3-Move Undeniable Signature Scheme

Kaoru Kurosawa[1] and Swee-Huay Heng[2]

[1] Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki 316-8511, Japan
kurosawa@cis.ibaraki.ac.jp
[2] Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia
shheng@mmu.edu.my

Abstract. In undeniable signature schemes, zero-knowledgeness and non-transferability have been identified so far. In this paper, by separating these two notions, we show the first 3-move confirmation and disavowal protocols for Chaum's undeniable signature scheme which is secure against active and concurrent attacks. Our main observation is that while the signer has one public key and one secret key, there exist two witnesses in the confirmation and disavowal proofs of Chaum's scheme.

Keywords: Undeniable signature, witness indistinguishability

1 Introduction

1.1 Background and Motivation

The concept of undeniable signatures was due to Chaum and van Antwerpen [11]. As opposed to the standard digital signatures which are universally verifiable, the validity of undeniable signatures can be verified only with the signer's consent, by engaging interactively or non-interactively in either a confirmation protocol or a disavowal protocol. There have been a wide range of research covering a variety of different schemes for undeniable signatures over the past 15 years. Among others, we have [8, 3, 10, 9, 22, 15, 20, 24, 6, 19, 18, 23, 5, 25, 26]. Most of these schemes are discrete logarithm based, with the exception of a few RSA-based schemes [20, 19, 18], a pairing-based (identity-based) scheme [23] and some other schemes [5, 25, 26]. These schemes possess variable degrees of security and additional features such as convertibility [3, 15, 24, 18], designated verifier technique [22], designated confirmer technique [9, 27], and so on. At the same time, undeniable signatures also find various applications in cryptography such as in licensing softwares (this is in fact the original motivation of Chaum and van Antwerpen) [11], electronic cash [12, 4, 28], electronic voting and auctions.

An undeniable signature scheme is said to be secure (against active attack) if it is unforgeable, invisible and the confirmation and disavowal protocols are both zero-knowledge. The zero-knowledgeness is required to make undeniable signatures non-transferable, which is indeed the purpose of undeniable signature schemes. Further, it is believed that a 3-move protocol cannot be zero-knowledge from the result of [17].

Therefore, no 3-move undeniable signature scheme which is secure against active attack is known. In fact, in the existing literature, the zero-knowledge confirmation protocol is at least 4 moves. No constant moves zero-knowledge disavowal protocol has been known so far.

This is mainly because zero-knowledgeness and non-transferability have been identified so far implicitly or explicitly. In other words, the search for a 3-move undeniable signature scheme which is provably secure against active attack remains as an challenging open problem since the introduction of the concept of undeniable signatures in 1989.

1.2 Our Contributions

We say that an undeniable signature scheme is "3-move" if the confirmation protocol and the disavowal protocol are both 3-move (where the signer S sends a to the verifier V, V sends b to S and S sends c to V).

In this paper, we propose the first "3-move" undeniable signature scheme which is provably secure against active and concurrent attacks, by exploiting the fact that DH-tuples possess two witnesses, and also that non DH-tuples possess two witnesses. It is achieved

– by separating two notions, zero-knowledgeness and non-transferability, and
– by incorporating the concept of witness indistinguishability [16] in a novel way.

A naive approach for witness indistinguishability would be to use two public keys, where the two corresponding secret keys are two witnesses, as suggested by Feige and Shamir in [16]. More precisely, the signer issues two undeniable signatures σ_1 and σ_2 on a message m. He then proves that σ_1 is valid or σ_2 is valid by a witness indistinguishable protocol. Unfortunately, this approach does not work. This is because, from De Morgan's law ($\overline{X \vee Y} = \overline{X} \wedge \overline{Y}$), both the confirmation protocol and the disavowal protocol cannot be witness indistinguishable simultaneously. For more details, see Section 6. Further, such a two public-key scheme would be costly.

On the other hand, we show 3-move confirmation and disavowal protocols for Chaum's undeniable signature scheme [8], where the signer has only one public key. Our main idea is as follows. In the confirmation (disavowal) protocol, the signer proves that a tuple (g, g^u, g^v, g^w) is a DH-tuple (non DH-tuple). Now observe that a DH-tuple (g, g^u, g^v, g^{uv}) has two witnesses, u and v. A similar observation holds for non DH-tuples too. Thus, our main observation is that while the signer has one public key and one secret key, there exist two witnesses in the confirmation and disavowal proofs of Chaum's scheme. This allows us to use the concept of witness indistinguishability (WI) in the confirmation and disavowal protocols. As a result, we manage to circumvent the problem encountered earlier in the naive approach.

More precisely, in order to prove that (g, g^u, g^v, g^{uv}) is a DH-tuple, knowledge of either one of the two witnesses, i.e. u or v is sufficient. This observation is critical in the simulation of the confirmation/disavowal oracle in the security analyses.

Chaum's original scheme (which does not employ a cryptographic hash function) is not secure as it succumbed to the basic multiplicative attacks. Therefore, we apply the above idea on the full-domain hash (FDH) [14] variant of Chaum's scheme so that we can treat the hash function as a random oracle in the security analyses. In this scheme, the signer has a single public key $y = g^x$ and a single secret key x. Remember that the signer does not have to prove that he knows x in the confirmation protocol. All he needs to prove is the validity of a signature σ on a message m under the public key y, i.e. he proves that a given tuple $(g, y, H(m), \sigma)$ is a DH-tuple where $\sigma = H(m)^x$ and H is a random oracle. We notice that the DH-tuple has two witnesses by accident in this case. The same argument applies in proving the invalidity of a signature $\sigma \neq H(m)^x$, i.e. by proving that $(g, y, H(m), \sigma)$ is a non DH-tuple using a disavowal protocol.

Traditionally, two main security notions for undeniable signatures are the notion of existential unforgeability and invisibility under adaptive chosen message attack. The existential unforgeability of our proposed scheme is equivalent to the computational Diffie-Hellman (CDH) problem while it is invisible assuming the hardness of the decisional Diffie-Hellman (DDH) problem.

In this paper, we also introduce another important security notion with regard to undeniable signatures, namely, the security against impersonation attack. As all of us are aware that the purpose of undeniable signature scheme is to construct a signature which is non-transferable. This is equivalent to prevent impersonation by employing confirmation and disavowal protocols. This security notion has not been formalized so far mainly because zero-knowledgeness implies non-transferability. Since our newly proposed scheme is a novel work separating zero-knowledgeness and non-transferability while adopting the witness-indistinguishability property, we are led to this particular security notion. We manage to prove that the security against impersonation attack of our proposed scheme is equivalent to the discrete logarithm (DLOG) problem.

A brief summary of security analyses of our newly proposed scheme is given in the following table. The table holds in the random oracle model while our confirmation (disavowal) protocol is a WI protocol for a (non-) DH tuple in the standard model.

Security	Unforgeability	Invisibility	Impersonation
Equivalence	CDH	$\geq$ DDH	DLOG

As a result, we successfully devise a 3-move undeniable signature scheme which is secure against active and concurrent attacks with respect to the security notions of existential unforgeability, invisibility and impersonation, with a weaker requirement than zero-knowledgeness.

Notice that schemes which adopt the non-interactive designated verifier proof technique [22] is trivially secure against impersonation attack. It is true that if a verifier has a public key, then we can make the confirmation and disavowal protocols non-interactive by using the designated verifier proof technique. However, if the verifier does not have a public key, then obviously the confirmation and disavowal protocols must remain interactive. Otherwise, non-transferability is broken.

As an aside, we remark that the idea we obtain from the observation that there exist two witnesses in a DH-tuple is of independent interest. Hopefully, it may find applications on some other interactive protocols which involve the proving of the validity of a DH-tuple in the security analyses. For example, the same idea can be readily applied to the identity-based undeniable signatures by Libert and Quisquater [23]. In [23], the scheme is proven secure using the non-interactive designated verifier proof technique only.

2 Preliminaries

Let G be an Abelian group of prime order q, and let g be a generator of G. We say that (g, g^u, g^v, g^w) is a DH-tuple if $w = uv \bmod q$.

The DDH problem is to decide if (g, g^u, g^v, g^w) is a DH-tuple. The CDH problem is to compute g^{uv} from (g, g^u, g^v) and the DLOG problem is to compute u from g^u.

2.1 The FDH Variant of Chaum's Scheme

The full domain hash (FDH) variant [14] of Chaum's undeniable signature scheme [8] is described as follows.

Let G be an Abelian group of prime order q, and let g be a generator of G.

- **Key Generation.** On input 1^k, choose $x \in Z_q$ randomly and compute $y = g^x$. Choose a cryptographic hash function $H : \{0,1\}^* \rightarrow G$. 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 pair (m, σ), the signer proves that $(g, y, H(m), \sigma)$ is not a DH-tuple in zero-knowledge.

The existing zero-knowledge confirmation protocol requires 4 moves and no zero-knowledge disavowal protocol with constant moves is known so far [8].

3 How to Formalize the Security

Traditionally, an adversary in an undeniable signature scheme intends to achieve two main adversarial goals, namely, to forge a signature and to distinguish whether a message-signature pair is valid or invalid, which correspond to the security notions of existential unforgeability and invisibility, respectively.

In this section, we introduce another new adversarial goal which is *impersonation*. We are led to this notion since our proposed confirmation/disavowal protocols are not zero-knowledge but witness indistinguishable, linking impersonation attack for identification schemes to non-transferability. Surprisingly, this goal *impersonation* has been overlooked in the past while being different from the notions of unforgeability and invisibility.

Meanwhile, there exist three kinds of attacks, namely, passive attack, active attack and concurrent attack. We will elaborate more on this in the sequel.

3.1 Unforgeability

The first security notion is similar to the one for ordinary digital signatures, which is the notion of existential unforgeability against adaptive chosen message attack [21]. The only difference is that besides the signing oracle access, the adversary is also allowed to access to the confirmation/disavowal oracle. The confirmation/disavowal oracle is simulated based on the kind of attacks mounted, i.e. passive attack or active/concurrent attack. Generally, in a passive attack the adversary does not interact with the signer/prover. What the adversary does is eavesdropping and she is in possession of transcripts of conversations between the prover and the verifier. In an active/concurrent attack, the adversary gets to play the role of a cheating verifier, interacting with the prover several times, in an effort to extract some useful information. We will give a more formal definition shortly.

To the best of our knowledge, this is the first time passive, active and concurrent attacks are being defined explicitly and rigorously with respect to the security notions of undeniable signatures. Specifically, concurrent attack is more relevant to identification scheme [2]. However, we remark that since confirmation and disavowal protocols of an undeniable signature scheme are usually performed interactively as in an identification protocol, concurrent attack should be taken into account as well.

The difference between active and concurrent attacks is that in an active attack, the adversary interacts serially with the prover "clones"; while in a concurrent attack, the adversary is allowed to interact with many different prover "clones" concurrently. Apparently, the active/concurrent adversary has higher capability than the passive adversary.

We consider the following game.

1. Let pk be the input to a forger F.
2. The forger F is permitted to issue a series of queries:
   - Signing queries: F submits a message m and receives a signature σ on m. (We consider adaptive queries here – subsequent queries is made based on previously obtained signatures.)
   - Confirmation/disavowal queries: F submits a message-signature pair (m, σ), and the oracle responds based on whether a passive attack or an active/concurrent attack is mounted.
3. At the end of this attack game, F outputs a message-signature pair (m^*, σ^*).

- In a passive attack, the confirmation/disavowal oracle first checks the validity of (m, σ). If it is a valid pair, then the oracle returns a bit $\mu = 1$ and

a transcript of confirmation protocol. Otherwise, the oracle returns a bit $\mu = 0$ and a transcript of disavowal protocol.
– In an active/concurrent attack, the confirmation/disavowal oracle first checks the validity of (m, σ). If it is a valid pair, then the oracle returns a bit $\mu = 1$ and proceeds with the execution of the confirmation protocol with the forger F (acting as a cheating verifier). Otherwise, the oracle returns a bit $\mu = 0$ and executes the disavowal protocol with F accordingly.

The forger F wins the game if F outputs a valid message-signature pair (m^*, σ^*) such that m^* has never been queried to the signing oracle, or it queries a valid (m^*, σ^*) to the confirmation/disavowal oracle such that m^* has never been queried to the signing oracle.

F's advantage in this game is defined to be $Adv(F) = \Pr[F \text{wins}]$.

Definition 1. *An undeniable signature scheme is said to be existential unforgeable under adaptive chosen message attack if no probabilistic polynomial time (PPT) forger F has a non-negligible advantage in the above game.*

3.2 Invisibility

The second security notion of undeniable signatures is invisibility, a notion due to Chaum, van Heijst and Pfitzmann [10]. This notion is essentially the inability to determine whether a given message-signature pair is valid. There are many variations in defining invisibility, for example it is defined in terms of simulatability in [10] and it is defined in terms of distinguishing whether a signature σ is corresponding to a message m_0 or m_1 in [6].

In this paper, we adopt the following definition given by Galbraith and Mao [18] as they have proven that if a scheme satisfies invisibility in the sense of Definition 2 then it also satisfies invisibility in the sense of [6].

Consider the following game.

1. Let pk be the input to a distinguisher D.
2. The distinguisher D is permitted to issue a series of queries: signing queries and confirmation/disavowal queries as in Section 3.1.
3. At some point, D outputs a message m^* which has never been queried to the signing oracle, and requests a challenge signature σ^* on m^*. The challenge signature σ^* is generated based on the outcome of a hidden coin toss b. If $b = 1$, then σ^* is generated as usual using the signing oracle, otherwise σ^* is chosen uniformly at random from the signature space S.
4. D performs some signing and confirmation/disavowal queries again with the restriction that no signing query on m^* is allowed, and no confirmation/disavowal query on the challenge message-signature pair (m^*, σ^*) is allowed.
5. At the end of this attack game, D outputs a guess b'.

The distinguisher D wins the game if $b' = b$. D's advantage in this game is defined to be $Adv(D) = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 2. *An undeniable signature scheme is said to have the property of invisibility under adaptive chosen message attack if no PPT distinguisher D has a non-negligible advantage in the above game.*

The difference between the above definition and the one in [6] is such that in the latter the distinguisher D outputs two messages m_0 and m_1 and the challenge signature σ^* is generated for m_b where b is the hidden bit.

3.3 Impersonation

As noted earlier, the third (and new) security notion of undeniable signatures is the security against impersonation attack. We consider the following game.

1. Let pk be the input to an impersonator I.
2. The impersonator I enters the learning phase where it performs a series of queries: signing queries and confirmation/disavowal queries as in Section 3.1 and Section 3.2. At the end of this phase, I outputs a tuple (m^*, σ^*, μ) which consists of a message-signature pair and a bit μ (where $\mu = 1$ indicates valid and $\mu = 0$ indicates invalid).
3. In the impersonation phase, if $\mu = 1$, then the impersonator I executes the confirmation protocol with a verifier on input (m^*, σ^*). If $\mu = 0$, I executes the disavowal protocol with a verifier on input (m^*, σ^*).

The impersonator I wins the game if it can convince the verifier that (m^*, σ^*) is either valid or invalid (depending on the bit μ it outputs earlier). I's advantage in this game is defined to be $Adv(I) = \Pr[I \text{wins}]$.

Definition 3. *An undeniable signature scheme is said to be secure against impersonation under adaptive chosen message attack if no PPT impersonator I has a non-negligible advantage in the above game.*

4 WI Protocol on DH-Tuple

In this section, we present our main idea, that is, we give the descriptions of DH-tuple witness indistinguishable (WI) protocol and non DH-tuple WI protocol.

The concept of witness indistinguishability and witness hiding was introduced by Feige and Shamir [16]. Generally speaking, a two-party protocol between a prover and a verifier, in which the prover uses one of the several secret witnesses to an NP assertion, is *witness indistinguishable* if the verifier cannot tell which witness the prover is actually using. The protocol is *witness hiding* if at the end of the protocol the verifier cannot compute any new witness which he did not know before the protocol began. The result in [16] says that if a statement has at least two independent witnesses, then any witness indistinguishable protocol for this statement is also witness hiding. WI protocols have been used to construct identification schemes [16] and blind signature schemes [29, 1].

In our proposal, the prover demonstrates the knowledge of 1-out-of-2 witnesses corresponding to a problem instance (a DH-tuple) without revealing which is known, thus it is a witness indistinguishable and witness hiding protocol.

4.1 WI Protocol for DH-Tuple

Let (g, U, V, W) be a DH-tuple, where $U = g^u, V = g^v, W = g^{uv}$. Now we observe that there are two witnesses, u and v. Then by using the technique of [13], we can construct a 3-move witness indistinguishable protocol such that the prover knows u or v of a DH-tuple.

We start from a 3-move honest verifier zero-knowledge proof system (HVZK) such that the prover knows u of a DH-tuple [12]. It is depicted in Fig. 1-(a). We can obtain a similar HVZK protocol such that the prover knows v. It is symmetry to Fig. 1-(a) and thus we omit the details.

	Prover		Verifier
	$r \xleftarrow{R} Z_q$		
	$z_1 = g^r$		
1	$z_2 = V^r$	$\xrightarrow{z_1,z_2}$	
2		$\xleftarrow{c}$	$c \xleftarrow{R} Z_q$
3	$d = r + cu \bmod q$	$\xrightarrow{d}$	
			$g^d \stackrel{?}{=} z_1 U^c$
			$V^d \stackrel{?}{=} z_2 W^c$

(a) Prover knows u of a DH-tuple

	Prover		Verifier
	$r \xleftarrow{R} Z_q$		
	$A = (V^u/W)^r$		
	$\alpha, \beta \xleftarrow{R} Z_q$		
	$z_1 = V^\alpha / W^\beta$		
1	$z_2 = g^\alpha / U^\beta$	$\xrightarrow{A,z_1,z_2}$	$A \stackrel{?}{\neq} 1$
2		$\xleftarrow{c}$	$c \xleftarrow{R} Z_q$
	$d_1 = \alpha + c(ur) \bmod q$		
3	$d_2 = \beta + cr \bmod q$	$\xrightarrow{d_1,d_2}$	
			$V^{d_1}/W^{d_2} \stackrel{?}{=} z_1 A^c$
			$g^{d_1}/U^{d_2} \stackrel{?}{=} z_2$

(b) Prover knows u of a non DH-tuple

Fig. 1. 3-move protocols

We finally present a 3-move WI protocol such that the prover knows u or v of a DH-tuple. For this protocol, we assume that the prover knows u (but not v).

1. The prover chooses $c_2, d_2 \in Z_q$ randomly. He computes $z'_1 = g^{d_2}/V^{c_2}$ and $z'_2 = U^{d_2}/W^{c_2}$.
   He also chooses $r \in Z_q$ randomly and computes $z_1 = g^r$ and $z_2 = V^r$.
   Next, he sends (z_1, z_2, z'_1, z'_2) to the verifier.
2. The verifier chooses $c \in Z_q$ randomly and sends c to the prover.
3. The prover computes $c_1 = c - c_2 \bmod q$ and $d_1 = r + c_1 u \bmod q$. He sends (c_1, c_2, d_1, d_2) to the verifier.
4. The verifier checks if $c = c_1 + c_2 \bmod q$ and

$$g^{d_1} = z_1 U^{c_1}, \; V^{d_1} = z_2 W^{c_1};$$
$$g^{d_2} = z'_1 V^{c_2}, \; U^{d_2} = z'_2 W^{c_2}.$$

4.2 WI Protocol for Non DH-Tuple

Suppose that (g, U, V, W) is not a DH-tuple, where $U = g^u, V = g^v, W = g^w$ and $w \neq uv \bmod q$. Then similarly to Section 4.1, we can construct a 3-move WI protocol such that the prover knows u or v of a non DH-tuple.

We start from a 3-move HVZK protocol such that the prover knows u of a non DH-tuple, as proposed in [7]. The protocols is as illustrated in Fig. 1-(b). Similarly, we can obtain a 3-move HVZK protocol such that the prover knows v. It is the symmetric counterpart of Fig. 1-(b).

We finally present a 3-move WI protocol such that the prover knows u or v of a non DH-tuple. For this protocol, we assume that the prover knows u (but not v).

1. The prover chooses $c_2, d_1', d_2' \in Z_q$ randomly and $A' \in G$ such that $A' \neq 1$ randomly. He computes $z_1' = U^{d_1'}/(W^{d_2'} A'^{c_2})$ and $z_2' = g^{d_1'}/V^{d_2'}$.
   He also chooses $r \in Z_q$ randomly and computes $A = (V^u/W)^r$. Next, he chooses $\alpha, \beta \in Z_q$ randomly and computes $z_1 = V^\alpha/W^\beta$ and $z_2 = g^\alpha/U^\beta$. Finally, he sends $(A, A', z_1, z_2, z_1', z_2')$ to the verifier.
2. The verifier first checks if $A \neq 1$ and $A' \neq 1$. Next, he chooses $c \in Z_q$ randomly and sends c to the prover.
3. The prover computes $c_1 = c - c_2 \bmod q$, and $d_1 = \alpha + c_1(ur) \bmod q$ and $d_2 = \beta + c_1 r \bmod q$. He sends $(c_1, c_2, d_1, d_2, d_1', d_2')$ to the verifier.
4. The verifier checks if $c = c_1 + c_2 \bmod q$ and

$$V^{d_1}/W^{d_2} = z_1 A^{c_1}, \ g^{d_1}/U^{d_2} = z_2;$$
$$U^{d_1'}/W^{d_2'} = z_1' A'^{c_2}, \ g^{d_1'}/V^{d_2'} = z_2'.$$

5 Proposed 3-Move Undeniable Signature Scheme

In this section, we show a *3-move* undeniable signature scheme which is secure against active and concurrent attacks. Our scheme builds on the FDH variant of Chaum's scheme which is described earlier, by incorporating the idea from the previous section.

Since the core of this paper is to propose a 3-move undeniable signature scheme which is secure against active and concurrent attacks, we consider only security against these two kinds of attacks. Nevertheless, a scheme which is secure against active/concurrent attack will definitely secure against passive attack too.

5.1 Scheme

The key generation algorithm and the signing algorithm are the same as those of the FDH variant of Chaum's undeniable signature scheme.

(Confirmation protocol) By using the 3-move WI protocol of Section 4.1, the signer proves that $(g, y, H(m), \sigma)$ is a DH-tuple, where (m, σ) is a valid message-signature pair.

(Disavowal protocol) By using the 3-move WI protocol of Section 4.2, the signer proves that $(g, y, H(m), \sigma)$ is not a DH-tuple, where (m, σ) is not a valid message-signature pair.

5.2 Security

We show that the existential unforgeability of our proposed scheme against active and concurrent attacks is equivalent to the CDH problem in the random oracle model. Similarly, we prove that our scheme is invisible under the DDH assumption and the impersonation is equivalent to the DLOG problem.

Theorem 1. *The existential unforgeability of the above 3-move undeniable signature scheme against active and concurrent attacks is equivalent to the CDH problem in the random oracle model.*

Proof. Please refer to Appendix A. □

Theorem 2. *The above 3-move undeniable signature scheme is invisible against active and concurrent attacks under the DDH assumption in the random oracle model.*

Proof. Please refer to Appendix B. □

Theorem 3. *The security against impersonation under active and concurrent attacks of the above 3-move undeniable signature scheme is equivalent to the DLOG problem in the random oracle model.*

Proof. Please refer to Appendix C. □

6 Discussion

A naive approach for witness indistinguishability would be to use two public keys, where the two corresponding secret keys are two witnesses, as suggested in [16]. However, this approach does not work as shown below.

In this approach, the signer has two public keys, $y_1 = g^{x_1}$ and $y_2 = g^{x_2}$. The undeniable signature on a message m is $\sigma = (\sigma_1, \sigma_2)$, where $\sigma_1 = H(m)^{x_1}$ and $\sigma_2 = H(m)^{x_2}$. The secret key of the signer is x_1 or x_2. In the confirmation protocol, the signer proves that σ_1 is valid OR σ_2 is valid.

However, in the disavowal protocol, the signer has to prove that "σ_1 is invalid AND σ_2 is invalid" because De Morgan's law claims that $\overline{X \vee Y} = \overline{X} \wedge \overline{Y}$. Therefore, the disavowal protocol cannot be witness indistinguishable. In general, from De Morgan's law, both the confirmation protocol and the disavowal protocol cannot be witness indistinguishable simultaneously. On the other hand, we manage to circumvent this problem by our new approach.

We further remark that our proposed scheme is almost as efficient as the FDH variant of Chaum's scheme, except that the computation and communication complexity in the confirmation and disavowal protocols are almost twice the original scheme. However, we stress that this slight efficiency loss is worthwhile in achieving the security against active and concurrent attacks with only *3-move* confirmation and disavowal protocols. This is indeed a significant contribution to the literature of undeniable signatures.

7 Conclusion

We proposed the first 3-move undeniable signature scheme which is provably secure against active and concurrent attacks, by exploiting the fact that DH-tuples possess two witnesses, and also that non DH-tuples possess two witnesses. Thus, this allows us to use the concept of witness indistinguishability and witness hiding in the confirmation and disavowal protocols of the FDH variant of Chaum's scheme. The existential unforgeability of our proposed scheme against adaptive chosen message attack is equivalent to the CDH problem. The scheme satisfies the property of invisibility assuming the intractability of the DDH problem. Moreover, we also introduced another security notion which is impersonation attack. We proved that the security against impersonation of our proposed scheme is equivalent to the DLOG problem.

References

1. M. Abe and T. Okamoto. Provably secure partially blind signatures. *Advances in Cryptology — CRYPTO '00*, LNCS 1880, pp. 271–286, Springer-Verlag, 2000.
2. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks. *Advances in Cryptology — CRYPTO '02*, LNCS 2442, pp. 162–177, Springer-Verlag, 2002.
3. J. Boyar, D. Chaum, I. Damgård and T. Pedersen. Convertible undeniable signatures. *Advances in Cryptology — CRYPTO '90*, LNCS 537, pp. 189–208, Springer-Verlag, 1990.
4. C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. *Advances in Cryptology — ASIACRYPT '98*, LNCS 1514, pp. 271–285, Springer-Verlag, 1998.
5. 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
6. J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. *Advances in Cryptology — EUROCRYPT '00*, LNCS 1870, pp. 243–258, Springer-Verlag, 2000.
7. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. *Advances in Cryptology — CRYPTO '03*, LNCS 2729, pp. 126–144, Springer-Verlag, 2003.
8. D. Chaum. Zero-knowledge undeniable signatures. *Advances in Cryptology — EUROCRYPT '90*, LNCS 473, pp. 458–464, Springer-Verlag, 1990.
9. D. Chaum. Designated confirmer signatures. *Advances in Cryptology — EUROCRYPT '94*, LNCS 950, pp. 86–91, Springer-Verlag, 1995.
10. D. Chaum, E. van Heijst and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. *Advances in Cryptology — CRYPTO '91*, LNCS 576, pp. 470–484, Springer-Verlag, 1991.

11. D. Chaum and H. van Antwerpen. Undeniable signatures. *Advances in Cryptology — CRYPTO '89*, LNCS 435, pp. 212–216, Springer-Verlag, 1989.
12. T. Chaum and T. P. Pedersen. Wallet databases with observers. *Advances in Cryptology — CRYPTO '92*, LNCS 740, pp. 89–105, Springer-Verlag, 1993.
13. R. Cramer, I. Damgård and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. *Advances in Cryptology — CRYPTO '94*, LNCS 839, pp. 174–187, Springer-Verlag, 1994.
14. J. Coron. On the exact security of full domain hash. *Advances in Cryptology — CRYPTO '00*, LNCS 1880, pp. 229–235, Springer-Verlag, 2000.
15. I. Damgård and T. Pedersen. New convertible undeniable signature schemes. *Advances in Cryptology — EUROCRYPT '96*, LNCS 1070, pp. 372–386, Springer-Verlag, 1996.
16. U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. *ACM Symposium on Theory of Computing — STOC '90*, pp. 416–426, 1990.
17. L. Fortnow. The complexity of perfect zero-knowledge (extended abstract). *ACM Symposium on Theory of Computing — STOC '87*, pp. 204-209, 1987.
18. 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.
19. S. Galbraith, W. Mao and K. G. Paterson. RSA-based undeniable signatures for general moduli. *Topics in Cryptology — CT-RSA '02*, LNCS 2271, pp. 200–217, Springer Verlag, 2002.
20. R. Gennaro, H. Krawczyk and T. Rabin. RSA-based undeniable signatures. *Advances in Cryptology — CRYPTO '97*, LNCS 1294, pp. 132–149, Springer-Verlag, 1997.
21. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptative chosen-message attacks. *SIAM Journal of Computing*, vol. 17, no. 2, pp. 281–308, 1988.
22. M. Jakobsson, K. Sako and R. Impagliazzo. Designated verifier proofs and their applications. *Advances in Cryptology — EUROCRYPT '96*, LNCS 1070, pp. 143–154, Springer-Verlag, 1996.
23. B. Libert and J.-J Quisquater. Identity based undeniable signatures. *Topics in Cryptology — CT-RSA '04*, LNCS 2964, pp. 112–125, Springer-Verlag, 2004.
24. M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. *Selected Areas in Cryptography — SAC '97*, pp. 231–244, Springer-Verlag, 1997.
25. J. Monnerat and S. Vaudenay. Undeniable signatures based on characters: how to sign with one bit. *Public Key Cryptography — PKC'04*, LNCS 2947, pp. 361–396, Springer-Verlag, 2004.
26. J. Monnerat and S. Vaudenay. Generic homomorphic undeniable signatures. *Advances in Cryptology — Asiacrypt '04*, LNCS 3329, pp. 354–371, Springer-Verlag, 2004.
27. T. Okamoto. Designated confirmer signatures and public key encryption are equivalent. *Advances in Cryptology — CRYPTO '94*, LNCS 839, pp. 61–74, Springer-Verlag, 1994.
28. D. Pointcheval. Self-scrambling anonymizers. *Financial Cryptography — FC '00*, LNCS 1962, pp. 259–275, Springer-Verlag, 2000.
29. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, vol. 13, no. 3, pp. 361–396, Springer-Verlag, 2000.

A Proof of Theorem 1

Firstly, we show that if there exists an algorithm M that solves the CDH problem with advantage ϵ_M, then one can construct a forger F that can forge in the universal way with advantage ϵ_F, by running M as a subroutine. The forger F is given the public key (g, y, H) where $y = g^x$. For any message m, F computes $h = H(m)$ and gives the triple (g, y, h) as input to M. When M outputs h^x, F simply outputs the forgery as $(m, \sigma = h^x)$. It is clear that $\epsilon_F = \epsilon_M$. This completes the first half of our proof.

Secondly, we show that if there exists an existential forger F with advantage ϵ_F, then one can construct an algorithm M that can solve the CDH problem with advantage ϵ_M, by running F as a subroutine. Suppose the input to M is (g, g^x, g^z). M then starts running F by feeding F with the public key $(g, y = g^x, H)$ where H is a random oracle that will be simulated by M. M also simulates the signing oracle and the confirmation/disavowal oracle itself. Let q_S and q_H be the number of signing queries and H queries that F issues respectively. We assume that when F requests a signature on a message m_i, it has already made the corresponding H query on m_i.

When F makes a H query for a message m_i, M responds with $h_i = H(m_i) = g^{v_i}$ with probability δ and $h_i = H(m_i) = (g^z)^{v_i}$ with probability $1 - \delta$, where v_i is chosen randomly from Z_q and δ is a fixed probability which will be determined later. Suppose that F makes a signing query for a message m_i. If M has responded with $h_i = g^{v_i}$ to the H query for a message m_i, then M returns $\sigma_i = y^{v_i}$ as the valid signature (since $y^{v_i} = (g^x)^{v_i} = h_i^x = H(m_i)^x$). Otherwise, M aborts and it fails to solve the CDH problem.

Next, we consider the case when F makes a confirmation/disavowal query. Let q_v be the number of queries that F issues to the confirmation/disavowal oracle. For convenience, we consider that the final output of F is the $(q_v + 1)$th query. We say that (m_i, σ_i') is special if it is a valid message-signature pair queried by F to the confirmation/disavowal oracle such that m_i has never been queried to the signing oracle. M guesses the first special query. More precisely, M guesses the first i such that the ith query (m_i, σ_i') is special. So, at the beginning, M chooses $\mathit{Guess} \in \{1, 2, \cdots, q_v + 1\}$ randomly. There are two cases to be considered here, namely, $i < \mathit{Guess}$ and $i = \mathit{Guess}$. First suppose that $i < \mathit{Guess}$.

- If F has never made a signing query for m_i, then M returns $\mu = 0$ and runs the disavowal protocol with F.
- Otherwise, F has already made a signing query for m_i, and M answered with a valid signature σ_i with probability δ (with probability $(1 - \delta)$ M aborts). If $\sigma_i = \sigma_i'$ then M returns $\mu = 1$ and runs the confirmation protocol with F. Otherwise, M returns $\mu = 0$ and runs the disavowal protocol with F.

Notice that since M knows v_i, it can simulate the confirmation/disavowal oracle perfectly. (Recall that the execution of the confirmation/disavowal oracle is to prove whether $(g, g^x, H(m_i) = g^{v_i}, H(m_i)^x = g^{v_i x})$ is a DH-tuple or not. Since M knows one of the witnesses which is v_i, it can simulate the interactive proof perfectly.)

Now suppose that $i = Guess$. Let (m^*, σ^*) be the ith query. If F has queried m^* to the signing oracle, then M aborts. Otherwise, we assume that F has queried the H-oracle on m^* and so $m^* = m_j$ for some j. If $h_j = (g^z)^{v_j}$, then we have $\sigma^* = h_j^x = (g^{zv_j})^x$. Consequently, M outputs $g^{xz} = (\sigma^*)^{1/v_j}$ and thus it solves the CDH problem. Otherwise, M aborts and it fails to solve the CDH problem.

To complete the proof, it remains to calculate the probability that M does not abort. M guesses the first special query with probability $1/(q_v + 1)$. The probability that M answers to all the signing queries is δ^{q_S} and M outputs g^{zr} with probability $1 - \delta$. Therefore, the probability that M does not abort during the simulation is $\delta^{q_S}(1-\delta)/(q_v+1)$. This value is maximized at $\delta_{opt} = 1-1/(q_S+1)$. This shows that M's advantage ϵ_M is at least $(1/e(1+q_S))\epsilon_F/(q_v+1)$, where e is the base of the natural logarithm. This is because the value $(1-1/(q_S+1))^{q_S}$ approaches $1/e$ for large q_S. This completes our proof.

B Proof of Theorem 2

We show that if there exists an invisibility distinguisher D with advantage ϵ_D, then one can construct an DDH distinguisher D' with advantage $\epsilon_{D'}$, by running D as a subroutine. Suppose the input to D' is (g, g^x, g^z, g^t). D' then starts running D by feeding D with the public key $(g, y = g^x, H)$ where H is a random oracle that will be simulated by D'. D' also simulates the signing oracle and the confirmation/disavowal oracle itself. Let q_S and q_H be the number of signing queries and H queries that D issues respectively. We assume that when D requests a signature on a message m_i, it has already made the corresponding H query on m_i.

When D makes a H query for a message m_i, D' responds with $h_i = H(m_i) = g^{v_i}$ with probability δ and $h_i = H(m_i) = (g^z)^{v_i}$ with probability $1 - \delta$, where v_i is chosen randomly from Z_q and δ is a fixed probability which will be determined later. Suppose that D makes a signing query for a message m_i. If D' has responded with $h_i = g^{v_i}$ to the H query for a message m_i, then D' returns $\sigma_i = y^{v_i}$ as the valid signature (since $y^{v_i} = (g^x)^{v_i} = h_i^x = H(m_i)^x$). Otherwise, D' aborts and it fails to solve the DDH problem.

Eventually, D outputs a message m^*. We assume that D has queried the H-oracle on m^* and so $m^* = m_i$ for some i. If $h_i = (g^z)^{v_i}$, then D' returns the challenge signature $\sigma = (g^t)^{v_i}$. Otherwise, D' aborts and it fails to solve the DDH problem.

Next, D performs some H queries, signing queries and confirmation/disavowal queries again with the restriction that no signing queries on m^* is allowed, and no confirmation/disavowal query on the challenge message-signature pair (m^*, σ^*) is allowed.

Finally, D outputs a bit b' which it thinks is equal to the hidden bit b. More precisely, D outputs $b' = 1$ if it finds that (m^*, σ^*) is a valid message-signature pair and it outputs $b' = 0$ if it finds that σ^* is chosen uniformly at random from the signature space S.

Subsequently, D' provides the same output as D which is b'. Note that if (m^*, σ^*) is a valid message-signature pair, then (g, g^x, g^z, g^t) is a DH-tuple. This is indeed the case since $\sigma^* = h_i^x$ implies that $t = xz \bmod q$, where $\sigma^* = (g^t)^{v_i}$ and $h_i = (g^z)^{v_i}$. Otherwise (g, g^x, g^z, g^t) is not a DH-tuple. This is indeed the case since $\sigma^* \neq h_i^x$ implies that $t \neq xz \bmod q$. Therefore, if D is an invisibility distinguisher then D' is a DDH distinguisher.

Now, we show how to simulate the confirmation/disavowal oracle. If CDH problem is easy, then DDH problem is easy. Hence D' can solve the DDH problem (without using D) in this case.

Suppose that CDH problem is hard. Then D cannot forge (m_i, σ_i) with non-negligible probability because forgery is equivalent to CDH problem from Theorem 1. Now assume that D queries (m_i, σ_i') to the confirmation/disavowal oracle.

- If D has never made a signing query for m_i, then D' returns $\mu = 0$ and runs the disavowal protocol with D. This is justified because D cannot forge as mentioned above.
- Otherwise, D has already made a signing query for m_i, and D' has answered with a valid signature σ_i. If $\sigma_i = \sigma_i'$ then D' returns $\mu = 1$ and runs the confirmation protocol with D. Otherwise, D' returns $\mu = 0$ and runs the disavowal protocol with D.

(M can run the confirmation/disavowal protocol as in the proof of Theorem 1.)

To complete the proof, it remains to calculate the probability that D' does not abort. The probability that D' answers to all the signing queries is δ^{q_S} and D' succeeds in distinguishing the DDH problem with probability $1-\delta$. Therefore, the probability that D' does not abort during the simulation is $\delta^{q_S}(1-\delta)$. This value is maximized at $\delta_{opt} = 1 - 1/(q_S + 1)$. This shows that D''s advantage $\epsilon_{D'}$ is at least $(1/e(1 + q_S))\epsilon_D$, where e is the base of the natural logarithm. This is because the value $(1 - 1/(q_S + 1))^{q_S}$ approaches $1/e$ for large q_S. This completes our proof.

C Proof of Theorem 3

Firstly, we show that if there exists an algorithm M that solves the DLOG problem with advantage ϵ_M, then one can construct an impersonator I that can succeed in an impersonation by running M as a subroutine, with advantage ϵ_I. At first, the impersonator I is given the public key (g, y, H) where $y = g^x$. Since I can obtain the secret key x by feeding y to the algorithm M, it can impersonate the signer with the knowledge of x. It is clear that $\epsilon_I = \epsilon_M$. This completes the first half of our proof.

Secondly, we show that if there exists an impersonator I with advantage ϵ_I, then one can construct an algorithm M that can solve the DLOG problem with advantage ϵ_M, by running I as a subroutine. Suppose the input to M is (g, g^x). M first chooses a bit coin.

Suppose that $\mathsf{coin} = 0$. M then starts running I by feeding I with the public key $(g, y = g^x, H)$ where H is a random oracle that will be simulated by M.

M also simulates the signing oracle and the confirmation/disavowal oracle itself. We assume that when I requests a signature on a message m_i, it has already made the corresponding H query on m_i.

In the learning phase, I starts a series of queries. When I makes a H query for a message m_i, M responds with $h_i = H(m_i) = g^{v_i}$, where v_i is chosen randomly from Z_q. When I makes a signing query for a message m_i, M returns $\sigma_i = y^{v_i}$ as the valid signature (since $y^{v_i} = (g^x)^{v_i} = h_i^x = H(m_i)^x$).

Suppose that I makes a confirmation/disavowal query for a message-signature pair (m_i, σ_i'). If m_i has never been queried to the signing oracle by I, then M simulates the signing oracle as above by itself. Hence M knows a valid signature σ_i anyway. Then M returns $\mu = 1$ if $\sigma_i' = \sigma_i$ and $\mu = 0$ if $\sigma_i' \neq \sigma_i$. M also runs the confirmation or disavowal protocol accordingly, where M can run the confirmation/disavowal protocol as in the proof of Theorem 1.

At the end of this learning phase, I outputs a tuple (m^*, σ^*, μ).

Next, I enters the impersonation phase. If $\mu = 1$, then I executes the confirmation protocol with M (acting as a verifier) on input (m^*, σ^*). M runs I to obtain its commitment (z_1, z_2, z_1', z_2'), randomly selects a challenge $c \in Z_q$, and runs I to obtain its response (c_1, c_2, d_1, d_2). M next resets I to the step whereby I has sent (z_1, z_2, z_1', z_2'). M then randomly selects a fresh challenge $c' \in Z_q$, and re-runs I to obtain its response (c_1', c_2', d_1', d_2').

If both conversations are accepted and $c \neq c'$, then M can extract the DLOG of y (which is x) or the DLOG of $H(m^*)$ (which is $v = v_i$ for some v_i) as follows. Before this, remember that $c = c_1 + c_2 \bmod q$ and $c' = c_1' + c_2' \bmod q$. This implies that $c_1 \neq c_1'$ or $c_2 \neq c_2'$, otherwise $c_1 = c_2$ which contradicts the above assumption.

From the first conversation, we obtain

$$g^{d_1} = z_1 y^{c_1}, \quad H(m^*)^{d_1} = z_2(\sigma^*)^{c_1};$$
$$g^{d_2} = z_1' H(m^*)^{c_2}, \quad y^{d_2} = z_2'(\sigma^*)^{c_2}.$$

From the second conversation, we obtain

$$g^{d_1'} = z_1 y^{c_1'}, \quad H(m^*)^{d_1'} = z_2(\sigma^*)^{c_1'};$$
$$g^{d_2'} = z_1' H(m^*)^{c_2'}, \quad y^{d_2'} = z_2'(\sigma^*)^{c_2'}.$$

Then it is not difficult to see that

$$g^{d_1-d_1'} = y^{c_1-c_1'}, \quad H(m^*)^{d_1-d_1'} = (\sigma^*)^{c_1-c_1'}; \tag{1}$$
$$g^{d_2-d_2'} = H(m^*)^{c_2-c_2'}, \quad y^{d_2-d_2'} = (\sigma^*)^{c_2-c_2'}. \tag{2}$$

When $c_1 \neq c_1'$, since $y = g^x$ and $\sigma^* = H(m^*)^x$, M can extract $x = \frac{d_1-d_1'}{c_1-c_1'} \bmod q$ from (1). When $c_2 \neq c_2'$, M can extract $v = \frac{d_2-d_2'}{c_2-c_2'} \bmod q$ from (2).

On the other hand, if $\mu = 0$, then the impersonator I executes the disavowal protocol with M (acting as a verifier) on input (m^*, σ^*). M runs I to obtain its commitment $(A, A', z_1, z_2, z_1', z_2')$, randomly selects a challenge $c \in Z_q$, and runs I to obtain its response $(c_1, c_2, d_{11}, d_{12}, d'_{11}, d'_{12})$. M next resets I to the step whereby I has sent $(A, A', z_1, z_2, z'_1, z'_2)$. M then randomly selects a fresh challenge $c' \in Z_q$, and re-runs I to obtain its response $(c'_1, c'_2, d_{21}, d_{22}, d'_{21}, d'_{22})$.

Again, if both conversations are accepted and $c \neq c'$, then M can extract the DLOG of y (which is x) or the DLOG of $H(m^*)$ (which is v) as follows. With the same argument as above, since $c = c_1 + c_2 \bmod q$ and $c' = c'_1 + c'_2 \bmod q$, this implies that $c_1 \neq c'_1$ or $c_2 \neq c'_2$.

From the first conversation, we obtain

$$H(m^*)^{d_{11}}(\sigma^*)^{-d_{12}} = z_1 A^{c_1}, \quad g^{d_{11}} y^{-d_{12}} = z_2;$$
$$y^{d'_{11}}(\sigma^*)^{-d'_{12}} = z'_1 A'^{c_2}, \quad g^{d'_{11}} H(m^*)^{-d'_{12}} = z'_2.$$

From the second conversation, we obtain

$$H(m^*)^{d_{21}}(\sigma^*)^{-d_{22}} = z_1 A^{c'_1}, \quad g^{d_{21}} y^{-d_{22}} = z_2;$$
$$y^{d'_{21}}(\sigma^*)^{-d'_{22}} = z'_1 A'^{c'_2}, \quad g^{d'_{21}} H(m^*)^{-d'_{22}} = z'_2.$$

From the above equations, we would obtain

$$H(m^*)^{d_{11}-d_{21}}(\sigma^*)^{-(d_{12}-d_{22})} = A^{c_1-c'_1}, \quad g^{d_{11}-d_{21}} y^{-(d_{12}-d_{22})} = 1; \qquad (3)$$
$$y^{d'_{11}-d'_{21}}(\sigma^*)^{-(d'_{12}-d'_{22})} = A'^{c_2-c'_2}, \quad g^{d'_{11}-d'_{21}} H(m^*)^{-(d'_{12}-d'_{22})} = 1. \qquad (4)$$

When $c_1 \neq c'_1$, since $y = g^x$ and $A = (H(m^*)^x/(\sigma^*))^r$, M can extract $x = \frac{d_{11}-d_{21}}{d_{12}-d_{22}} \bmod q$ from (3). When $c_2 \neq c'_2$, since $y = g^x$ and $A' = ((g^x)^v/(\sigma^*))^r$, M can extract $v = \frac{d'_{11}-d'_{21}}{d'_{12}-d'_{22}} \bmod q$ from (4).

Finally, for both confirmation and disavowal protocols, by Reset Lemma [2], the probability that algorithm M accepts both conversations and that $c \neq c'$ is at least $(\epsilon_I - \frac{1}{q})^2$. This shows that M can extract the DLOG of y (which is x) or the DLOG of $H(m^*)$ (which is $v = v_i$ for some v_i) with probability at least $(\epsilon_I - \frac{1}{q})^2$.

Suppose that $\mathsf{coin} = 1$. In this case, M behaves as above with the modifications as follows: M chooses $\alpha \in Z_q$ randomly, and let $y = g^\alpha$, $H(m_i) = (g^x)^{v_i}$ and $\sigma_i = (g^x)^{\alpha v_i}$, where v_i is chosen randomly from Z_q. Finally, M can extract the DLOG of y (which is α) or the DLOG of $H(m^*)$ (which is xv_i for some v_i) with probability at least $(\epsilon_I - \frac{1}{q})^2$ as in the case of $\mathsf{coin} = 0$.

This means that M's advantage in extracting x is at least $\frac{1}{2}(\epsilon_I - \frac{1}{q})^2$ because I has no information on coin. This completes our proof.