International Journal of Cryptology Research 4(1): 102 - 123 (2013)

Security of NIDV Proof Systems for Certificate-Free Undeniable Signature Schemes

[1]RouzbehBehnia, [1]Swee-HuayHeng and [2]Che-Sheng Gan

[1] Faculty of Information Science and Technology, Multimedia University

[2]Faculty of Engineering and Technology, Multimedia University

E-mail: rouzbeh.behnia@mmu.edu.my, shheng@mmu.edu.my and csgan@mmu.edu.my

ABSTRACT

Undeniable signature schemes are not publicly verifiable. Therefore, in order to prove the validity/invalidity of a message-signature pair, the signer needs to provide a proof for the rightful verifier. Since the introduction of undeniable signature schemes, various proof systems with different properties and features have been introduced to be incorporated in the structure of such schemes. Among all, the non-interactive designated verifier proof generation system of Jakobsson et al. with its distinguishing properties and features has been recognized as the most practical proof system. Due to its interesting features, a variation of Jakobsson et al. proof system has been employed in all of the proposed identity-based and certificateless undeniable signature schemes. In this paper, we analyze the security of the variation of such proof system in identity-based settings and present a secure identity-based non-interactive proof system with complete set of security proofs.

Keywords:pairing-based; undeniable signature; designated verifier; identity-based; certificateless.

1. INTRODUCTION

The idea of identity-based cryptography was first mentioned by Shamir (1985) with the aim of addressing the costly issues inherited in conventional public key cryptography. In identity-based systems, the user's public key is directly computed from her publicly available information (i.e. email address, IP address, etc.) while her secret key is calculated by a fully trusted third party called the Private Key Generator (PKG) using the master secret key. Hence, the need to issue and manage signed certificates for each user's public key is completely eliminated. The fact that the implementation and maintenance of identity-based systems was much cheaper and easier than the traditional public key infrastructure, created a promising line of research. However, the first successful implementation of identity-based

systems was until the seminal work of Boneh and Franklin (2001) which was implemented using bilinear pairing over elliptic curves. After their proposal, many researchers employed the same technique (i.e. bilinear pairing over elliptic curves) to develop a large selection of identity-based schemes with a variety of properties (Choon & Hee Cheon, 2002; Hess, 2003).

In 1989, Chaum and van Antwerpen(1989) proffered a variant of digital signatures called undeniable signature. Unlike ordinary digital signatures, the public verifiability of an undeniable signature is limited and the validity/invalidity of a particular signature can only be verified with the direct help of its signer. Technically, an undeniable signature protects the signer's rights to the privacy of the signed document. The signer in an undeniable signature scheme is able to confirm or deny the validity of her signatures by engaging in either the confirmation or disavowal protocol with the verifier. For the main applications of undeniable signatures, we can name software licensing, e-voting and e-cash (Boyd & Foo, 1998; Sakurai & Miyazaki, 2000). Since its proposal, many variations of undeniable signature schemes with different levels of security and special features have been proposed in the literature (Damgård & Pedersen, 1996; Duan, 2008; Kurosawa & Heng, 2005; Ogata, Kurosawa, & Heng, 2006).

To prevent blackmailing and man in the middle attacks on undeniable signature schemes, Jakobsson, Sako and Impagliazzo(1996) introduced the concept of non-interactive designated verifier (NIDV) proof systems. Their main goal was to enable the signer of an undeniable signature to decide not only when, but also by whom her signature is being verified. Furthermore, NIDVproof systems are more efficient than the interactive ones (e.g. 3-move honest verifier zero knowledge proof system) since they minimize the interaction between the signer and the verifier to one move.

Libert and Quisquater(2004) proposed the first provably secure identity-based undeniable signature scheme. As the first provable secure undeniable signature scheme which did not require any certificates for users, their scheme as well as the proposed security model was later used as a model for developing other undeniable signature schemes, such as the convertible identity-based undeniable signature scheme of Wu, Mu, Susilo, and Huang (2008), and Duan's(2008)certificateless undeniable signature scheme. Libert and Quisquater employed the signature structure of Goh and Jarecki(2003) and developed a pairing-based variation of NIDV proof systems of Jakobsson et al. (1996) in order to generate proof transcripts on validity/invalidity of the message-signature pairs for a designated verifier.

Our Contribution

For a NIDV proof system to be secure, it has to meet three main security notions. These security notions are, namely, completeness, non-transferability and soundness. In the work proposed by Libert and Quisquater (2004), the main stress was on the security of the signature itself (i.e. they provided a comprehensive security proof to assure the unforgeability and invisibility of the signature), and they only provided a sketchy proof in order to assure the security of the proof generation systems employed in the body of the confirmation and disavowal protocols.

Development of certificate-free (i.e. identity-based and certificateless) opaque signatures (e.g. undeniable signatures, designated verifier signatures, designated confirmer signatures, etc.) is much relied on the structure of the proof generation systems that they incorporate.

Wang's (2003) attack on NIDV proof systems of Jakobsson et al. (1996) motivated Kudla and Paterson (2005) to propose the first formal definition of NIDV proof systems and define a security model for such systems for the first time.

Due to the vital rule of the pairing-based NIDV proof systems in development of certificate-free undeniable signature schemes, and inspired by the work of Kudla and Paterson (2005), we stress on the security of pairing-based NIDV proof systems. We first extend Wang's attack (2003) on the original proposal of NIDV proof system of Jakobsson et al. (1996) to the pairing-based version of such proofs developed by Libert and Quisquater (2004). The attack enables a malicious signer to violate the soundness property of such proof systems. More specifically, a malicious signer can provide a fraudulent confirmation proof transcript for an invalid signature and later, deny the validity of the signature by generating a correct disavowal proof transcript.

Moreover, we formalize the security model of NIDV proof systems in an identity-based setting and define the properties that such proof systems should satisfy. We then put forth a secure version of pairing-based NIDV proof system with a complete set of security proofs. In our security proofs, we show that if the adversary is able to win the soundness game, then the challenger which uses the adversary as its subroutine is able to solve the Bilinear Diffie-Hellman problem and compute the master secret key and therefore, compromise the whole system.

Organization of the Paper

The remainder of the paper is organized as follows. In the next section, we introduce some preliminaries and definitions which will be used throughout this paper. In Section 3, we review Libert and Quisquater's scheme in order to familiarize the reader with their scheme. In Section 4, we propose the attack on the confirmation protocol of their scheme. In Section 5, we proffer the security model of pairing-based NIDV proof systems. In Section 6, we propose a secure version of such pairing-based NIDV proof systems and present a security analysis. We conclude our paper in Section 7.

2. PRELIMINARIES

Throughout this paper, the notation $a \leftarrow_R X$ indicates that $a$is chosen randomly from the set X.

Bilinear Pairing

We let $\mathbb{G}_1$ be an additive cyclic group of prime order q with P as its generator, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order. An admissible bilinear pairing $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$is given which is to satisfy the following properties:

Bilinearity: For $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$ we have: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}and\hat{e}(aP, bQ) = \hat{e}(abP, Q)$.

Non-degeneracy: There exist P and $Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.

Computability:$\hat{e}$is efficiently computable.

Mathematical Assumptions

The *Discrete Logarithm problem* (DL) is, given P as a generator of $\mathbb{G}_1$and $aP \in \mathbb{G}_1$, to find the value of a where $a \leftarrow_R \mathbb{Z}_q$.

The*Bilinear Diffie-Hellman problem* (BDH) is, given P as a generator of $\mathbb{G}_1$ and $aP, bP, cP \in \mathbb{G}_1$for unknown $a, b, c \leftarrow_R \mathbb{Z}_q$to compute $\hat{e}(P, P)^{abc}$.

The *Decisional Bilinear Diffie-Hellman problem* (DBDH) is, given P as a generator of $\mathbb{G}_1, aP, bP, cP \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ for unknown $a, b, c \leftarrow_R \mathbb{Z}_q$, to decide whether $h = \hat{e}(P, P)^{abc}$.

$f^{-1}$is defined to be the inverse of the isomorphism where if $W = aP$ and $Q = bP$; in which P is a generator of $\mathbb{G}_1$ then $f_{(P,P)}^{-1}\,\hat{e}(P,W) = f_{(Q,P)}^{-1}\hat{e}(Q,W)$.

3. LIBERT AND QUISQUATER'S SCHEME

Libert and Quisquater(2004) proposed the first provably secure undeniable signature scheme in identity-based paradigm and showed the unforgeability and invisibility of their scheme relies on the hardness of the BDH and the DBDH assumptions respectively. Their scheme along with its security model has inspired the development of other schemes such as (Duan, 2008; Wu, et al., 2008). Following the work of Jakobsson et al. (1996), Libert and Quisquater (2004) developed a pairing-based designated verifier confirmation/disavowal protocol to prove the validity/invalidity of the signature in a non-interactive manner. In the following, we provide a quick review on Libert and Quisquater's undeniable signature scheme (consisting of 3 algorithms (setup, extract, and sign) and 2 protocols (confirmation and disavowal)). Throughout this paper, the signer and the designated verifier are represented by their key pair as (ID_S, d_{ID_S}) and$\big(ID_V, d_{ID_V}\big)$ respectively.

Setup: On inputting security parameters k and l, this algorithm generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q \geq 2^k$, a generator P of $\mathbb{G}_1$ and an admissible bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. The algorithm also chooses 4 cryptographic hash functions: $H_1: \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2: \{0,1\}^* \times \{0,1\}^l \times \{0,1\}^* \rightarrow \mathbb{G}_1andH_3, H_4: \{0,1\}^* \rightarrow \mathbb{Z}_q$. The PKG sets $s \longleftarrow_R \mathbb{Z}_q$ as its secret key and calculates $P_{pub} = sP$ as its public key. The PKG's public key and system parameters $(params)$ will be available to all the users in the system.

$$params = \big(q, \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, H_1, H_2, H_3, H_4\big)$$

Extract: Given the user's identity ID, PKG uses the master secret key s to compute the user's secret value $d_{ID} = sQ_{ID} = sH_1(ID)$ based on his/her identity ID.

Sign: Suppose the signer with identity ID_Swants to sign a message $m \in \{0,1\}^*$. She chooses a random salt $r \longleftarrow_R \{0,1\}^l$ and computes the value of $\lambda = \hat{e}\big(H_2(m, r, ID_S),\ d_{ID_S}\big)$ to form the signature $\sigma = (r,\ \lambda) = (r, \hat{e}\big(H_2(m, r, ID_S),\ d_{ID_S}\big))$.

Confirmation: In order for the signer to compute a confirmation proof transcript on the validity of a message-signature pair (m, σ) for the designated verifier with identity ID_V, she sets the values of $Q_{ID_V} = H_1(ID_V)$ and $h_2 = H_2(m, r, ID_S)$ and picks $U, R \leftarrow_R \mathbb{G}_1$ and $v \leftarrow_R \mathbb{Z}_q^*$ to compute $c = \hat{e}(P, U)\hat{e}\left(P_{pub}, Q_{ID_V}\right)^v, g_1 = \hat{e}(P, R), g_2 = \hat{e}(h_2, R), h = H_3(c, g_1, g_2)$, and $S = R + (h + v)\, d_{ID_S}$ and sends the confirmation proof transcript as (U, v, h, S).

Upon receiving the confirmation proof transcript(U, v, h, S), the designated verifier sets $h_2 = H_2(m, r, ID_S)$ and computes $c' = \hat{e}(P, U)\hat{e}\left(P_{Pub}, Q_{ID_V}\right), g_1' = \hat{e}(P, S)\hat{e}\left(P_{pub}, Q_{ID_S}\right)^{h+v}, g_2'$ d $h' = H_3(c', g_1', g_2')$ and the verifier accepts the proof if and only if $h' = h$.

In the above protocol, the signer forms the proof (U, v, h, S) in order to convince the designated verifier that $f_{(P,P)}^{-1}\, \hat{e}\left(P_{pub}, Q_{ID_S}\right) = f_{(H_2(m,r,ID_S),P)}^{-1}\, \hat{e}\left(H_2(m, r, ID_S), d_{ID_S}\right)$.

Disavowal: In order for the signer ID_S to convince the designated verifier ID_V about the invalidity of a message-signature pair(m, σ), she takes the following steps:

- Picks $\omega, v \leftarrow_R \mathbb{Z}_q^*$, $U \leftarrow_R \mathbb{G}_1$ and computes $h_2 = H_2(m, r, ID_S), C = \left(\frac{\hat{e}\left(h_2, d_{ID_S}\right)}{\lambda}\right)^{\omega}$ and $c = \hat{e}(P, U)\hat{e}\left(P_{pub}, Q_{ID_V}\right)^v$.
- She has to prove her knowledge of a pair (R, α) such that, $C = \frac{\hat{e}(h_2, R)}{\lambda^{\alpha}}$and $1 = \frac{\hat{e}(P, R)}{\hat{e}\left(P_{pub}, Q_{ID_S}\right)^{\alpha}}$.
- In order to do so, she picks $z \leftarrow_R \mathbb{Z}_q^*$ and $V \leftarrow_R \mathbb{G}_1$ and computes $\rho_1 = \hat{e}(h_2, V)\lambda^{-z}, \rho_2 = \hat{e}(P, V)\hat{e}\left(P_{Pub}, Q_{ID_S}\right)^{-z}$, $h = H_4(\mathrm{C}, \mathrm{c}\rho_1, \rho_2)$, $S = V + (h + v)R$, and $s = z + (h + v)\alpha$and sends the proof as (C, U, v, h, S, s).
- Upon receiving the disavowal proof transcript (C, U, v, h, S, s), the designated verifier ID_V first checks if $C = 1$, he will reject the proof. Otherwise, he forms $h_2 = H_2(m, r, ID_S), c' = \hat{e}(P, U)\hat{e}\left(P_{Pub}, Q_{ID_V}\right)\rho_1' = \hat{e}(H_2(m, r, ID_S), S)\lambda^{-s}C^{-(h+v)}, \rho_2' = \hat{e}(P, S)y^{-s}$, and $h' =$

$H_4(C, c', \rho_1', \rho_2')$ and accepts the proof if and only if $h = h'$.

In the above protocol, the signer forms the proof (C, U, v, h, S, s) in order to convince the designated verifier that

$$f_{(P,P)}^{-1}\,\hat{e}\left(P_{pub}, Q_{ID_S}\right) \neq f_{(H_2(m,r,ID_S),P)}^{-1}\hat{e}\left(H_2(m, r, ID_S), d_{ID_S}\right).$$

Both of the above protocols are non-transferable. More precisely, the designated verifier is able to create confirmation/disavowal proof transcripts which are indistinguishable from those generated by the signer by computing the commitment collisions (U', v'), using his secret key d_{ID_V}.

4. THE ATTACK

As previously mentioned, the confirmation and disavowal protocols of Libert and Quisquater's scheme is a variation of the method proposed by Jakobsson et al. (1996). Due to its advantages and efficiencies, the confirmation and disavowal protocols of their scheme were slightly modified and employed in other pairing-based undeniable signature schemes (Duan, 2008; Wu, et al., 2008).

Wang (2003) proposed an attack on Jakobsson et al.'s NIDV proof system. Inspired by Wang's work, we show that the pairing-based version of such proof systems developed by Libert and Quisquater is not secure. Our attack enables a dishonest signer to provide a valid confirmation proof for a falsely generated signature and later, deny the validity of the signature in court by generating a correct disavowal protocol proof transcript.

In the confirmation proof transcript of Libert and Quisquater's scheme, the signer proves to the designated verifier that equality $f_{(P,P)}^{-1}\,\hat{e}\left(P_{pub}, Q_{ID_S}\right) = f_{(H_2(m,r,ID_S),P)}^{-1}\,\lambda$ holds. However, the flaw that the signer is not obligated to prove the equality of the values of R in g_1 and g_2 leads in mounting the following attack.

This attack for instance, can be mounted by a malicious e-bidder in an e-auction website. The malicious e-bidder initiates the attack by presenting a falsely generated undeniable signature on a message stating his bid on a specific item to the seller and provides a fraudulent confirmation proof transcript along. After the seller receives and checks the validity of the

malicious e-bidder's signature along with the confirmation protocol transcript, he will assign the highest bid to the e-bidder. Later, the malicious e-bidder is able to deny his bid by running the disavowal protocol on the falsely generated signature, and therefore, violate the rules and regulations of the e-auction website.

The malicious signer forms the attack by generating the fraudulent proof and then the false signature as follows.

Fraudulent Proof Generation: In order for the malicious signer ID_{S^*} to form a fraudulent proof for the target designated verifier ID_{V^*}, she chooses $U \longleftarrow_R \mathbb{G}_1$ and $v, t, t' \longleftarrow_R \mathbb{Z}_q^*$randomly and computes $Q_{ID_{V^*}} = H_1(ID_{V^*})$. She then forms the fraudulent proof as follows:

$$\psi = (U, v, h, S)\begin{cases} R = tP, \quad R' = t'P \\ c = \hat{e}(P, U)\hat{e}\left(P_{pub}, Q_{ID_{V^*}}\right)^v \\ g_1 = \hat{e}(P, R), \quad g_2^* = \hat{e}(H_2(m, r, ID_{S^*}), R') \\ h = H_3(c, g_1, g_2^*) \\ S = R + (h + v)d_{ID_{S^*}} \end{cases}$$

Signature Generation: After forming the fraudulent proof, the dishonest signer sets the value of $h_2 = H_2(m, r, ID_{S^*})$ and computes the values of λ^* as follows:

$$\lambda^* = (\hat{e}(h_2, P)^{t'-t}\hat{e}\left(h_2, d_{ID_{S^*}}\right)^{-(h+v)})^{-(h+v)}$$

She then forms the non-standard signature as $\sigma^* = (r, \lambda^*)$ and submits $(\sigma^*, \psi)$to the designated verifier.

Upon receiving the invalid signature proof tuple (σ^*, ψ), the designated verifier ID_{V^*} forms $h_2 = H_2(m, r, ID_{S^*})$ and computes the following:

$$c' = \hat{e}(P, U)\hat{e}\left(P_{pub}, Q_{ID_{V^*}}\right)^v$$
$$g_1' = \hat{e}(P, S)\hat{e}\left(P_{pub}, Q_{ID_{S^*}}\right)^{h+v}$$
$$g_2^{*'} = \hat{e}(h_2, S)\lambda^{*\,h+v}$$

The verifier will accept the proof if and only if $h' = h$, where $h' = H_3(c', g_1', g_2^{*'})$.

*Remark 1.*The designated verifier will be convinced about the veracity of the falsely formed confirmation proof if it passes the consistency check above.

The values of c' and g_1' are calculated identical to the original transcript of the proof (as in the confirmation protocol in the previoussection) by the verifier. For the value of $g_2^{*'}$, we know that,$g_2^* = \hat{e}(h_2, R')$, and $\hat{e}(h_2, S) = \hat{e}\left(h_2, R + (h+v)\, d_{ID_{S^*}}\right)$, so we expand as follows.

$$
\begin{aligned}
g_2^{*'} &= \hat{e}\,(h_2, S)\lambda^{*\,h+v} \\
&= \hat{e}\,(h_2, S)((\hat{e}(h_2, P)^{t'-t}\hat{e}\left(h_2, d_{ID_{S^*}}\right)^{-(h+v)})^{-(h+v)})^{(h+v)} \\
&= \hat{e}\left(h_2, d_{ID_{S^*}}\right)^{(h+v)}\hat{e}\left(h_2, d_{ID_{S^*}}\right)^{-(h+v)}\hat{e}(h_2, P)^{t}\hat{e}(h_2, P)^{-t} \\
&\quad \hat{e}(h_2, P)^{t'} \\
&= \hat{e}(h_2, P)^{t'} = g_2^*
\end{aligned}
$$

Remark 2. Based on the disavowal protocol depicted in the previous section, the signer is able to generate the disavowal proof transcript if for $C = \left(\frac{\hat{e}\left(h_2, d_{ID_{S^*}}\right)}{\lambda}\right)^{\omega}$ then $C \neq 1$. We can see that the falsely generated signature will pass such test since $\left(\frac{\hat{e}\left(h_2, d_{ID_{S^*}}\right)}{(\hat{e}(h_2,P)^{t'-t}\hat{e}\left(h_2, d_{ID_{S^*}}\right)^{-(h+v)})^{-(h+v)}}\right)^{\omega} \neq 1$.

5. SECURITY OF PAIRING-BASED NIDV PROOFS

Due to its efficiency and added security features, and after its introduction by Libert and Quisquater(2004), pairing-based NIDV proof systems have been employed in the structure of other identity-based (Wu et al., 2008) and certificateless(Duan, 2008) undeniable signature schemes. Unfortunately, none of the proposed schemes which take advantage of employing such proof systems provided a security model in order to assure their security.

Following the work of Kudla and Paterson (2005), we treat such NIDV proof system separately and provide a comprehensive set of security proofs in order to relate their security to the difficulty of a well-known complexity assumption. The main reason is the fact that in undeniable signature schemes, many proofs maybe generated for different verifiers in order to prove the validity/invalidity of a single signature. In comparison with Kudla and Paterson (2005)security proofs, we show that if an adversary is able to win the soundness game, then the challenger that runs the adversary as its subroutine is able to solve the Discrete Logarithm problem

in order to compute the system master secret key and compromise the whole system.

We consider apairing-based NIDV proof system to be secure if it meets the following security properties.

Completeness

We say that a proof system is complete if the designated verifier accepts the proof if and only if it is generated by the true signer (using her secret key) on the validity/invalidity of a given message-signature pair.

Non-transferability

Non-transferability is a weaker assumption than zero-knowledge-ness. A pairing-based NIDV proof system is non-transferable if there exists a polynomial time algorithm that on input of a tuple (σ, d_{ID_V}, ID_S), where d_{ID_V} is the secret key of the designated verifier, ID_S is the public key of the signer, but σ is not essentially a valid signature, produces a proof transcript which is indistinguishable from the one generated by using the secret key corresponding to ID_S.

Soundness

A pairing-based NIDV proof system is sound if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage in the following game:

1. The adversary is provided with the system wide public parameters $params$ and the public key of the PKG (i.e. P_{pub}).
2. The adversary $\mathcal{A}$ performs a series of queries:
   I. *Signature queries:* $\mathcal{A}$ produces a message m and an identity ID and queries the signature generation oracle in order to receive asignature on m which is generated using the secret key corresponding to ID.
   II. *Key extraction queries:* $\mathcal{A}$ produces an identity ID and queries the key extraction oracle for the secret key d_{ID} corresponding to ID.
   III. *Confirmation/disavowal proof transcript queries:*
      i. Adversary $\mathcal{A}$ produces a message-signature pair (m, σ), a signer's identity ID_S, and a designated verifier's identity ID_V. $\mathcal{A}$ queries the signer's proof generation oracle to produce a NIDV proof transcript ψ on the validity/invalidity of the tuple $(m, \sigma, ID_S)$for the designated verifier with identity ID_V.
      ii. Adversary $\mathcal{A}$ produces a message-signature pair (m, σ), a signer's identity ID_S, and a designated verifier's identity ID_V.

$\mathcal{A}$queries the designated verifier's proof simulation oracle to generate a NIDV proof transcript ψ on the validity/invalidity of the tuple(m, σ, ID_S).

3. At the end of the game, $\mathcal{A}$ outputs$(ID_S^*, ID_V^*, \sigma^*, \psi^*)$,where ID_V^* has not been queried to the key extraction oracle, ψ^* can successfully pass the verification steps to prove the validity/invalidity of the tuple (m, σ^*, ID_S^*) , and ψ^* was not the output of the confirmation/disavowal proof generation oracle. Adversary $\mathcal{A}$ wins if either:

   i. ID_S^* has not been queried to the key extraction oracle, or
   ii. $\sigma^*$is invalid and ψ^* is a confirmation proof transcript, or σ^* is valid and ψ^* is a disavowal proof transcript.

We say that a pairing-based NIDV proof system is sound if no polynomially bounded adversary is able to win the above game with a non-negligible probability.

The above game assures that a signer is not able to cheat in proving the validity/invalidity of a message-signature pair(m, σ). Put differently, if an uncorrupted designated verifier (where his secret key was never queried from the key extraction oracle) receives a valid pairing-based non-interactive proof; it was created using the signer's secret key d_{ID_S}. This property is very vital for undeniable signature schemes as only the signer should have the ability to generate such a proof for a particular designated verifier.

6. SECURE PAIRING-BASED NIDV PROOF SYSTEM

The easy fix proposed by Wang (2003) consisted of adding two more values (i.e. the signature and the signer's public key) to the hash function of the confirmation protocol to prevent a malicious signer to form the fraudulent confirmation proof before generating the signature.

Here, we propose a secure version of the pairing-based NIDV proof system employed in Libert and Quisquater's scheme (2004). Since the public key of the user in identity-based setting is actually the identity of the user and it has already been incorporated in the signature structure; therefore, it is not required to add the user's public key to the confirmation or disavowal protocol hash functions.

We assume that the Setup, Extract, and Sign algorithms take place identical to Libert and Quisquater's scheme. Below are the secure versions of pairing-based NIDV proof systems.

Confirmation Protocol: In order to form the confirmation protocol proof transcript to prove the validity of a message-signature pair (m, σ), the signer ID_S picks $v, u, r \leftarrow_R \mathbb{Z}_q^*$ and computes the following:

$$\begin{aligned} c &= \hat{e}(P, uP)\hat{e}\left(P_{pub}, Q_{ID_V}\right)^v \\ g_1 &= \hat{e}(P, rP) \\ g_2 &= \hat{e}(H_2(m, r, ID_S), rP) \\ h &= H_3(c, g_1, g_2, \sigma) \\ S &= rP + (h + v)d_{ID_S} \end{aligned}$$

The proof $\psi = (u, v, h, S)$ is formed, and will be checked by the verifier similarly as shown in the original scheme.

Disavowal Protocol: In order to form the disavowal protocol proof transcript, upon receiving an invalid message-signature pair $(m, \sigma^* = (r, \lambda^*))$, the signer ID_S picks $v, u \leftarrow_R \mathbb{Z}_q^*$ and generates the disavowal protocol proof transcript as follows:

- Computes $c = \hat{e}(P, uP)\hat{e}\left(P_{pub}, Q_{ID_V}\right)^v$ and picks $\omega \leftarrow_R \mathbb{Z}_q^*$ in order to compute the value of $C = \left(\frac{\hat{e}\left(H_2(m,r,ID_S), d_{ID_S}\right)}{\lambda^*}\right)^{\omega}$.
- She has to prove her knowledge of a pair $(L, \alpha) \in \mathbb{G}_1 \times \mathbb{Z}_q^*$ in a zero-knowledge way, such that $C = \frac{\hat{e}(H_2(m,r,ID_S),L)}{\lambda^{*\alpha}}$ and $\frac{\hat{e}(P,L)}{\hat{e}\left(P_{pub},Q_{ID_S}\right)^{\alpha}} = 1$. Therefore, she picks $b,\ k \leftarrow_R \mathbb{Z}_q^*$ and computes the following:

$$\begin{aligned} \rho_1 &= \hat{e}(H_2(m, r, ID_S), bP)\lambda^{*-k} \\ \rho_2 &= \hat{e}(P, bP)\hat{e}\left(P_{pub}, Q_{ID_S}\right)^{-k} \\ h &= H_4(\mathrm{C}, \mathrm{c}, \rho_1, \rho_2, \sigma) \\ S &= bP + (h + v)L \\ s &= k + (h + v)\alpha \end{aligned}$$

The proof $\psi = (C, u, v, h, S, s)$ is formed, and will be checked by the verifier similarly as shown in the original scheme.

An Efficient Construction

The confirmation/disavowal protocol can be made more efficient by taking advantage of bilinear property of pairing. Employing this method, we can save one pairing computation in both the generation and verification of the confirmation/disavowal proof transcript.

In the confirmation protocol, instead of computing g_1 and g_2, the signer computes the value of $g_S = \hat{e}(H_2(m, r, ID_S) + P, rP)$; and the verifier computes $g'_S = \hat{e}(H_2(m, r, ID_S), S)\lambda^{(h+v)}\hat{e}(P, S)\hat{e}\left(P_{Pub}, Q_{ID_S}\right)^{(h+v)}$. The same method can also be applied in computing the disavowal protocol proof transcript.

Security Analysis

Theorem 1: The above pairing-based NIDV proof systems are complete. Proving that the proof that is generated by a true signer is always accepted by the verifier is trivial and therefore, the proof of Theorem 1 is omitted.

Theorem 2: The above pairing-based NIDV proof systems are non-transferable.

*Proof.*We define the designated verifier's ID_V simulation algorithm as follows.

On input $(ID_S, d_{ID_V}, m, \sigma)$, it picks $u^*, s, y \leftarrow_R \mathbb{Z}_q$ and calculates the following:

$$
\begin{aligned}
c &= \hat{e}(P, u^*P) \\
g_1 &= \hat{e}(P, sP)\hat{e}\left(P_{pub}, Q_{ID_S}\right)^y \\
g_2 &= \hat{e}(H_2(m, r, ID_S), sP)\lambda^y \\
h &= H_3(c, g_1, g_2, \sigma) \\
v &= h - y \\
U &= u^*P - vd_{ID_V} \\
\psi &= (U, v, h, s)
\end{aligned}
$$

It is easy to check that the simulated proof transcript $\psi = (U, v, h, s)$ will be accepted by the verifier, and it is indistinguishable from any proof transcript $\psi^* = (U^*, v^*, h^*, s^*)$ generated by the signer (using her secret key d_{ID}). □

Theorem 3: In the random oracle model, if there exists an adversary $\mathcal{A}$ that is able to succeed in the soundness game after performing q_{H} queries to random oracles H_i for $i \in \{1,2,3\}$, q_{S} signature queries, q_{E} key extraction queries, and q_{CD} confirmation and disavowal queries with advantage ϵ; then, there exists an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ as its subroutine and solves the Discrete Logarithm problem in $\mathbb{G}_1$ in order to compute the master secret key with advantage ϵ' where:

$$\epsilon' \geq \frac{\epsilon - \left(\left(q_{CD}\left(q_{CD} + q_{H_3}\right)/2^k\right)^2 + 8/2^k\right)}{q_{H_3}}$$

Proof. We show that if there is an adversary $\mathcal{A}$ which is able to win the soundness game with probability ϵ; then, one can build an algorithm $\mathcal{B}$ that can solve the Discrete Logarithm problem in $\mathbb{G}_1$ with probability ϵ'. $\mathcal{B}$ runs $\mathcal{A}$ as its subroutine and plays the role of $\mathcal{A}$'s challenger, it starts by providing $\mathcal{A}$ with public parameters $\left(q, \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, H_i (i = 1, 2, 3)\right)$ such that $P_{pub} = cP$ (where c is unknown to $\mathcal{B}$) and H_i is a random oracle.

$\mathcal{A}$ performs a series of queries as mentioned in the soundness game, $\mathcal{B}$ responds to these queries by keeping lists $\mathcal{K}_i (i = 1, 2, 3)$. $\mathcal{A}$ is able to perform q_H hash function queries, q_S signature queries, q_E key extraction queries, and q_{CD} confirmation and disavowal queries. We presuppose that every key extraction query on an identity ID is preceded by a query on the random oracle $H_1(ID)$. Queries made by $\mathcal{A}$ are handled as follows:

Query on H_1: $\mathcal{B}$ simulates H_1 oracle by keeping a list $\mathcal{K}_1$ of tuples (ID_i, μ_i). When H_1 is queried with $ID_i \in \{1, 0\}^*$, $\mathcal{B}$ responds as follows:

- If ID_i had been queried to H_1 before, then it is already in the list $\mathcal{K}_1$. Therefore, $\mathcal{B}$ retrieves (ID_i, μ_i) and returns $\mu_i P$ to $\mathcal{A}$.
- Otherwise, $\mathcal{B}$ picks $\mu_i \longleftarrow_R \mathbb{Z}_q$, returns μP to $\mathcal{A}$, and adds (ID_i, μ_i) to $\mathcal{K}_1$.

Query on H_2: $\mathcal{B}$ simulates H_2 oracle in the same way as H_1 by keeping a list of tuples (m_i, r_i, ID_i, d_i) in $\mathcal{K}_2$; where $\mathcal{B}$ picks $d \longleftarrow_R \mathbb{Z}_q$, returns $H_2(m_i, r_i, ID_i) = dP$ to $\mathcal{A}$, and stores (m_i, r_i, ID_i, d) in $\mathcal{K}_2$.

Signature queries: $\mathcal{A}$ is allowed to query for a signature on a message $m_i \in \{1, 0\}^*$ and an identity ID_i. $\mathcal{B}$ responds to such signature queries as follows:

- $\mathcal{B}$ picks $r \leftarrow_R \{1,0\}^l$ and checks $\mathcal{K}_2$, if (m_i, r, ID_i) already exists in $\mathcal{K}_2$; then, $\mathcal{B}$ picks a new r until it finds a tuple (m_i, r, ID_i) for which no record exists in $\mathcal{K}_2$.
- Then, $\mathcal{B}$ simulates $H_2(m_i, r, ID_i) = dP$, and returns the signature $\sigma = (r, \hat{e}(dP_{pub}, Q_{ID_i}))$ to $\mathcal{A}$.

Key extraction queries: $\mathcal{A}$is able to query for the secret key associate with an identity ID_i of his choice. $\mathcal{B}$responds to such queries as follows:

- fID_i had been queried to H_1 before, then it is already in the list $\mathcal{K}_1$. Therefore, $\mathcal{B}$ retrieves (ID_i, μ_i) and returns $\mu_i P_{pub}$ as the corresponding secret key of ID_i to $\mathcal{A}$.
- Otherwise, $\mathcal{B}$ picks $\mu \leftarrow_R \mathbb{Z}_q$, returns μP_{pub} as the secret key of ID_i, and adds (ID_i, μ) to $\mathcal{K}_1$.

Proof transcript queries: As mentioned in the soundness game, $\mathcal{A}$ is able to query the challenger for confirmation proof transcript on validity of a given message-signature pair (m, σ). In designated verifier proofs, the designated verifier is able to use his secret key in order to simulate proof transcripts which are indistinguishable from those generated by the original signer. Hence, in the game between the adversary and the challenger, we assume that the adversary is also able to query the designated verifier simulation oracle for the simulated version of the proof transcript. $\mathcal{A}$'s queries to the signer'sID_Sproof generation oracle and the designated verifier's ID_V proof simulation oracle are handled as follow.

$\mathcal{A}$ can query for confirmation proof transcript on a tuple (m, σ, ID_V) from the signer's ID_S proof generation oracle, where ID_V is the identity of the designated verifier for which the proof is to be generated. $\mathcal{B}$first parses σ to (λ, r) and runs the signature oracle on (m, r, ID_S) to get σ'; if $\sigma = \sigma'$, $\mathcal{B}$ picks $u, v, s, h \leftarrow_R \mathbb{Z}_q$and generates the confirmation proof transcript as follows:

$$c = \hat{e}(P,P)^u \hat{e}\left(P_{pub}, Q_{ID_V}\right)^v = \hat{e}(P,P)^u \hat{e}(cP, \mu_V P)^v$$
$$g_1 = \hat{e}(P,P)^s \hat{e}\left(P_{pub}, Q_{ID_S}\right)^{h+v} = \hat{e}(P,P)^s \hat{e}(cP, \mu_S P)^{h+v}$$
$$g_2 = \hat{e}\,(H_2(m, r, ID_S), P)^s \hat{e}\left(dP_{pub}, Q_{ID_S}\right)^{h+v}$$
$$= \hat{e}\,(dP, P)^s \hat{e}(dcP, \mu_S P)^{h+v}$$

If the oracle H_3 was previously queried on $Q = (c, g_1, g_2, \sigma)$, $\mathcal{B}$ stops and outputs "failure"; Otherwise, $\mathcal{B}$ adds the tuple (Q, h) to $\mathcal{K}_3$, and returns the proof transcript $\psi = (u, v, h, sP)$ to $\mathcal{A}$.

- $\mathcal{A}$ can query for confirmation proof transcript on tuple (m, σ, ID_S) from the designated verifier's ID_V proof simulation oracle, where ID_S is the identity of the signer. To compute the simulated version of the confirmation proof transcript, $\mathcal{B}$ picks $u', v', s', h' \leftarrow_R \mathbb{Z}_q$ and computes$c', g_1'$and g_2' same as above.

Identical to the above case, If the oracle H_3 was previously queried on $Q' = (c', g_1', g_2', \sigma)$, $\mathcal{B}$ stops and outputs "failure"; Otherwise, $\mathcal{B}$ adds the tuple (Q', h') to $\mathcal{K}_3$ and returns the proof transcript $\psi' = (u', v', h', s'P)$ to $\mathcal{A}$.

Output: Finally, $\mathcal{A}$ outputs a valid tuple $(ID_S^*, ID_V^*, m^*, \sigma^*, \psi^*)$, where ID_V^* has never been queried to the key extraction oracle, and the proof $\psi^* = (u^*, v^*, h^*, s^*P)$ was never outputted from the proof generation oracle (i.e. either the signer's proof generation oracle, or the designated verifier's proof simulation oracle). Then, $\mathcal{A}$ wins if either:

I. ID_S^* has never been queried to the key extraction oracle, or
II. (m^*, σ^*, ID_S^*) is an invalid tuple.

Case I: Suppose that $ID_S^*$was never queried to the key extraction oracle.

If h^* was never outputted from either the signer's proof generation oracle, or the designated verifier's proof simulation oracle, then based on forking lemma (Pointcheval & Stern, 1996), $\mathcal{B}$ can repeat the simulation process until $\mathcal{A}$ outputs another tuple $(ID_S^*, ID_V, m^*, \sigma^*, \psi)$ with a non-negligible probability, where $\psi = (u, v, h, s)$ and $h \neq h^*$. Therefore, will get the following equations:

$$\hat{e}(P,P)^{u^*}\hat{e}(cP, {\mu_V}^*P)^{v^*} = \hat{e}(P,P)^{u}\hat{e}(cP, \mu_V P)^{v} \quad (1.a)$$
$$\hat{e}(P,P)^{s^*}\hat{e}(cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(P,P)^{s}\hat{e}(cP, {\mu_S}^*P)^{h+v} \quad (1.b)$$
$$\hat{e}(d^*P,P)^{s^*}\hat{e}(d^*cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(dP,P)^{s}\hat{e}(dcP, {\mu_S}^*P)^{h+v} \quad (1.c)$$

If $ID_V^* \neq ID_V$ or $v^* \neq v$ and both ID_V^* and ID_Vhad been queried to H_1oracle before, then $\mathcal{B}$ can solve (1.a) for discrete logarithm of $P_{pub} = cP$.

The probability that ID_V^* and ID_V have never been queried to H_1 oracle is smaller than $2/2^k$. On the other hand, if $ID_V^* = ID_V$ and

$v^* = v$ then definitely $u^* = u$. Given $h \neq h^*$, we have $h^* + v^* \neq h + v$ therefore, $\mathcal{B}$ can solve (1.b) for the discrete logarithm of P_{pub}. In this case, $\mathcal{B}$ fails if ID_V^* was never queried to H_1 oracle. The probability that ID_S^* was never queried to H_1 is smaller than $1/2^k$.

If h^* was an output from the proof transcript generation oracle, then it was generated either by the signer's proof generation oracle, or the designated verifier's proof simulation oracle.

If h^* was output by some previous query to signer's ID_S^* proof generation oracle on (m^*, σ^*, ID_V) which generated the proof $\psi = (u, v, h^*, s)$, then since h and h^* were outputs from H_3, we assume that with overwhelming probability that the inputs to $H_3(c, g_1, g_2, m, \sigma, ID_S)$ were identical.

$$\hat{e}(P,P)^{u^*}\hat{e}(cP, {\mu_V}^*P)^{v^*} = \hat{e}(P,P)^{u}\hat{e}(cP, \mu_V P)^{v} \quad (1.d)$$
$$\hat{e}(P,P)^{s^*}\hat{e}(cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(P,P)^{s}\hat{e}(cP, {\mu_S}^*P)^{h+v} (1.e)$$
$$\hat{e}\,(d^*P,P)^{s^*}\hat{e}(d^*cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}\,(dP,P)^{s}\hat{e}(dcP, {\mu_S}^*P)^{h+v} (1.f)$$

Same as in equation (1.a), if $ID_V^* \neq ID_V$ or $v^* \neq v$ and both ID_V^* and ID_V had been queried to H_1 oracle before, then $\mathcal{B}$ can solve (1.d) for the discrete logarithm of P_{pub}. The probability that ID_V^* and ID_V have never been queried to H_1 oracle is smaller than $2/2^k$. If $ID_V^* = ID_V$ and $v^* = v$ then definitely $u^* = u$, so $h^* + v^* = h + v$, which gives us $s = s^*$.

Having $s = s^*$ implies that $\psi^* = (u^*, v^*, h^*, s^*)$ was an output of the signer's proof generation oracle, contradicting our assumption.

If h^* was output by some previous query to the designated verifier's ID_V^* proof simulation oracle on (m^*, σ^*, ID_S) which produced the proof $\psi = (u, v, h^*, s)$, then again, with overwhelming probability, we will obtain the following equations:

$$\hat{e}(P,P)^{u^*}\hat{e}(cP, {\mu_V}^*P)^{v^*} = \hat{e}(P,P)^{u}\hat{e}(cP, {\mu_V}^*P)^{v} \quad (1.g)$$
$$\hat{e}(P,P)^{s^*}\hat{e}(cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(P,P)^{s}\hat{e}(cP, \mu_S P)^{h+v} \quad (1.h)$$
$$\hat{e}\,(d^*P,P)^{s^*}\hat{e}(d^*cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}\,(dP,P)^{s}\hat{e}(dcP, {\mu_S}^*P)^{h+v} \quad (1.i)$$

Now if $v^* \neq v$ and ID_V^* had been queried to H_1 oracle before (the probability that ID_V^* was never queried to H_1 is $1/2^k$), then $\mathcal{B}$ can solve

(1.g) for the discrete logarithm ofP_{pub}. If $ID_S^* \neq ID_S$ or $v^* \neq v$ and both ID_S^* and ID_Shad been queried to H_1oracle before, then $\mathcal{B}$ can solve (1.h) for the discrete logarithm ofP_{pub}. The probability that ID_S^* and ID_Shave never been queried to H_1 oracle is smaller than $2/2^k$.

If $v^* = v$, then definitely $u^* = u$, and $h^* + v^* = h + v$, which gives us $s = s^*$. However, this means that $\psi^* = (u^*, v^*, h^*, s^*)$ was outputted from the designated verifier's proof simulation oracle, contradicting our assumption.

Case II: Suppose that (m^*, σ^*, ID_S^*) is an invalid tuple.

If h^* was never queried to either the signer's proof generation oracle or the designated verifier's proof simulation oracle, then based on forking lemma (Pointcheval & Stern, 1996), $\mathcal{B}$ can repeat the simulation process until $\mathcal{A}$ outputs another tuple $(ID_S^*, ID_V, m^*, \sigma^*, \psi)$ with a non-negligible probability, where $\psi = (u, v, h, s)$ and $h \neq h^*$. Therefore, will get the following equations:

$$\hat{e}(P,P)^{u^*}\hat{e}(cP, {\mu_V}^*P)^{v^*} = \hat{e}(P,P)^{u}\hat{e}(cP, \mu_V P)^{v} \quad (2.a)$$
$$\hat{e}(P,P)^{s^*}\hat{e}(cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(P,P)^{s}\hat{e}(cP, {\mu_S}^*P)^{h+v} \quad (2.b)$$
$$\hat{e}\,(d^*P,P)^{s^*}\hat{e}(d^*cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}\,(dP,P)^{s}\hat{e}(dcP, \mu_S P)^{h+v} \quad (2.c)$$

As in equation (1.a), if $ID_V^* \neq ID_V$ or $v^* \neq v$ and both ID_V^* and ID_Vhad been queried to H_1oracle before, then $\mathcal{B}$ can solve (2.a) for discrete logarithm of$P_{pub} = cP$. The probability that ID_V^* and ID_Vhave never been queried to H_1oracle is smaller than $2/2^k$.

If$ID_V^* = ID_V$, $v^* = v$ and $u^* = u$. However, since $h \neq h^*$ we can observe from equations (2.b) and (2.c) that $f_{(P,P)}^{-1}\,\hat{e}(cP, {\mu_S}^*P) = f_{(d^*P,P)}^{-1}\,\hat{e}(d^*cP, {\mu_S}^*P)$ stating that the signature is valid for signer ID_S^*, contradicting our assumption above.

If h^* was outputted by some previous query to the designated verifier's ID_V^*proof simulation oracle on (m^*, σ^*, ID_S) which produced the proof $\psi = (u, v, h^*, s)$, we obtain the following equations with overwhelming probability.

$$\hat{e}(P,P)^{u^*}\hat{e}(cP, {\mu_V}^*P)^{v^*} = \hat{e}(P,P)^{u}\hat{e}(cP, {\mu_V}^*P)^{v} \quad (2.d)$$
$$\hat{e}(P,P)^{s^*}\hat{e}(cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}(P,P)^{s}\hat{e}(cP, \mu_S P)^{h+v} \quad (2.e)$$
$$\hat{e}\,(d^*P,P)^{s^*}\hat{e}(d^*cP, {\mu_S}^*P)^{h^*+v^*} = \hat{e}\,(dP,P)^{s}\hat{e}(dcP, \mu_S P)^{h+v} \quad (2.f)$$

Now if $v^* \neq v$, then $\mathcal{B}$ can solve (2.d) for the discrete logarithm of P_{pub}. In this case $\mathcal{B}$ will fail if ID_V^* was never queried to oracle. The probability that ID_V^* was never queried to H_1 is smaller than $1/2^k$. If $ID_S^* \neq ID_S$ or $v^* \neq v$ and both ID_S^* and ID_S had been queried to H_1 oracle before, then $\mathcal{B}$ can solve (2.e) for the discrete logarithm of P_{pub}. The probability that ID_S^* and ID_S have never been queried to H_1 oracle is smaller than $2/2^k$.

If $v^* = v$, then definitely $u^* = u$, and $h^* + v^* = h + v$, which gives us $s = s^*$. However, this means that $\psi^* = (u^*, v^*, h^*, s^*)$ was an output of the designated verifier's proof simulation oracle; therefore, contradicting our assumption.

We know that h^* was never outputted from the signer's ID_S^* proof generation oracle since if an invalid tuple (m^*, σ^*, ID_V) is submitted to the signer's ID_S^* proof generation oracle it will output invalid with overwhelming probability.

*Solving the DL Problem:*Let $\epsilon_{failure}$ be the probability that algorithm $\mathcal{B}$ outputs failure in the simulation process and let $\epsilon_{compute}$ be the probability that $\mathcal{B}$ fails in computing the discrete logarithm of P_{pub} after the successful attack by $\mathcal{A}$. Then the probability that $\mathcal{B}$ outputs a correct solution to the discrete logarithm of P_{pub} is at least $\epsilon - (\epsilon_{failure} + \epsilon_{compute})$. We compute the upper bound for $(\epsilon_{failure} + \epsilon_{compute})$ in the worst case as follows. $\mathcal{B}$ can fail in the simulation process when $\mathcal{A}$ queries on proof transcript generation oracles (either from the signer's proof generation oracle or the designated verifier's proof simulation oracle); this can happen with the probability $(q_{CD}(q_{CD} + q_{H_3})/2^k)^2$.
Therefore, the probability that $\mathcal{B}$ outputs failure in the simulation process is $\epsilon_{failure} = (q_{CD}(q_{CD} + q_{H_3})/2^k)^2$ and the probability of $\mathcal{B}$'s failure in computing the Discrete Logarithm problem after successful attack by $\mathcal{A}$ is $\epsilon_{compute} = 8/2^k$ (in Case I and $\epsilon_{compute} = 5/2^k$ for Case II). Hence, the upper bound for $\mathcal{B}$ to solve discrete logarithm problem is as follows:

$$\epsilon' \geq \frac{\epsilon - \left((q_{CD}(q_{CD} + q_{H_3})/2^k)^2 + 8/2^k\right)}{q_{H_3}}$$

The computation time bound can be computed considering the fact that every hash query on $H_i(i = 1, 2, 3)$ and key extraction query requires an exponentiation in$\mathbb{G}_1$, a signature query requires a exponentiation in $\mathbb{G}_1$ and a pairing evaluation, and generation of the confirmation proof transcript as the most expensive computation requires 6 pairing evaluation and 6 exponentiation in$\mathbb{G}_2$. However, $\mathcal{B}$ can reduce the pairing evaluations of the confirmation protocol to 1 by pre-computing $\hat{e}(P, P)$; nonetheless, $\mathcal{B}$ needs to compute 6 multi-exponentiation in $\mathbb{G}_2$. □

7. CONCLUSION

Proof generation protocols are the main attributes of undeniable signature schemes. Since the introduction of NIDV proof systems of Jakobsson et al. (1996), such special proof systems have become very popular due to the properties and efficiency advantages they bring along.

The development of certificate-free undeniable signature schemes however, has been very much relied on employing such proof systems. In this paper, we provided the first provable secure NIDV proof system in pairing-based setting and related its soundness to the difficulty of the Discrete Logarithm problem. Due to their similarities, the proposed proof system can also be incorporated in the body of certificateless and certificate-based undeniable signature schemes.

ACKNOWLEDGEMENT

This research was supported by the FRGS grant (FRGS/2/2010/TK/MMU/02/03) and Multimedia University Graduate Research Assistant Scheme.

REFERENCES

Behnia, R., Heng, S.H. and Gan, C. S. 2012. On the security of pairing-based non-interactive designated verifier proofs of undeniable signature schemes. In *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012 IEEE Conference on* (pp. 207-212).

Boneh, D., and Franklin, M. 2001. Identity-Based Encryption from the Weil Pairing. In J. Kilian (Ed.), *Advances in Cryptology - CRYPTO 2001* (Vol. 2139, pp. 213-229): Springer Berlin / Heidelberg.

Boyd, C. and Foo, E. 1998. Off-line Fair Payment Protocols using Convertible Signatures. In K. Ohta & D. Pei (Eds.), *Advances in Cryptology - ASIACRYPT '98* (Vol. 1514, pp. 271-285): Springer Berlin / Heidelberg.

Chaum, D. and van Antwerpen, H. 1989. Undeniable Signatures. In G. Brassard (Ed.), *Advances in Cryptology — CRYPTO' 89 Proceedings* (Vol. 435, pp. 212-216): Springer Berlin / Heidelberg.

Choon, J., and Hee Cheon, J. 2002. An Identity-Based Signature from Gap Diffie-Hellman Groups. In Y. Desmedt (Ed.), *Public Key Cryptography - PKC 2003* (Vol. 2567, pp. 18-30): Springer Berlin / Heidelberg.

Damgård, I., and Pedersen, T. 1996. *New Convertible Undeniable Signature Schemes*. Paper presented at the Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques.

Duan, S. S. 2008. Certificateless Undeniable Signature Scheme. *Information Sciences.* **178**(3): 742-755.

Goh, E. J. and Jarecki, S. 2003. A signature scheme as secure as the Diffle-Hellman problem. *Advances in Cryptology-Eurocrypt 2003.***2656**: 401-415.

Hess, F. 2003. Efficient Identity Based Signature Schemes Based on Pairings. In K. Nyberg & H. Heys (Eds.), *Selected Areas in Cryptography* (Vol. 2595, pp. 310-324): Springer Berlin / Heidelberg.

Jakobsson, M., Sako, K. and Impagliazzo, R. 1996. *Designated verifier proofs and their applications*. Paper presented at the Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques.

Kudla, C. and Paterson, K. 2005. Non-interactive Designated Verifier Proofs and Undeniable Signatures. In N. Smart (Ed.), *Cryptography and Coding* (Vol. 3796, pp. 136-154): Springer Berlin / Heidelberg.

Kurosawa, K. and Heng, S. H. 2005. 3-Move Undeniable Signature Scheme. *Proceedings ofAdvances in Cryptology - Eurocrypt 2005.***3494**: 181-197.

Libert, B. and Quisquater, J. J. 2004. Identity-Based Undeniable Signatures. *Proceedings ofTopics in Cryptology - Ct-Rsa 2004.* **2964**: 112-125.

Ogata, W., Kurosawa, K. and Heng, S.-H. 2006. The Security of the FDH Variant of Chaum's Undeniable Signature Scheme. *IEEE Transactions on Information Theory.* **52**(5): 2006-2017.

Pointcheval, D. and Stern, J. 1996. Security Proofs for Signature Schemes. In U. Maurer (Ed.), *Advances in Cryptology — EUROCRYPT '96* (Vol. 1070, pp. 387-398): Springer Berlin / Heidelberg.

Sakurai, K. and Miyazaki, S. 2000. An Anonymous Electronic Bidding Protocol Based on a New Convertible Group Signature Scheme. In E. Dawson, A. Clark & C. Boyd (Eds.), *Information Security and Privacy* (Vol. 1841, pp. 385-399): Springer Berlin / Heidelberg.

Shamir, A. 1985. Identity-Based Cryptosystems and Signature Schemes. In G. Blakley & D. Chaum (Eds.), *Advances in Cryptology* (Vol. 196, pp. 47-53): Springer Berlin / Heidelberg.

Wang, G. 2003. An attack on not-interactive deisgnated verifier proofs for undeniable signatures. *Cryptology ePrint Archive, Report 2003/243. Avalilable form: http://eprint.iacr.org/.*

Wu, W., Mu, Y., Susilo, W. and Huang, X. Y. 2008. Provably Secure Identity-Based Undeniable Signatures with Selective and Universal Convertibility. *Information Security and Cryptology.***4990**: 25-39.