New Approach for Selectively Convertible Undeniable Signature Schemes

Kaoru Kurosawa[1] and Tsuyoshi Takagi[2]

[1] Ibaraki University, Japan, kurosawa@mx.ibaraki.ac.jp
[2] Future University-Hakodate, Japan, takagi@fun.ac.jp

Abstract. In this paper, we propose a new approach for constructing selectively convertible undeniable signature schemes, and present two efficient schemes based on RSA. Our approach allows a more *direct* selective conversion than the previous schemes, and the security can be proved formally. Further, our disavowal protocols do not require parallelization techniques to reach a significant soundness probability. Also, our second scheme is the first selectively convertible scheme which is provably secure without random oracles.

Keywords. undeniable signature, selective conversion, RSA

1 Introduction

1.1 Background

The concept of undeniable signature **(US)** schemes was introduced by Chaum and van Antwerpen [10]. In an US scheme, the signer issues an undeniable signature τ which is not publicly verifiable. She then proves the validity or invalidity of τ in zero-knowledge by running a confirmation protocol or disavowal protocol with the receiver. US schemes have found various applications in cryptography such as in licensing software [10], electronic cash [11, 2, 31], electronic voting and auctions. Then there have been a wide range of research covering a variety of different schemes for undeniable signatures over the past 15 years [7, 1, 9, 8, 19, 14, 18, 25, 4, 17, 16, 22, 3, 26, 27].

Recently, the security of Chaum's US scheme was proved formally in the random oracle model by [28]. Laguillaumie and Vergnaud showed an US scheme which is secure in the standard model under the strong Diffie-Hellman (DH) assumption [23]. The relations among the security notions for US schemes was given by [21].

The notion of *convertible* US schemes was introduced by Boyar et al. [1]. A *selectively* convertible US scheme allows the signer to convert an undeniable signature τ into a regular signature by releasing a piece of information α at a later time. *All* conversion means that the signer can convert all undeniable signatures into regular ones. They showed that if there exists a digital signature **(DS)** scheme, then there exists a convertible US scheme. However, this construction is not practical.

Damgård and Pedersen showed two selectively convertible US scheme schemes based on ElGamal signature scheme [14]. In their schemes, a part of the ElGamal signature is encrypted by Rabin encryption scheme or by ElGamal encryption scheme. However, invisibility is not proved in these schemes [1], where the invisibility means that we cannot decide if (m, τ) is a valid (message, undeniable signature) pair. Note that the invisibility is an essential property required for US schemes from the definition.

Gennaro-Krawczyk-Rabin proposed an RSA-based US scheme which allows all conversion efficiently [18]. [2] They also showed a method of selective conversion such that the signer releases a non-interactive proof which shows that (m, τ) is a valid (message, undeniable signature) pair.

1.2 Our Contribution

In this paper, we propose a new approach for constructing selectively convertible undeniable signature schemes, and present two efficient schemes based on RSA. Our approach allows a more *direct* selective conversion than the previous schemes, and the security can be proved formally. Further, our disavowal protocols do not require parallelization techniques to reach a significant soundness probability. Also, our second scheme is the first selectively convertible US scheme whose security can be proved without random oracles.

A selectively convertible US scheme has two modes, the US signature issueing mode and the selective conversion mode. In our approach, we consider a DS signature issueing mode as well which is described as follows: For a message m,

- The signer issues an undeniable signature τ in the US mode.
- In the DS mode, the signer issues σ as a regular signature on m.
- In the selective conversion mode, the signer releases σ (which is the same as above) to convert the already issued undeniable signature τ into a regular signature. By using σ, the validity of (m, τ) is made publicly verifiable.

We first formalize such US schemes as two-sided undeniable/signature schemes ("two-sided scheme" for short). In the security model, we consider adversaries who have access to both the DS-sign oracle and the US-sign oracle. Adversaries then try to forge a digital signature σ (DS-forgery) or an undeniable signature τ (US-forgery). See Figure 1. Both types of forgery must be impossible, and invisibility must be satisfied.

We next show an efficient two-sided scheme based on RSA signature scheme and Paillier's encryption scheme [29]. In this scheme, the public-key is an RSA modulus $N(= pq)$.

[1] In Sec.5.1 and Sec.5.2 of [14], the authors wrote only that "We therefore conjecture that ..." on the invisibility of their schemes.

[2] GRK US scheme assumes that there exits an encoding method of messages so that the RSA-based DS scheme is unforgeable. However, no such encoding method is known in the standard model. Hence GRK US scheme is secure in the random oracle model only currently.

$$\begin{array}{c} \text{DS forgery} \\ \uparrow \\ \text{DS-sign oracle} \Longleftrightarrow \text{Adversary} \Longleftrightarrow \text{US-sign oracle} \\ \downarrow \\ \text{US forgery} \end{array}$$

Fig. 1. Adversary in Two-sided scheme

- Our DS mode is the same as the RSA signature scheme with $e = N$. That is, the signer issues a digital signature $\sigma \in Z_N^*$ on a message m such that

  $$\sigma^N = H(m) \bmod N,$$

  where H is a hash function.
- Now replace mod$N$ with modN^2 in the above equation. Then we obtain that

  $$\sigma^N = H(m) + \tau N \bmod N^2 \tag{1}$$

  for some $\tau \in Z_N$. We consider that this τ is an undeniable signature on m. That is, in the US mode, the signer issues the above τ as an undeniable signature.
- In the selective conversion mode, the signer releases σ (which is the RSA signature on m) to convert the already issued τ into a regular signature. The validity of (m, τ) is publicly verified by checking eq.(1).

  This piece of information σ released for selective conversion is smaller than that of GRK US scheme [18], where the latter is based on the Fiat-Shamir heuristic. [3]

Not only the above technique is new, but also our confirmation and disavowal protocols are based on a novel approach. In particular, our (zero-knowledge) disavowal protocol does not require parallelization techniques to reach a significant soundness probability. In the previous US schemes, only confirmation protocols are known which do not require parallelization techniques.

We then prove the security of our scheme in the random oracle model. Roughly speaking, our scheme relies on RSA assumption and the Nth residuosity assumption. [4]

Finally, we show the first selectively convertible US scheme which is provably secure in the standard model. It is a two-sided scheme, and it is obtained by applying our technique to Cramer-Shoup DS scheme [13] which is known to be secure in the standard model.

[3] Since our scheme does not use the Fiat-Shamir heuristic, it uses one random oracle H while GRK scheme must use two random oracles (see footnote 2).

[4] On the other hand, GRK US scheme [18] relies on RSA assumption and DDH assumption over Z_N^*, where the security model does not consider DS-sign oracle nor DS forgery.

Remark 1. In GRK US scheme [18], $N = pq$, where p and q must be safe primes. Galbraith et al. showed a method which can eliminate this restriction [17]. Our schemes are totally different from [18, 17], and p and q can be any primes.

2 Model and Definitions

For an algorithm A and its input x, we write $y \leftarrow A(x)$ if y is an output of $A(x)$.

2.1 Syntax

A two-sided scheme consists of six polynomial time algorithms ($\mathsf{Key}, \mathsf{DSign}, \mathsf{DVerify}, \mathsf{USign}, \mathsf{Convert}, \mathsf{UVerify}$), and two protocols, a confirmation protocol $\mathsf{Confirm}$ and a disavowal protocol $\mathsf{Disavow}$.

Key is a probabilistic algorithm which outputs a public-key pk and a secret-key sk on input 1^ℓ, where ℓ is a security parameter. The public-key pk specifies the message space $\mathcal{M}$, the space of digital signatures $\mathcal{D}$, and the space of undeniable signatures $\mathcal{U}$.

DSign is a (either probabilistic or deterministic) algorithm which outputs a digital signature σ on input (sk, m), where m is a message. We say that (m, σ) is a valid D-pair if there exists a random tape such that the algorithm $\mathsf{DSign}(sk, m)$ outputs σ.

$\mathsf{DVerify}$ is an algorithm which, on input (pk, m, σ), outputs *accept* if (m, σ) is a valid D-pair, and *reject* otherwise.

USign is a (probabilistic) algorithm which outputs an undeniable signature τ on input (sk, m), where m is a message. We say that (m, τ) is a valid U-pair if there exists a random tape such that the algorithm $\mathsf{USign}(sk, m)$ outputs τ.

$\mathsf{Convert}$ is an algorithm which outputs a digital signature σ for a valid U-pair (m, τ). More precisely, on input (sk, m, τ), it outputs *some* $\sigma \leftarrow \mathsf{DSign}(sk, m)$ if (m, τ) is a valid U-pair, and $\perp$ otherwise. Then by using $\mathsf{UVerify}$ shown below, the validity of (m, τ) is made publicly verifiable.
Note that the above σ is not necessarily a random output of $\mathsf{DSign}(sk, m)$. It must be related to τ so that the validity of (m, τ) is made publicly verifiable with $\mathsf{UVerify}$.

$\mathsf{UVerify}$ is an algorithm which verifies the validity of (m, τ) by using $\sigma \leftarrow \mathsf{Convert}(sk, m, \tau)$. More precisely, on input (pk, m, τ, σ), it outputs *accept* if (m, τ) is a valid U-pair and $\sigma \leftarrow \mathsf{Convert}(sk, m, \tau)$, and *reject* otherwise.

$\mathsf{Confirm}$ is a zero-knowledge proof system for valid U-pairs (m, τ).

$\mathsf{Disavow}$ is a zero-knowledge proof system for invalid U-pairs (m, τ).

A two-sided scheme has three modes as follows.

DS mode: ($\mathsf{Key}, \mathsf{DSign}, \mathsf{DVerify}$) is used as a DS scheme in an obvious way.

US mode: ($\mathsf{Key}, \mathsf{USign}, \mathsf{Confirm}, \mathsf{Disavow}$) is used as an US scheme in an obvious way.

Selective conversion mode: Convert and UVerify are used to convert an undeniable signature τ on m so that the validity of (m, τ) is made publicly verifiable.

The definitions of Convert and UVerify combine DS mode and US mode through selective conversion mode.

2.2 Security

In two-sided schemes, adversaries have three goals, DS-forgery, US-forgery and invisibility. In the attack game of each goal, we allow $\mathcal{A}$ to have oracle access to DSign-oracle, USign-oracle, Convert-oracle and Confirm/Disavow-oracle, where the last oracle is explained as follows. $\mathcal{A}$ queries (m, τ) to Confirm/Disavow-oracle. If (m, τ) is a valid U-pair, then the oracle returns *yes* and execute the protocol Confirm with $\mathcal{A}$. Otherwise, it returns *no* and execute the protocol Disavow with $\mathcal{A}$. In both cases, the oracle plays a role of the signer and $\mathcal{A}$ plays a role of the verifier.

We call DSign-oracle and USign-oracle *sign*-oracles, and Convert-oracle and Confirm/Disavow-oracle *decision*-oracles.

Sign-oracles	DSign-oracle, USign-oracle
Decision-oracles	Convert-oracle, Confirm/Disavow-oracle

Table 1. Sign-oracles and Decision-oracles

(1) We define DS-forgery as follows. Any adversary $\mathcal{A}$ can obtain a valid D-pair (m, σ) if $\mathcal{A}$ queries m to DSign-oracle or $\mathcal{A}$ queries a valid U-pair (m, τ) to Convert-oracle. (In the latter case, Convert-oracle returns σ.) We require that there is no other method for $\mathcal{A}$ to output a valid D-pair. Formally, we consider the following game. An adversary $\mathcal{A}$ is given a randomly generated public-key pk. $\mathcal{A}$ then has access to all oracles. Finally $\mathcal{A}$ outputs a forgery (m^*, σ^*).

We say that (m^*, σ^*) is not fresh if $\mathcal{A}$ queries m^* to DSign-oracle or $\mathcal{A}$ queries a valid U-pair (m^*, τ) to Convert-oracle for some τ. Otherwise we say that (m^*, σ^*) is fresh. We say that $\mathcal{A}$ DS-forges if (m^*, σ^*) is a valid D-pair, and it is fresh.

We show an example by using Table 2. In this example,

1. $\mathcal{A}$ queried m_i to DSign-oracle and received σ_i.
2. $\mathcal{A}$ queried m_j to USign-oracle and received τ_j. $\mathcal{A}$ next queried (m_j, τ_j) to Convert-oracle and received σ_j.
3. $\mathcal{A}$ queried m_k to USign-oracle and received τ_k.
4. $\mathcal{A}$ queried m_ℓ to no sign-oracle.

$\mathcal{A}$ finally outputs (m^*, σ^*). If (m^*, σ^*) is a valid D-pair and $m^* = m_\ell$, then $\mathcal{A}$ succeeds in DS-forgery. $\mathcal{A}$ also succeeds even if $m^* = m_k$. It is easy to see that (m^*, σ^*) is fresh in these cases.

Definition 1. *We say that a two-sided scheme is DS-secure if* $\Pr[\mathcal{A}$ *DS-forges*$]$ *is negligible for any PPT adversary* $\mathcal{A}$.

In selective convertible US schemes, $\mathcal{A}$ should not be able to forge a converter α for an already issued U-pair (m, τ). In two-sided schemes, this security notion is included in the above definition.

(2) We define US-forgery as follows. Suppose that an adversary $\mathcal{A}$ finally outputs a valid U-pair (m^*, τ^*), where $\mathcal{A}$ has never queried m^* to USign-oracle, but it queried m^* to DSign-oracle. Is it a forgery ? Our definitions of Sec.2.1, however, does not exclude the possibility that one can construct τ^* from a valid D-pair (m^*, σ^*). Indeed, this is the case in our constructions.

Hence we consider that $\mathcal{A}$ succeeds in US-forgery if $\mathcal{A}$ has never queried m^* to any sign-oracle. We say that a valid U-pair (m^*, τ^*) is fresh if $\mathcal{A}$ has never queried m^* to any sign-oracle. We also consider that $\mathcal{A}$ succeeds in US-forgery if she queries a fresh (m^*, τ^*) to one of the decision oracles during the attack game.

Formally, we consider the following game. An adversary $\mathcal{A}$ is given a randomly generated public-key pk. $\mathcal{A}$ then has access to all oracles. We say that $\mathcal{A}$ US-forges if $\mathcal{A}$ outputs a fresh (m^*, τ^*) or $\mathcal{A}$ queries a fresh (m^*, τ^*) to one of the decision-oracles.

Let's consider the example which is shown in the previous case (1) by using Table 2. Suppose that $\mathcal{A}$ finally outputs a valid U-pair (m^*, τ^*). If $m^* = m_\ell$, then $\mathcal{A}$ succeeds in US-forgery. However, $\mathcal{A}$ does not succeed if $m^* = m_i$.

Definition 2. *We say that a two-sided scheme is US-secure if* $\Pr[\mathcal{A}$ *US-forges*$]$ *is negligible for any PPT adversary* $\mathcal{A}$.

	m_i	m_j	m_k	m_ℓ
DSign-oracle	σ_i		(σ^*)	(σ^*)
USign-oracle		τ_j	τ_k	(τ^*)
$\mathsf{Convert}$-oracle		σ_j		

Table 2. Query pattern and DS/US forgery

(3) The third security notion is invisibility, a notion due to Chaum, van Heijst and Pfitzmann [9]. This notion is essentially the inability to determine whether a given U-pair is valid. We consider the following game on a distinguisher D.

1. D is given a randomly generated public-key pk. D then has access to all oracles.
2. At some point, D outputs a message m^* which has never been queried to any oracle, and requests a challenge undeniable signature $\tau^\dagger$ on m^*.
3. $\tau^\dagger$ is generated based on the outcome of a hidden coin toss b. If $b = 1$, then $\tau^\dagger$ is generated as usual using USign-oracle, otherwise $\tau^\dagger$ is chosen uniformly at random from the undeniable signature space $\mathcal{U}$.
4. D performs oracle queries again with the restriction that no sign-oracle query on m^* is allowed, and no decision-oracle query on $(m^*, \tau^\dagger)$ is allowed.

5. At the end of this attack game, D outputs a guess b'.

Define $Adv_{inv}(D) = |\Pr(b' = b) - (1/2)|$.

Definition 3. *A two-sided scheme is invisible if $Adv_{inv}(D)$ is negligible for any PPT D.*

Definition 4. *We say that a two-sided scheme is secure if it is DS-secure, US-secure and invisible.*

3 Proposed Two-Sided Scheme in RO Model

Now we show an efficient two-sided scheme in the random oracle model based on RSA and Paillier's encryption scheme [29].

3.1 Paillier's Encryption Scheme

In Paillier's encryption scheme [29], the public-key is $N(= pq)$, and the private-key is (p, q), where p and q are large primes. The encryption function for a message $m \in Z_N$ is given by

$$E(m, r) = r^N (1 + mN) \bmod N^2,$$

where $r \in Z_N^*$ is randomly chosen. E has a homomorphic property such that

$$E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2 \bmod N, r_1 r_2 \bmod N) \bmod N^2.$$

(For decryption, see [29].) We say that $Y \in Z_{N^2}^*$ is an Nth residue if $Y = x^N \bmod N^2$ for some $x \in Z_N^*$. Note that $E(0, r)$ is an Nth residue.

3.2 Proposed Scheme

The proposed two-sided scheme is described as follows. Let $m \in \{0, 1\}^*$ be a message.

- **Key Generation.** On input 1^ℓ, choose two primes p, q such that $|p| = |q| = \ell$ randomly and compute $N = pq$. Find d such that $Nd = 1 \bmod lcm(p - 1, q - 1)$. Let $H : \{0, 1\}^* \to Z_N^*$ be a hash function. Set the public key as $pk = (N, H)$ and the secret key as d.
- **DSign.** Compute $\sigma = H(m)^d \bmod N$ and return σ as the digital signature.
- **DVerify.** For a given (m, σ), output *accept* if $\sigma^N = H(m) \bmod N$ and *reject* otherwise.
- **USign.** First compute $\sigma = H(m)^d \bmod N$. Next compute τ such that

$$\sigma^N = H(m) + \tau N \bmod N^2. \qquad (2)$$

  Finally return τ as the undeniable signature.

- Convert. For a given (m, τ), first compute $\sigma = H(m)^d \bmod N$. Next output σ if eq.(2) is satisfied, and $\perp$ otherwise.
- UVerify. For a given (m, τ, σ), output *accept* if eq.(2) is satisfied, and *reject* otherwise.

For confirmation/disavowal protocols, we use the following Lemma.

Lemma 1. *(m, τ) is a valid U-pair if and only if there exists $\sigma \in Z_N^*$ such that*

$$E(0, \sigma) = H(m) + \tau N \bmod N^2,$$

where E is an encryption function of Paillier's encryption scheme.

The proof is clear from eq.(2). Now given (m, τ), the signer computes $\beta \in Z_N$ such that

$$E(\beta, \sigma) = H(m) + \tau N \bmod N^2. \tag{3}$$

If $\beta = 0$, then the signer runs a confirmation protocol which proves that $\beta = 0$. Otherwise, the signer runs a disavowal protocol which proves that $\beta \neq 0$.

We will show efficient protocols based on the homomorphic property of Paillier's encryption scheme [29].

3.3 Confirmation Protocol

We first show a basic confirmation protocol which proves that $\beta = 0$ in eq.(3).

1. The verifier chooses $u, v \in Z_N$ and $w \in Z_N^*$ randomly, and compute

   $$y = (H(m) + \tau N)^u E(v, w) \bmod N^2.$$

   He then sends y to the signer. Note that it holds that for some $r \in Z_N^*$,

   $$y = E(0, \sigma)^u E(v, w) = E(0 \times u + v, r) = E(v, r) \bmod N^2.$$

2. By using the decryption algorithm of Paillier's encryption scheme, the signer decrypts y and obtains v' such that $y = E(v', r')$ for some r'. Then she sends v' to the verifier.
3. The verifier accepts if $v' = v$, and rejects otherwise.

Theorem 1. Completeness. *If (m, τ) is a valid U-pair, then the verifier always accepts.*

Soundness. *If (m, τ) is not a valid U-pair, then the verifier rejects with overwhelming probability.*

The proof is given in Appendix A. Finally, we construct a zero-knowledge confirmation protocol as follows, where *commit*(x) is a commitment function.

1. The verifier sends

   $$y = (H(m) + \tau N)^u E(v, w) \bmod N^2 \tag{4}$$

   to the signer, where $u, v \in Z_N$ and $w \in Z_N^*$ are randomly chosen.

2. The signer computes v' such that $y = E(v', r')$, and sends $c = commit(v')$ to the verifier.
3. The verifier reveals u, v, w.
4. The signer checks if eq.(4) holds by using u, v, w. If it holds, then the signer opens $c = commit(v')$. Otherwise, she aborts.
5. The verifier accepts if $v' = v$, and rejects otherwise.

Theorem 2. *The above protocol is zero-knowledge confirmation protocol if (i) $commit(x)$ reveals no information on x, and (ii) the signer cannot find x' such that $commit(x) = commit(x')$.*

The proof will be given in the final version. In the random oracle model, we can use a simple $commit(x)$ shown by Pass [30, Sec.4.1] as follows.

Commit phase. For $x \in Z_N$, Alice chooses $r \in Z_N^*$ randomly and sends $c = H(x, r)$ to Bob.
Reveal phase. Alice sends (x, r) to Bob. Bob checks that $c = H(x, r)$.

3.4 Disavowal Protocol

We first show a basic disavowal protocol which proves that $\beta \neq 0$ in eq.(3).

1. The verifier chooses $u \in Z_N$ and $w \in Z_N^*$ randomly, and computes

$$y = (H(m) + \tau N)^u E(0, w) \bmod N^2.$$

   He sends y to the signer. Note that for some $r \in Z_N^*$,

$$y = E(\beta, \sigma)^u E(0, w) = E(\beta \times u \bmod N, r) \bmod N^2. \quad (5)$$

2. The signer first computes x such that $y = E(x, r')$, where $x = \beta \cdot u \bmod N$ from eq.(5). She next computes $u' = x/\beta \bmod N$. Then she sends u' to the verifier.
3. The verifier accepts if $u' = u$, and rejects otherwise.

Similarly to Theorem 1, we can prove the following theorem.

Theorem 3. Completeness. *If (m, τ) is not a valid U-pair, then the verifier always accepts.*
Soundness. *If (m, τ) is a valid U-pair, then the verifier rejects with overwhelming probability.*

Finally we construct a zero-knowledge disavowal protocol as follows, where $commit(x)$ is a commitment function given in the previous subsection.

1. The verifier sends

$$y = (H(m) + \tau N)^u E(0, w) \bmod N^2 \quad (6)$$

   to the signer, where $u \in Z_N$ and $w \in Z_N^*$ are randomly chosen.

2. The signer first computes β of eq.(3) and x such that $y = E(x, r')$. She next computes $u' = x/\beta \bmod N$. Then she sends $c = commit(u')$ to the verifier.
3. The verifier reveals u, w.
4. The signer checks if eq.(6) holds by using u, w. If it holds, then the signer opens $c = commit(u')$. Otherwise, she aborts.
5. The verifier accepts if $u' = u$, and rejects otherwise.

Theorem 4. *The above protocol is zero-knowledge disavowal protocol if (i) $commit(x)$ reveals no information on x, and (ii) the signer cannot find x' such that $commit(x) = commit(x')$.*

The proof will be given in the final version.

3.5 Security of Our Scheme

RSA assumption with $e = N$ (N-RSA Problem) claims that given an RSA modulus N and a random $y \in Z_N^*$, it is hard to compute $x \in Z_N^*$ such that $y = x^N \bmod N$. We now define the N^2-RSA problem as follows. Given an RSA modulus N and a random Nth residue $Y \in Z_{N^2}^*$, compute $x \in Z_N^*$ such that $Y = x^N \bmod N^2$. The N^2-RSA assumption claims that the N^2-RSA problem is hard. We then prove that the proposed scheme is DS-secure under the N^2-RSA assumption.

Theorem 5. *The proposed scheme is DS-secure under the N^2-RSA assumption in the random oracle model.*

The proof is given in Appendix B. It use the techniques of Coron [12] which was also used by [28].

Given an RSA modulus N and a random $y \in Z_N^*$, the computational Nth Residuosity (CNR) problem is to find $z \in Z_N$ such that $y + zN = x^N \bmod N^2$ for some $x \in Z_N^*$. The CNR assumption claims that the CNR problem is hard. Catalano et al. proved that CNR problem is as intractable as the one-wayness of Paillier cryptosystem [6]. We prove that the proposed scheme is US-secure under the CNR assumption.

Theorem 6. *The proposed scheme is US-secure under CNR assumption in the random oracle model.*

The proof will be given in the final paper.

Let $\mathsf{Residue}_N = \{Y \mid Y = x^N \bmod N^2 \text{ for some } x \in Z_N^*\}$. Decisional Nth Residuosity (DNR) assumption claims that $\mathsf{Residue}_N$ and $Z_{N^2}^*$ are indistinguishable. More precisely, we consider the following game between a challenger and a distinguisher D. For a given $N(= pq)$:

1. The challenger chooses a random bit b. If $b = 0$, then he chooses Y from $\mathsf{Residue}_N$ randomly. If $b = 1$, then he chooses Y from $Z_{N^2}^*$ randomly. He then gives Y to D.

2. D outputs a bit b'.

Define $Adv_{dnr}(D) = |\Pr(b' = b) - (1/2)|$. The DNR assumption claims that $Adv_{dnr}(D)$ is negligible for any PPT distinguisher D. This problem was first addressed in Paillier cryptosystem, namely Paillier cryptosystem is IND-CPA under DNR assumption [29].

We prove that the proposed scheme is invisible under DCR assumption.

Theorem 7. *The proposed scheme is invisible under DNR assumption in the random oracle model.*

The proof will be given in the final paper.

It is easy to see that the following reductions hold for the underlying problems.

1. N-RSA Problem $\Rightarrow$ CNR Problem $\Rightarrow$ DNR Problem,
2. N-RSA Problem $\Rightarrow$ N^2-RSA Problem,
3. CNR Problem + N^2-RSA Problem $\Rightarrow$ N-RSA Problem.

4 How To Remove Random Oracle

In this section, we show an efficient two-sided scheme in the standard model. Cramer-Shoup showed an adaptively secure DS scheme under strong RSA assumption in the standard model [13]. It can be seen as a special case of Shamir-Tauman construction [32] which transforms a weakly secure DS scheme (secure against weak non-adaptive chosen message attack) to an adaptively secure one by combining with a trapdoor commitment scheme. In particular, in Cramer-Shoup scheme, a trapdoor commitment scheme is based on GQ identification scheme [15].

Our two-sided scheme is constructed by modifying Cramer-Shoup DS scheme as follows. First, our DSign algorithm is almost the same as Cramer-Shoup DS scheme except that we use two moduli, $N_1(= p_1q_1)$ for GQ-based trapdoor commitment scheme and $N_2(= p_2q_2)$ for a weakly secure signature part, while Cramer-Shoup scheme uses a single modulus. Next our USign algorithm is obtained by extending our technique of Sec.3 to the GQ-based trapdoor commitment scheme.

4.1 Scheme

(Key Generation) Let ℓ be a security parameter.

1. Choose four ℓ-bit primes p_1, q_1, p_2, q_2 randomly such that $p_2 = 2p' + 1$ and $q_2 = 2q' + 1$, where p' and q' are primes. Let $N_1 = p_1q_1$ and $N_2 = p_2q_2$.
2. Choose $h_1 \in Z^*_{N_1}$ and $h_2, x \in QR_{N_2}$ randomly, where QR_N denotes the set of quadratic residues of $\bmod N$.

3. Find d such that $N_1 d = 1 \bmod lcm(p_1 - 1, q_1 - 1)$. Let H be a collision-resistant hash function whose output can be interpreted as a positive integer less than 2^ℓ.
4. Set the public-key as $pk = (N_1, h_1, N_2, h_2, x, H)$ and the secret-key as $sk = (d, p_2, q_2)$.

DSign. For a message $m \in \{0,1\}^*$, first choose $y' \in Z_{N_1}^*$ randomly and compute $x' \in Z_{N_1}$ such that

$$(y')^{N_1} = x' h_1^{H(m)} \bmod N_1, \tag{7}$$

(where x' can be seen as a commitment of m). Next choose a $(\ell+1)$-bit prime e randomly and compute y such that

$$y^e = x h_2^{H(x')} \bmod N_2, \tag{8}$$

(where (e, y) is a weakly secure signature on x'). The digital signature on m is $\sigma = (e, y, y')$.

DVerify. For a given (m, σ), first check if e is an $(\ell+1)$-bit number. Second, $x' = (y')^{N_1} h_1^{-H(m)} \bmod N_1$ is computed. Third, it is checked that $x = y^e h_2^{-H(x')} \bmod N_2$.

USign. For a message $m \in \{0,1\}^*$, first compute $\sigma = (e, y, y')$ as shown in DSign. Next compute $\omega \in Z_{N_1}$ such that

$$(y')^{N_1} = u + \omega N_1 \bmod N_1^2, \tag{9}$$

where $u = x' h_1^{H(m)} \bmod N_1$. Finally return $\tau = (e, y, x', \omega)$ as the undeniable signature on m. (Note that the above equation is basically the same as eq.(2)).

Convert. For a given m and $\tau = (e, y, x', \omega)$, first check if e is an $(\ell+1)$-bit number and (e, y, x') satisfies eq.(8). Next compute $y' \in Z_{N_1}$ which satisfies eq.(7). Finally check if (y', ω) satisfies eq.(7). If everything is OK, then output $\sigma = (e, y, y')$. Otherwise, output $\perp$.

UVerify. For a given m, $\tau = (e, y, x', \omega)$ and $\sigma = (e, y, y')$, output *accept* if e is an $(\ell+1)$-bit number, and eq.(7), eq.(8) and eq.(9) are satisfied, and *reject* otherwise.

In the confirmation protocol, the signer proves that for a valid U-pair, m and $\tau = (e, y, x', \omega)$, there exists $\sigma = (e, y, y')$ which satisfies eq.(7), eq.(8) and eq.(9). Essentially, this means that the signer proves that there exists $y' \in Z_{N_1}$ which satisfies eq.(9). Such a zero-knowledge protocol can be constructed similarly to Sec.3.3.

In the disavowal protocol, the signer proves that for an invalid U-pair m and $\tau = (e, y, x', \omega)$, there exists no $\sigma = (e, y, y')$ which satisfies eq.(7), eq.(8) and eq.(9). If eq.(8) is not satisfied, then we have done. If eq.(8) is satisfied, then the signer proves that there exists no $y' \in Z_{N_1}$ which satisfies eq.(9). Such a zero-knowledge protocol can be constructed similarly to Sec.3.4.

In these protocols, we can use a commitment function based on RSA assumption as shown in [20, Sec.3]. Also, see [18, page 405].

4.2 Security

The strong RSA assumption claims that given an RSA modulus N and a random $y \in Z_N^*$, it is hard to find $e > 1$ and $x \in Z_N^*$ such that $y = x^e \bmod N$

We define the strong CNR problem as follows. Given an RSA modulus N and a random $z \in Z_N^*$, find $a > 1$ and $c \in Z_N$ such that $w = z^a + cN \bmod N^2$ is an Nth residue. Solving the CNR problem implies an algorithm for solving the strong CNR problem, but the other direction is unknown. The strong CNR assumption claims that the strong CNR problem is hard.

Theorem 8. *The above scheme is US-secure under the strong RSA assumption and the strong CNR assumption in the standard model.*

Theorem 9. *The above scheme is DS-secure under the strong RSA assumption and the strong CNR assumption in the standard model.*

Theorem 10. *The above scheme is invisible under DNR assumption in the standard model.*

All the proofs will be given in the final paper.

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. J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. *EUROCRYPT '00*, LNCS 1870, pp.243–258, Springer-Verlag, 2000.
5. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. *CRYPTO '03*, LNCS 2729, pp.126–144, Springer-Verlag, 2003.
6. 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.
7. D. Chaum. Zero-knowledge undeniable signatures. *EUROCRYPT '90*, LNCS 473, pp.458–464, Springer-Verlag, 1990.
8. D. Chaum. Designated confirmer signatures. *EUROCRYPT '94*, LNCS 950, pp.86–91, Springer-Verlag, 1995.
9. 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.
10. D. Chaum and H. van Antwerpen. Undeniable signatures. *CRYPTO '89*, LNCS 435, pp.212–216, Springer-Verlag, 1989.
11. T. Chaum and T. P. Pedersen. Wallet databases with observers. *CRYPTO '92*, LNCS 740, pp.89–105, Springer-Verlag, 1993.

12. J. -S. Coron. On the exact security of full domain hash. *CRYPTO '00*, LNCS 1880, pp.229–235, Springer-Verlag, 2000.
13. 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.
14. I. Damgård and T. Pedersen. New convertible undeniable signature schemes. *EUROCRYPT '96*, LNCS 1070, pp.372–386, Springer-Verlag, 1996.
15. 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.
16. 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.
17. 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.
18. R. Gennaro, T. Rabin and H. Krawczyk. RSA-based undeniable signatures. *Journal of Cryptology*, 13(4), pp.397–416, 2000.
19. M. Jakobsson, K. Sako and R. Impagliazzo. Designated verifier proofs and their applications. *EUROCRYPT '96*, LNCS 1070, pp.143–154, Springer-Verlag, 1996.
20. K. Kurosawa and S. Heng: The Power of identification schemes. *PKC '06*, LNCS 3958, pp.364–377, Springer-Verlag, 2006.
21. K. Kurosawa and S. Heng: Relations among security notions for undeniable signature schemes. accepted by SCN 2006.
22. B. Libert and J.-J Quisquater. Identity based undeniable signatures. *CT-RSA '04*, LNCS 2964, pp.112–125, Springer-Verlag, 2004.
23. F. Laguillaumie and D. Vergnaud: Short undeniable signatures without random oracles: The Missing Link. *INDOCRYPT '05*, LNCS 3797, pp.283–296, Springer-Verlag, 2005.
24. M. Michels, H. Petersen and P. Hoster. Breaking and repairing a convertible undeniable signature scheme. In *3rd ACM CCCS*, pp.148–152, 1996.
25. M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. *SAC '97*, pp.231–244, Springer-Verlag, 1997.
26. 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.
27. J. Monnerat and S. Vaudenay. Generic homomorphic undeniable signatures. *ASIACRYPT '04*, LNCS 3329, pp.354–371, Springer-Verlag, 2004.
28. 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.
29. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *EUROCRYPT '99*, LNCS 1592, pp.223–238, Springer-Verlag, 1999.
30. R. Pass. On deniability in the common reference string and random oracle model. *CRYPTO '03*, LNCS 2729, pp.316–337, Springer-Verlag, 2003.
31. D. Pointcheval. Self-scrambling anonymizers. *FC '00*, LNCS 1962, pp.259–275, Springer-Verlag, 2000.
32. A. Shamir and Y. Tauman. Improved online/offline signature schemes. *CRYPTO '01*, LNCS 2139, pp.355–367, Springer-Verlag, 2001.

A Proof of Theorem 1

The completeness is clear. We prove the soundness. Suppose that (m, τ) is not a valid U-pair. Then we can write

$$E(\beta, \sigma) = H(m) + \tau N \bmod N^2$$

for some $\beta \in Z_N$ and $\sigma \in Z_N^*$, where $\beta \neq 0$ from Lemma 1. Then y is written as

$$y = E(\beta, \sigma)^u E(v, w) = E(t, r),$$

where

$$t = \beta \cdot u + v \bmod N \text{ and } r = \sigma^u \cdot w \bmod N.$$

Now it is easy to see that for any $v' \in Z_N$, there exists unique $u', w' \in Z_N$ such that

$$t = \beta \cdot u' + v' \bmod N \text{ and } r = \sigma^{u'} \cdot w' \bmod N$$

if $\gcd(\beta, N) = 1$. This means that the prover cannot compute v correctly more than guessing. Hence the verifier rejects with overwhelming probability.

B Proof of Theorem 5

We show that if there exists a PPT adversary $\mathcal{A}$ with $\Pr[\mathcal{A}$ DS-forges$] = \epsilon_A$, then one can construct a PPT algorithm M that can solve the N^2-RSA problem with probability ϵ_M, by running $\mathcal{A}$ as a subroutine. Suppose the input to M is (N, Y), where $Y = x^N \bmod N^2$ for some $x \in Z_N^*$.

M then starts running $\mathcal{A}$ by feeding $\mathcal{A}$ with the public key (N, H) where H is a random oracle that will be simulated by M. M also simulates the sign-oracles and the decision-oracles itself.

We assume that when $\mathcal{A}$ requests a sign-oracle query or a decision-oracle query on a message m_i, it has already made the corresponding H query on m_i. When $\mathcal{A}$ makes a H-oracle query for a message m_i, M chooses $r_i \in Z_N^*$ randomly and behaves as follows.

- With probability δ, return $h_i = H(m_i) = r_i^N \bmod N$. Let $flag_i = 0$, $\sigma_i = r_i$, and compute $\tau_i \in Z_N^*$ such that $r_i^N = h_i + \tau_i N \bmod N^2$.
- With probability $1 - \delta$, return $h_i = H(m_i) = Y r_i^N \bmod N$. Let $flag_i = 1$, and compute $\tau_i \in Z_N^*$ such that $r_i^N Y = h_i + \tau_i N \bmod N^2$.

In the above, δ is a fixed probability which will be determined later.

Suppose that $\mathcal{A}$ makes a sign-oracle query for a message m_i.

- Suppose that $flag_i = 0$. If the query is a DSign-oracle query, then M returns σ_i. If it is a USign-oracle query, then M returns τ_i.
- Suppose that $flag_i = 1$. If the query is a USign-oracle query, then M returns τ_i. If the query is a DSign-oracle query, then M aborts and it fails to solve N^2-RSA problem.

$flag_i$		DSign-oracle query	USign-oracle query
0	$r_i^N = h_i + \tau_i N \bmod N^2$	$\sigma_i = r_i$	τ_i
1	$Y r_i^N = h_i + \tau_i N \bmod N^2$	Abort	τ_i

Next, suppose $\mathcal{A}$ makes a decision-oracle query for (m_i, τ_i').

- Suppose that $\tau_i' \neq \tau_i$. If the query is a Convert-oracle query, then M returns $\perp$. If the query is a Confirm/Disavow-oracle query, then M returns *no* and runs the disavowal protocol with $\mathcal{A}$.
- Otherwise, $\tau_i' = \tau_i$. If the query is a Confirm/Disavow-oracle query, then M returns *yes* and runs the confirmation protocol with $\mathcal{A}$.
  Suppose that the query is a Convert-oracle query. If $flag_i = 0$, then M returns σ_i. If $flag_i = 1$, then M aborts and it fails to solve N^2-RSA problem.

In the above, M can simulate the Confirm/Disavow oracle by using the rewinding technique because the protocols are zero-knowledge.

Now suppose that $\mathcal{A}$ DS-forges, and outputs a valid D-pair (m^*, σ^*) at the end of the game. We assume that $\mathcal{A}$ has queried the H-oracle on m^* and so $m^* = m_j$ for some j.

- If $flag_j = 0$, then M aborts.
- Otherwise, $flag_j = 1$. Since (m^*, σ^*) is a valid D-pair, it holds that

$$h_j + \tau_j N = (\sigma^*)^N \bmod N^2.$$

  On the other hand, $r_j^N Y = h_j + \tau_j N \bmod N^2$ since $flag_j = 1$. Therefore, it holds that

$$r_j^N Y = (\sigma^*)^N \bmod N^2.$$

$$Y = (\sigma^*/r_j)^N \bmod N^2.$$

  Now let $x = \sigma^*/r_j \bmod N$. Then it is easy to show that $x^N = (\sigma^*/r_j)^N \bmod N^2$. Therefore, it holds that

$$Y = x^N \bmod N^2.$$

  Consequently, M outputs $x \in Z_N^*$ and thus it solves N^2-RSA problem.

To complete the proof, it remains to calculate the probability that M does not abort. Let q_D be the number of DSign-oracle queries and that $\mathcal{A}$ issues. The probability that M answers to all DSign-oracle queries is δ^{q_D}, and $flag_j = 1$ for $m_j = m^*$ is $1 - \delta$. Therefore, the probability that M does not abort during the simulation is $\delta^{q_D}(1-\delta)$. This value is maximized at $\delta_{opt} = 1 - 1/(q_D + 1)$. This shows that ϵ_M is at least $(1/e(1 + q_D))\epsilon_A$, where e is the base of the natural logarithm. This is because the value $(1 - 1/(q_D + 1))^{q_D}$ approaches $1/e$ for large q_S. This completes our proof.