\texttt{Verify}, sid, m, \sigma, V)$, do:
   If (m, σ) is recorded, then P outputs valid. Otherwise P outputs invalid.

activates P with input $(\texttt{Sign}, sid, m_i)$ and the signing oracle returns σ_i for $i = 1, \cdots, q_s$. For $j = 1, \cdots, q_s$, $\mathbf{Game}_j$ is the same as Fig.2 except for that σ_i is a real signature for $i = 1, \cdots, j$, and σ_i is a simulated signature for $i = j + 1, \cdots, q_s$. Note that $\mathbf{Game}_{q_s}$ is the same as Fig.2, where σ_i are all real signatures.

From a view point of $\mathcal{Z}_S$, it is clear that $\mathbf{Game}_0$ is the ideal world and $\mathbf{Game}_{q_s}$ is the real world. Therefore from our assumption, $\mathcal{Z}_S$ can distinguish $\mathbf{Game}_0$ and $\mathbf{Game}_{q_s}$. Then it is easy to show that there exists J such that $\mathcal{Z}_S$ can distinguish $\mathbf{Game}_{J-1}$ and $\mathbf{Game}_J$.

Now we construct an adversary A who can break the invisibility by using $\mathcal{Z}_S$ as follows. A engages in the attack game on the invisibility. First, A is given (vk, Sim) by the challenger. It then runs $\mathcal{Z}_S$. When $\mathcal{Z}_S$ invokes some uncorrupted P, A returns (vk, Sim) to $\mathcal{Z}_S$.

Suppose that $\mathcal{Z}_S$ activates P with input $(\texttt{Sign}, sid, m_i)$.

- If $i < J$, then A queries m_i to his own signing oracle and receives a real signature σ_i. A records (m_i, σ_i).
- If $i > J$, then A computes a simulated signature $\sigma_i = \mathsf{Sim}(vk, m_i)$.
- If $i = J$, then A sends m_J to the challenger as a challenge message, and receives σ_J from the challenger.

A then returns the above σ_i to $\mathcal{Z}_S$.

Suppose that $\mathcal{Z}_S$ activates P with input $(\texttt{Verify}, sid, m, \sigma, V)$ for some party V. If (m, σ) is recorded, then A returns valid. Otherwise, A returns invalid. (Remember that **FORGE** never happens.)

Let b' be the final output of $\mathcal{Z}_S$. A outputs this b'.

It is clear that the view of $\mathcal{Z}_S$ is exactly the same as that of $\mathbf{Game}_{J-1}$ and $\mathbf{Game}_J$ according to the challenge bit b of the challenger. Therefore from the definition of J, $|\Pr(b' = b) - 1/2|$ is nonnegligible. This means

that A wins the attack game on the invisibility. However, this is a contradiciton.

This complets the proof of Lemma 2.

6 Application

In this section, we show that, Chaum's undeniable signature scheme is UC-secure against non-adaptive adversaries in the random oracle model if it uses a confirmation protocol and a disavowal protocol which are UC zero-knowledge.

Let G be an Abelian group of prime order q, and let g be a generator of G. We say that (g, g^x, g^r, g^z) is a DH-tuple if $z = xr \bmod q$. The CDH assumption claims that it is hard to compute g^{xr} from (g, g^x, g^r). The DDH assumption claims that it is hard to decide if (g, g^x, g^r, g^z) is a DH-tuple.

Then Chaum's undeniable signature scheme is described as follows.

- **Key Generation.** On input the security parameter 1^k, choose $x \in Z_q$ randomly and compute $y = g^x$. Choose a cryptographic hash function $H : \{0,1\}^* \to G$. Set the public key as (g, y, H) and the secret key as x.
- **Signing.** On input the public key (g, y, H), the secret key x and a message $m \in \{0,1\}^*$, the algorithm returns the signature as $\sigma = H(m)^x$.
- **Confirmation Protocol.** Given a message-signature pair (m, σ), the signer proves that $(g, y, H(m), \sigma)$ is a DH-tuple in zero-knowledge.
- **Disavowal Protocol.** Given a message-signature pair (m, σ), the signer proves that $(g, y, H(m), \sigma)$ is not a DH-tuple in zero-knowledge.

It is known that Chaum's undeniable signature scheme is (strongly) unforgeable under CDH assumption, and invisible under DDH assumption in the random oracle model [33]. Hence from Theorem 2, we have the following corollary.

Corollary 2. *Chaum's undeniable signature scheme is UC-secure against non-adaptive adversaries under the DDH assumption in the random oracle model if it uses a confirmation protocol and a disavowal protocol which are UC zero-knowledge.*

7 Impossibility Result

In this section, we show that it is impossible to construct an undeniable signature scheme which is UC-secure against adaptive adversaries.

Theorem 3. *There exists no undeniable signature scheme Σ which is UC-secure against adaptive adversaries even in the $\mathcal{F}_{ZK}$-hybrid model.*

A proof is given in Appendix B.

References

1. J. Boyar, D. Chaum, I. Damgård and T. Pedersen. Convertible undeniable signatures. *CRYPTO '90*, LNCS 537, pp.189–208, Springer-Verlag, 1990.
2. C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. *ASIACRYPT '98*, LNCS 1514, pp.271–285, Springer-Verlag, 1998.
3. I. Biehl, S. Paulus and T. Takagi. Efficient undeniable signature schemes based on ideal arithmetic in quadratic orders. *Designs, Codes and Cryptography*, Vol. 31, Issue 2, pp.99–123, 2004
4. Ran Canetti, Universally Composable Security: A New Paradigm for Cryptographic Protocols, Revision 1 of ECCC Report TR01-016 (2001)
5. Ran Canetti, Universally Composable Signatures, Certification and Authentication, IACR ePrint 2003/239
6. Ran Canetti, Universally Composable Security: A New Paradigm for Cryptographic Protocols, IACR ePrint 2000/067 (2005)
7. R.Canetti, M.Fischlin: Universally Composable Commitments. CRYPTO 2001: 19-40
8. R.Canetti, Y.Lindell, R.Ostrovsky, A.Sahai: Universally composable two-party and multi-party secure computation. STOC 2002: 494-503
9. J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. *EUROCRYPT '00*, LNCS 1870, pp.243–258, Springer-Verlag, 2000.
10. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. *CRYPTO '03*, LNCS 2729, pp.126–144, Springer-Verlag, 2003.
11. D. Catalano, P. Nguyen, J. Stern. The hardness of Hensel lifting: The Case of RSA and Discrete Logarithm. *ASIACRYPT '02*, LNCS 2501, pp.299-310, Springer-Verlag, 2002.
12. D. Chaum. Zero-knowledge undeniable signatures. *EUROCRYPT '90*, LNCS 473, pp.458–464, Springer-Verlag, 1990.
13. D. Chaum. Designated confirmer signatures. *EUROCRYPT '94*, LNCS 950, pp.86–91, Springer-Verlag, 1995.
14. D. Chaum, E. van Heijst and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. *CRYPTO '91*, LNCS 576, pp.470–484, Springer-Verlag, 1991.
15. D. Chaum and H. van Antwerpen. Undeniable signatures. *CRYPTO '89*, LNCS 435, pp.212–216, Springer-Verlag, 1989.
16. T. Chaum and T. P. Pedersen. Wallet databases with observers. *CRYPTO '92*, LNCS 740, pp.89–105, Springer-Verlag, 1993.

17. J. -S. Coron. On the exact security of full domain hash. *CRYPTO '00*, LNCS 1880, pp.229–235, Springer-Verlag, 2000.
18. R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, vol.3, no.3, pp.161—185, 2000.
19. I. Damgård and T. Pedersen. New convertible undeniable signature schemes. *EUROCRYPT '96*, LNCS 1070, pp.372–386, Springer-Verlag, 1996.
20. L. Guillou and J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory. *EUROCRYPT '88*, LNCS 330, pp.123–128, Springer-Verlag, 1989.
21. S. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. *Topics in Cryptology — CT-RSA '03*, LNCS 2612, pp.80–97, Springer Verlag, 2003.
22. S. Galbraith, W. Mao and K. G. Paterson. RSA-based undeniable signatures for general moduli. *CT-RSA '02*, LNCS 2271, pp. 200–217, Springer Verlag, 2002.
23. R. Gennaro, T. Rabin and H. Krawczyk. RSA-based undeniable signatures. *Journal of Cryptology*, 13(4), pp.397–416, 2000.
24. M. Jakobsson, K. Sako and R. Impagliazzo. Designated verifier proofs and their applications. *EUROCRYPT '96*, LNCS 1070, pp.143–154, Springer-Verlag, 1996.
25. K. Kurosawa and S. Heng: Relations among security notions for undeniable signature schemes. accepted by SCN 2006.
26. K.Kurosawa, T.Takagi: New Approach for Selectively Convertible Undeniable Signature Schemes. ASIACRYPT 2006: 428-443
27. B. Libert and J.-J Quisquater. Identity based undeniable signatures. *CT-RSA '04*, LNCS 2964, pp.112–125, Springer-Verlag, 2004.
28. F. Laguillaumie and D. Vergnaud: Short undeniable signatures without random oracles: The Missing Link. *INDOCRYPT '05*, LNCS 3797, pp.283–296, Springer-Verlag, 2005.
29. M. Michels, H. Petersen and P. Hoster. Breaking and repairing a convertible undeniable signature scheme. In *3rd ACM CCCS*, pp.148–152, 1996.
30. M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. *SAC '97*, pp.231–244, Springer-Verlag, 1997.
31. J. Monnerat and S. Vaudenay. Undeniable signatures based on characters: how to sign with one bit. *PKC '04*, LNCS 2947, pp.361–396, Springer-Verlag, 2004.
32. J. Monnerat and S. Vaudenay. Generic homomorphic undeniable signatures. *ASIACRYPT '04*, LNCS 3329, pp.354–371, Springer-Verlag, 2004.
33. W. Ogata, K. Kurosawa and S. Heng. The security of the FDH variant of Chaum's undeniable signature scheme. *IEEE Transactions on Information Theory*, 52(5), pp.2006–2017, 2006.
34. D. Pointcheval. Self-scrambling anonymizers. *FC '00*, LNCS 1962, pp.259–275, Springer-Verlag, 2000.

A Universal Composability

In this framework, there exists an environment $\mathcal{Z}$ which generates the input to all parties, reads all outputs, and in addition interacts with an adversary A in an arbitrary way throughout the computation.

A protocol π is said to securely realizes a given functionality $\mathcal{F}$ if for any adversary A, there exists an ideal-process adversary $\mathcal{S}$ such that no environment $\mathcal{Z}$ can tell whether it is interacting with A and parties running the protocol, or with $\mathcal{S}$ and parties that interact with $\mathcal{F}$ in the ideal process.

The following universal composition theorem is proven in [4, 5]. Consider a protocol π that operates in a hybrid model of computation where parties can communicate as usual, and in addition have ideal access to (an un unbounded number of copies of) some ideal functionality $\mathcal{F}$. Let ρ be a protocol that securely realizes as sketched above, and let π^ρ be the composed protocol. That is, π^ρ is identical to π with the exception that each interaction with some copy of $\mathcal{F}$ is replaced with a call to (or an invocation of) an appropriate instance of ρ. Similarly, ρ-outputs are treated as values provided by the appropriate copy of $\mathcal{F}$. Then π and π^ρ have essentially the same input/output behavior. In particular, if π securely realizes some ideal functionality $\mathcal{G}$ given ideal access to $\mathcal{F}$, then π^ρ securely realizes $\mathcal{G}$ from scratch.

For more details, see [4, 5].

A.1 UC Signature

We show a definition of the signature functionality $\mathcal{F}_{sig}$ given in [5]. It proceeds as follows.

Key Generation: Upon receiving a value $(\texttt{KeyGen}, sid)$ from some party P, verify that $sid = (P, sid')$ for some sid'. If not, then ignore the request. Else, hand $(\texttt{KeyGen}, sid)$ to the adversary.

Upon receiving $(\texttt{Verification Key}, sid, vk)$ from the adversary, output $(\texttt{Verification key}, sid, vk)$ to P, and the record the pair (P, vk).

Signature Generation: Upon receiving a value $(\texttt{Sign}, sid, m)$ from P, verify that $sid = (P, sid')$ for some sid'. If not, then ignore the request. Else send $(\texttt{Sign}, sid, m)$ to the adversary.

Upon receiving $(\texttt{Signature}, sid, m, \sigma)$ from the adversary, verify that no entry $(m, \sigma, vk, 0)$ is recorded. If it is, then output an error message to P and halt. Else, output $(\texttt{Signature}, sid, m, \sigma)$ to P, and record the entry $(m, \sigma, vk, 1)$.

Signature Verification: Upon receiving a value $(\texttt{Verify}, sid, m, \sigma, vk')$ from some party V, hand $(\texttt{Verify}, sid, m, \sigma, vk')$ to the adversary. Upon receiving $(\texttt{Verified}, sid, m, \sigma, \phi)$ from the adversary, do:

1. If $vk' = vk$ and the entry $(m, \sigma, vk, 1)$ is recorded, then set $f = 1$.

2. Else, if $vk' = vk$, the signer is not corrupted, and no entry $(m, \sigma', vk, 1)$ for any σ' is recorded, then set $f = 0$ and record the entry $(m, \sigma, vk, 0)$.
3. Else, if there is an entry (m, σ, vk', f') recorded, then let $f = f'$.
4. Else, let $f = \phi$ and record the entry (m, σ, vk', ϕ).

Output $(\texttt{Verified}, sid, m, \sigma, f)$ to V.

It is known that a signature scheme securely realizes $\mathcal{F}_{sig}$ if and only if it is existentially unforgeable against chosen message attack [5].

A.2 UC Zero-Knowledge

We use a definition of the zero-knowledge (ZK) functionality $\mathcal{F}_{ZK}$ given in [4]. It proceeds as follows, running with parties $P_1, \cdots, P_n$ and an adversary $\mathcal{S}$, given a binary relation R.

1. Upon receiving of a value $(\texttt{prover}, sid, P_i, P_j, x, w)$ from some party P_i, send $(\texttt{Verified}, sid, P_i, x, R(x, w))$ to P_j and $\mathcal{S}$, and halt.

It is known that $\mathcal{F}_{ZK}$ cannot be realized in the UC framework by plain protocols [4]. On the other hand, in the common random string model, interactive ZK protocols for any NP language were constructed in [7] and non-interactive ones were constructed in [8].

B Proof of Theorem 3

Proof. Suppose that there exists an undeniable signature scheme Σ such that π_Σ securely realizes $\mathcal{F}_{usig}$ against adaptive adversaries. Then it securely realizes $\mathcal{F}_{usig}$ against non-adaptive adversaries also. Hence Σ is strongly unforgeable from Theorem 1. Now we consider an environment $\mathcal{Z}$ and an adaptive adversary A as follows.

1. $\mathcal{Z}$ activates some party P with input $(\texttt{KeyGen}, sid)$ with $sid = (P, sid')$ for some sid', and receives (vk, Sim) from P.
2. $\mathcal{Z}$ chooses $m_1 \in \mathcal{M}$ [7] randomly, and activates P with input $(\texttt{Sign}, sid, m_1)$. $\mathcal{Z}$ then receives a signature σ_1 on m_1 from P.
3. A corrupts P and obtains the signing key sk. A then reports sk to $\mathcal{Z}$.
4. $\mathcal{Z}$ lets $b = 1$ if $\mathsf{Check}((vk, m_1, \sigma_1), sk) = 1$. Otherwise he lets $b = 0$. Finally $\mathcal{Z}$ outputs b and stops.

[7] $\mathcal{M}$ is the message space specified by vk.

First suppose that $\mathcal{Z}$ is interacting A and parties running π_{Σ}. Then it is clear that

$$\Pr(\mathcal{Z} \text{ outputs } 1) = \Pr[\mathsf{Check}((vk, m_1, \sigma_1), sk) = 1] = 1.$$

Next suppose that $\mathcal{Z}$ is interacting with an ideal process adversary $\mathcal{S}$ and dummy parties that interacts with $\mathcal{F}$. In this case, $\sigma_1 = \mathsf{Sim}(vk, m_1)$ for some PPT algorithm Sim. Suppose that $\Pr[\mathsf{Check}((vk, m_1, \sigma_1), sk) = 1]$ is nonnegligible. Then it is easy to see that Σ is strongly forgeable. Indeed, a forger chooses $m_1 \in \mathcal{M}$ randomly and computes $\sigma_1 = \mathsf{Sim}(vk, m_1)$. Then (m_1, σ_1) is a valid pair with nonnegligible probability. Therefore, $\Pr(\mathcal{Z} \text{ outputs } 1) = 1$ is negligible.

This means that $\mathcal{Z}$ can distinguish two worlds. This is against that π_{Σ} securely realizes $\mathcal{F}_{usig}$ against adaptive adversaries. Therefore, there exists no undeniable signature scheme Σ which is UC-secure against adaptive adversaries. □