2010

Escrowed deniable identification schemes

Pairat Thorncharoensri
University of Wollongong

Qiong Huang
City University of Hong Kong

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Man Ho Allen Au
University of Wollongong

Yi Mu
University of Wollongong, ymu@uow.edu.au

See next page for additional authors

Recommended Citation

Thorncharoensri, Pairat; Huang, Qiong; Susilo, Willy; Allen Au, Man Ho; Mu, Yi; and Wong, Duncan: Escrowed deniable identification schemes 2010.
https://ro.uow.edu.au/infopapers/3560

Escrowed deniable identification schemes

Abstract

Generally, the goal of identification schemes is to provide security assurance against impersonation attacks. Identification schemes based on zero knowledge protocols have more advantages, for example, deniability, which enables the prover to deny an identification proof so that the verifier couldn't persuade others that it is indeed the prover who identified itself to him. This kind of identifications is called 'deniable identification'. However, in some applications we require the existence of a (trusted) party being able to find out an evidence that a party did identify itself to a verifier is required, in order to prevent parties from misbehavior. So in this case 'undeniability' is needed. To the best of our knowledge, an identification scheme that provides both deniability and undeniability does not exist in the literature. In this work we propose the notion of escrowed deniable identification schemes, which integrates both 'escrowed deniability' (undeniability) and 'deniability' properties. Intuitively, in the online communication, a verifier may sometimes need to provide an evidence of a conversation between himself and the prover, for instance, an evidence for the case of misuse of the prover's privilege. We then provide an escrowed deniable identification scheme, and prove its security, i.e. impersonation, deniability and escrowed deniability, in the standard model based on some standard number theoretic assumptions.

Disciplines

Physical Sciences and Mathematics

Publication Details

Thorncharoensri, P., Huang, Q., Susilo, W., Au, M. Ho., Mu, Y. & Wong, D. (2010). Escrowed deniable identification schemes. International Journal of Security and Its Applications, 4 (1), 49-67.

Authors

Pairat Thorncharoensri, Qiong Huang, Willy Susilo, Man Ho Allen Au, Yi Mu, and Duncan Wong

Escrowed Deniable Identification Schemes*

Pairat Thorncharoensri[1], Qiong Huang[2], Willy Susilo[1], Man Ho Au[1], Yi Mu[1] and Duncan Wong[2]

[1]*School of Computer Science & Software Engineering, University of Wollongong, Australia*
[2]*Department of Computer Science, City University of Hong Kong, Hong Kong*
[1]*{pt78,wsusilo,mhaa456,ymu}@uow.edu.au*, [2]*csqhuang@student.cityu.edu.hk*,
[2]*duncan@cityu.edu.hk*

Abstract

Generally, the goal of identification schemes is to provide security assurance against impersonation attacks. Identification schemes based on zero knowledge protocols have more advantages, for example, deniability, which enables the prover to deny an identification proof so that the verifier couldn't persuade others that it is indeed the prover who identified itself to him. This kind of identifications is called 'deniable identification'*. However, in some applications we require the existence of a (trusted) party being able to find out an evidence that a party did identify itself to a verifier is required, in order to prevent parties from misbehavior. So in this case* 'undeniability' *is needed. To the best of our knowledge, an identification scheme that provides both deniability and undeniability does not exist in the literature. In this work we propose the notion of* escrowed deniable identification schemes*, which integrates both 'escrowed deniability' (undeniability) and 'deniability' properties. Intuitively, in the online communication, a verifier may sometimes need to provide an evidence of a conversation between himself and the prover, for instance, an evidence for the case of misuse of the prover's privilege. We then provide an escrowed deniable identification scheme, and prove its security, i.e. impersonation, deniability and escrowed deniability, in the standard model based on some standard number theoretic assumptions.*

Keywords: *identification, deniability, escrowed deniability, zero knowledge, transferability, standard model*

1 Introduction

Since the seminal introduction of zero-knowledge proof by Goldwasser, Micali and Rackoff many interactive identification schemes based on zero-knowledge proofs have been proposed. An interactive identification scheme is a protocol involving two parties, a prover named Peggy, and a verifier named Victor. An identification scheme is to provide an assurance for the verifier that the identity of the prover is indeed as declared. Imagine that Peggy tries to convince Victor of her identity in an online communication. In order to convince Victor, Peggy first provides a public information, which is publicly accessible to every one, and it is associated to her corresponding secret information. By using this secret information, Peggy communicates interactively with Victor, and proves that she is the one who possesses such a secret information corresponding to the public information. Generally, the public information is an instance of a hard problem, which cannot be solved by efficient algorithms, and the secret information is a solution to this instance.

* This work is partially supported by ARC Linkage Project Grant LP0667899.

The security of an identification scheme indicates that no efficient adversary can succeed in impersonating Peggy to Victor (with non-negligible probability), however, in most of the schemes in the literature an identification transcript does reveal the identity of the prover to everyone. That is, Victor can convince anyone that Peggy did identify herself to him. In the following, we borrow the politician example given in [24] and extend it. Politicians would like to enter a building equipped with a smart card identification system. A politician acts as a prover and the smart card reader acts as a verifier. In order to prevent the identities of politicians who entered the building from being revealed to paparazzi by the smart card verification system, *deniable identification* is needed in this case. Now imagine that at sometime, an emergency occurred in the building, and the administrator of it needs to find out who entered this building at certain time interval. If we still use deniable identification, an identification transcript does not necessarily mean that a politician did enter the building at that time interval. Hence, a new variant of identification schemes that we call '*escrowed deniable identification*' is required. In this primitive, there is a (trusted) party who is able to produce an evidence to prove that a prover has participated in the generation of the identification transcript, and furthermore, the verifier cannot do so without the help of the trusted party.

1.1 Related work

From the inspiration of the identification scheme given by Fiat and Shamir [11], some other important identification schemes such as [22, 17, 18, 12, 10] have been proposed. Feige, Fiat, and Shamir proposed an identification scheme in 1988, which is based on the difficulty of inverting RSA, which is a well known cryptographic primitive. Later, another identification scheme based on RSA was proposed by Guillou and Quisquarter [12]. In 1989, based on discrete logarithm problem, the state of art of identification scheme was introduced by Schnorr [22].

In the early 21th century, Bellare and Palacio [4] analyzed Guillou-Quisquarter scheme and Schnorr scheme, and showed that the security against passive attacks of the two schemes can be reduced to standard computational problems such as factoring or discrete logarithms and, their security against active and concurrent attacks can be proven under one-more RSA assumption and one-more discrete logarithm assumption.

After the introduction of bilinear pairing to cryptography, many new problems such as Gap Diffie-Hellman problem, Bilinear Diffie-Hellman problem and etc. have been studied. Many new identification schemes based on these new problems have been proposed in the literature. For example, the first identification based on bilinear Diffie-Hellman problem was given by Kim and Kim [15]. However, their scheme was later broken and improved by Yao, Wang and Wang [25].

For the security definition, Shoup first formalized the definition of impersonation of identification schemes for passive and active attacks [23]. Later, Bellare and Palacio in [4] analyzed the passive and active attacks and formalized the definition of concurrent attacks for identification schemes. However, none of these known schemes considers the notion of deniability.

The concept of deniability in authentication was first introduced by Dwork, Naor and Sahai [9]. Later, the deniable zero knowledge was formalized by Pass [19]. Following the above works on deniability, some works in other areas such as authentication and key exchange ([21, 20]) were proposed.

In 2008, Huang et al.[13] proposed a technique that transforms a weakly unforgeability secured signature scheme into a fully unforgeability secured signature scheme in the standard model. A strong one-time signature is used to sign on a message concatenated with a regular signature signed on a one-time public key. Such a transformation provides a security in the standard model if a signature scheme is the weakly unforgeability secured in the standard model (in other words, the signature is fully unforgeability secured in the random oracle model). We incorporate this technique to construct our EDID scheme in the standard model. In EUROCRYPT 1998, Asokan, Shoup and Waidner[2] proposed a publicly verifiable encryption (fair exchange) protocol. The encrypted signatures are fairly exchange and verifiable that they have been encrypted with the trusted third party's public key and the (encrypted) signatures is valid before each party reveals their decrypted signature in the last round. If one of parties is dishonest, then the other party can request the trusted third party to reveal the signature. We apply this concept in our EDID scheme to achieve the deniable and openable proof of the transcript.

1.2 Our Contributions

To the best of our knowledge, there is no identification which integrates both the deniability and *undeniability*. In this work we first propose the notion of '*escrowed deniable identification* schemes', which protects the identity of the prover from being revealed to the public by the verifier, and in the meanwhile, endows a (trusted) party with the ability to reveal the prover's identity from an identification transcript non-interactively, thus restricting provers from misbehavior. We then provide formal security definitions for escrowed deniable identification schemes, which includes impersonation, deniability, and transferability/escrowed deniability. Finally, we propose a concrete and efficient construction of escrowed deniable identification scheme, and prove its security in the standard model based on some standard number-theoretic assumptions.

We note that there are several works studying the 'escrow' property in other areas, such as verifiable escrowed signatures by Mao [16], 'escrowed linkability of ring signatures' by Chow, Susilo and Yuen [8], and etc. These works are closely related to our work in the definition of 'escrow', however, it is not trivial and easy to transform these works to obtain an escrowed deniable identification scheme. We also note that our new primitive shares some commonalities with the recent notion of 'ambiguous fair exchange' due to Huang et al. [14]. Nonetheless, our primitive requires the transferability property which allows the verifier can prove to others without revealing the prover's signature that the prover ever identified himself (c.f. [14]). Although the construction in [14] can be extended to provide this extra property, we take a different approach to achieve a more efficient scheme in our concrete scheme.

1.3 Paper Organization

In the next section we review some number-theoretic assumptions which will be used in our construction. We provide the definition of escrowed deniable identification scheme in Sec. 3. The formal models of security properties of an escrowed deniable identification scheme are also given here. In Sec. 4 we propose our efficient construction of escrowed deniable identification scheme. Its security is analyzed in Sec. 5. Section 6 is the conclusion of the paper.

2 Preliminaries

2.1 Notation

For the sake of consistency, the following notations will be used throughout the rest of the paper. We use the variable k as the security parameter. We say that an algorithm A is polynomial-time in

k if its running time is bounded by some polynomial. For simplicity, we simply call an algorithm PPT if it is probabilistic polynomial-time in k. A function $f : \mathbb{N} \rightarrow [0, 1]$ is said to be *negligible* in n, if for any constant c and for all sufficiently large n's, it holds that $f(n) < 1/n^c$. The operation of picking an element l at random from a (finite) set L is denoted by $l \overset{\$}{\leftarrow} L$.

In an identification scheme we denote by P (V) an honest prover (verifier), and by P^* (V^*) a malicious prover (verifier) which may deviate from the protocol in an arbitrary way. We also denote by $\langle A(.), B(.)\rangle(.)$ an execution of an interactive protocol between two PPT algorithms A and B, i.e., $\langle P(sk), V\rangle(pk)$ is an execution of the identification scheme between the prover P with a secret key sk and the verifier V on common input pk. We denote by $A^O(.)$ an algorithm A which has oracle access to another function O.

2.2 Basic Concepts on Bilinear Pairings and Complexity Assumptions

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic multiplicative groups generated by g_1 and g_2 respectively. The order of both generators is a prime p. Let $\mathbb{G}_T$ be a cyclic multiplicative group with the same order p. We say that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an admissible bilinear pairing if the followings hold:

1. Bilinearity: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, a, b \in \mathbb{Z}_p$.
2. Non-degeneracy: There exists $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ such that $\hat{e}(g_1, g_2) \neq 1$.
3. Computability: There exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1 \in \mathbb{G}_1,\ g_2 \in \mathbb{G}_2$.

Definition 1 **(q-Strong Diffie-Hellman (q-SDH) Problem).** *Given a $(q + 2)$-tuple $(g_1, g_2, g_2^s, g_2^{s^2}, ..., g_2^{s^q})$ as input, where g_1, g_2 are generators of cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order p and $s \in \mathbb{Z}_p$, output a pair $(c, g_1^{1/(s+c)})$ where $c \in \mathbb{Z}_p^*$. An algorithm $\mathcal{A}$ is said to (t, ϵ) solves the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mathcal{A}$ runs in time t, and*

$$\Pr\left[\mathcal{A}\left(g_1, g_2, g_2^s, g_2^{s^2}, ..., g_2^{s^q}\right) = \left(c, g_1^{\frac{1}{s+c}}\right)\right] \geq \epsilon,$$

where the probability is taken over the random choices of $c, s \in \mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$.

Assumption 1. (q-Strong Diffie-Hellman Assumption [6]) We say that the (q, t, ϵ)-SDH (or q-SDH, for simplicity) assumption in $(\mathbb{G}_1, \mathbb{G}_2)$ holds if there is no PPT algorithm that (t, ϵ) solves the q-SDH problem.

For simplicity, we assume that $\mathbb{G}_1 = \mathbb{G}_2$, and let $\mathbb{G}_1, \mathbb{G}_T$ be two cyclic (multiplicative) groups of prime order p with an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. Let g be a generator of $\mathbb{G}_1$.

Definition 2 **(Decision Linear Diffie-Hellman (DLDH) Problem).** *Given a random 6-tuple $(u, v, w, u^a, v^b, h) \in G_1$ as input, decide whether or not $h = w^{a+b}$. An algorithm $\mathcal{A}$ is said to (t, ϵ) solves the DLDH problem in $\mathbb{G}_1$, if $\mathcal{A}$ runs in time t, and*

$$\left|\Pr\left[\mathcal{A}\left(u, v, w, u^a, v^b, h = w^{a+b}\right) = 1\right] - \Pr\left[\mathcal{A}\left(u, v, w, u^a, v^b, h = w^c\right) = 1\right]\right| \geq \epsilon,$$

where the probability is taken over the random choices of $a, b, c \in \mathbb{Z}_p, u, v, w \in \mathbb{G}_1$, and the random bits consumed by $\mathcal{A}$.

Assumption 2. (Decision Linear Diffie-Hellman Assumption [5]) We say that the (t, ϵ)-DLDH assumption in $\mathbb{G}_1$ holds if there is no PPT algorithm that (t, ϵ) solves the DLDH problem.

Definition 3 (q-Discrete Logarithm (q-DL) Problem). *Given a $(q+2)$-tuple $(g, g^s, g^{s^2}, ..., g^{s^q})$ as input, where g is a generator of group $\mathbb{G}_1$ of prime order p and $s \in \mathbb{Z}_p$, output s. An algorithm $\mathcal{A}$ is said to (t, ϵ) solves q-DL problem in $\mathbb{G}_1$ if $\mathcal{A}$ runs in time t, and*

$$\Pr\left[\mathcal{A}\left(g, g^s, g^{s^2}, ..., g^{s^q}\right) = s\right] \geq \epsilon,$$

where the probability is taken over the random choice of $s \in \mathbb{Z}_p$ and the random bits consumed by $\mathcal{A}$.

Assumption 3. (q-Discrete Logarithm (q-DL) Assumption) We say that the (q, t, ϵ)-DL (or q-DL, for simplicity) assumption in $\mathbb{G}_1$ holds if there is no PPT algorithm that (t, ϵ) solves the q-DL problem.

2.3 Boneh-Boyen Short Signature Without Random Oracles

In this section, we briefly describe the Boneh-Boyen signature scheme (or $BB04$ signature, in short) [6] which we will incorporate to construct our escrowed deniable identification in the standard model. The $BB04$ signature scheme described as follows:

Key Generation (`KeyGen`): `KeyGen` randomly selects $g_a \in \mathbb{G}_1;\ g_b \in \mathbb{G}_2;\ \alpha, \eta \in \mathbb{Z}_p$ and computes $\mathcal{U} = g_2^{\alpha},\ \mathcal{V} = g_2^{\eta}, \mathcal{Z} = \hat{e}(g_1, g_2)$. The public key is $pk_S = (g_1, g_2, \mathcal{U}, \mathcal{V}, \mathcal{Z})$ and the secret key is $sk_S = (\alpha, \eta)$.

Signing (`Sign`): Given a secret key sk_S, a public key pk_S and a message $m \in \mathbb{Z}_p$, `Sign` randomly selects $r \in z_p$ and computes $\sigma \leftarrow g_1^{1/(\alpha + r\cdot\eta + m)}$. The signature on message m is (σ, r)

Verification (`Verf`): Given a signature, a public key pk_S and a message $m \in \mathbb{Z}_p$, `Verf` checks whether $\hat{e}(\sigma, \mathcal{U} \cdot \mathcal{V}^r \cdot g_2^m) = \mathcal{Z}$. If it holds then `Accept`, otherwise `Reject`.

Theorem 1. *If the (t', q, ϵ')-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$, then the $BB04$ signature scheme is (t, q_S, ϵ)-secure against strong existential forgery under an adaptive chosen message attack where T is the maximum time for an exponentiation in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{Z}_p$, $q_s \leq q$, $\epsilon \approx 2\epsilon'$ and $t \leq t' - \Theta(q^2 T)$ [6].*

Note that, for simplicity, we define $\mathbb{G}_1 = \mathbb{G}_2$ in our scheme and we denote the $BB04$ short signature as the weakly secure short signature. The $BB04$ short signature is different from the $BB04$ signature by removing $\mathcal{V}, \eta$ from the public key and secret key, and computing a signature as $\sigma \leftarrow g_1^{1/(\alpha+m)}$.

3 Escrowed Deniable Identification

In this section, we provide a formal model of an escrowed deniable identification scheme and its security model. Different from previous definitions of identification schemes [4, 18, 22, 23, 1, 3], we introduce a trusted third party into escrowed deniable identification, who has the power of invoking the deniability of the prover. Therefore, in the security models of escrowed deniable identification schemes, we consider a new property, '*transferability*'. It provides security guarantee for the trusted third party and the prover, which preserves the privacy for all other cases except the case in dispute.

3.1 Escrowed Deniable Identification Schemes

We introduce a notion called *escrowed deniable identification scheme* (EDID) that balances both the need for deniability and the need for undeniability in identification schemes. In an escrowed deniable identification scheme, in addition to that the prover can deny an identification transcript, a trusted authority can convert a deniable identification transcript into an undeniable one, enabling anyone to verify ownership of the transcript.

Formally, an EDID scheme involves a prover P, a verifier V, a trusted authority TA, and any third party V'. It consists of the following algorithms and protocols:

$\texttt{Setup}$**:** On input 1^k, where k is the security parameter, the algorithm generates a system parameter, i.e. $\mathsf{param} \leftarrow \texttt{Setup}(1^k)$.

$\texttt{KeyGen}^T$**:** On input param, it generates a public/secret key pair (pk_T, sk_T) for the trusted authority, i.e. $(pk_T, sk_T) \leftarrow \texttt{KeyGen}^T(\mathsf{param})$.

$\texttt{KeyGen}^P$**:** On input param, it generates a public/secret key pair (pk_P, sk_P) for the prover, i.e. $(pk_P, sk_P) \leftarrow \texttt{KeyGen}^P(\mathsf{param})$.

Identification protocol (P, V)**:** This is an interactive protocol between the prover P and the verifier V. It consists of four rounds of communication and six PPT algorithms, $(\texttt{Cmt}_V, \texttt{Cmt}_P, \texttt{Ch}, \texttt{Rsp}, \texttt{Check}_P, \texttt{Check}_V)$, where $(\texttt{Cmt}_P$, $\texttt{Check}_P)$ and $(\texttt{Check}_V$, $\texttt{Cmt}_V)$ are sets of algorithms to generate commitments and to verify the commitment run by the prover P and the verifier V, respectively, $\texttt{Ch}$ is an algorithm to disclose the challenge, and $\texttt{Rsp}$ is an algorithm run by the prover P to generate the response after the process of commitment generation, challenge disclosure and commitment verification.

- Step 1. V chooses a challenge c at random from a certain domain, and computes $T \leftarrow \texttt{Cmt}_V(c)$. V then sends T to P.
- Step 2. P chooses r at random from a certain domain, and computes $a \leftarrow \texttt{Cmt}_P(r)$. P then sends a to V.
- Step 3. V runs $\texttt{Ch}$ to reveal a random challenge c, and sends it to P.
- Step 4. After receiving V's challenge c, P then runs $b \leftarrow ck_P(c, T)$. If $b = 0$, P aborts; otherwise, it computes its response by running $z \leftarrow \texttt{Rsp}(sk_P, r, c)$, and sends z to V.
- Step 5. V checks the validity of P's response by running $\texttt{Check}_V(pk_T, pk_P, a, c, z)$. If the output is '1', V accepts; otherwise, it rejects.

Open protocol (TA, V)**:** An open protocol can be formalized by two (probabilistic) polynomial-time algorithms $\texttt{Open}, \texttt{Verf}$, where $\texttt{Open}$ is invoked by TA, and $\texttt{Verf}$ is executed by the verifier V. On input a transcript tr and the secret key of TA, $\texttt{Open}$ outputs an evidence to affirm the authenticity of tr. $\texttt{Verf}$ is an algorithm for validating the validity of the evidence with respect to tr and pk_P. It takes as input pk_T, pk_P, tr and the evidence, and outputs 1 for accepting or 0 for rejecting the evidence.

Transfer protocol (V, V')**:** A transfer protocol is an interactive protocol between the verifier V, who possesses a transcript tr and its affirmative evidence from the trusted authority (TA), and any third party V'. The aim of the protocol is to convince V' that tr indeed represents an execution of the identification protocol between P and V.

The completeness can be defined in a natural way. Next we define other security properties for an escrowed deniable identification scheme.

3.2 Deniability

Roughly speaking, deniability indicates that given a transcript of an execution of the identification protocol, the prover is able to deny that he is the prover in the execution. To achieve the deniability, we require that the verifier itself could generate this transcript. Formally, we consider the following definition, which share a similarity with that of zero knowledge.

Definition 4 (Deniability). *An escrowed deniable identification scheme EDID is* deniable *if for any $Param \leftarrow \texttt{Setup}(1^k)$, $(pk_T, sk_T) \leftarrow \texttt{KeyGen}^T(\textsf{param})$ and $(pk_P, sk_P) \leftarrow \texttt{KeyGen}^P(\textsf{param})$, for any PPT algorithm $\mathcal{D}$, for any verifier strategy V^*, there exists a PPT algorithm S which has oracle access to V^*, such that*

$$|\Pr[\mathrm{Expt}_1(k) = 1] - \Pr[\mathrm{Expt}_2(k) = 1]| = \epsilon(k),$$

where $\epsilon(\cdot)$ is a negligible function in k, and Expt$_1(k)$ and Expt$_2(k)$ are defined as follows:

$\mathrm{Expt}_1(k)$:	$\mathrm{Expt}_2(k)$:
$tr \leftarrow \langle P(sk_P), V^* \rangle (pk_T, pk_P)$	$tr' \leftarrow S^{V^*}(pk_T, pk_P)$
$b \leftarrow \mathcal{D}(pk_T, pk_P, tr)$	$b' \leftarrow \mathcal{D}(pk_T, pk_P, tr)$
return b	*return b*

where the probabilities are taken over the random bits used in `Setup`*,* $\texttt{KeyGen}^T$*,* $\texttt{KeyGen}^P$*, and random bits consumed by P, V^*, S and $\mathcal{D}$.*

3.3 Impersonation

An identification scheme is secure against the impersonation meant that no one except the prover P with its public key pk_P can identify itself to others as P. In this work we consider the most common impersonation attacks, i.e. passive attacks and active attacks, which are described as below:

- **Passive Attack** (imp-pa): This is the weakest form of attacks considered for impersonation. An adversary can only listen to the interaction between a prover and a verifier, and then begin to impersonate the prover after the interaction.
- **Active Attack** (imp-aa): This attack is stronger than the one above. In an active attack, the adversary, acting as a (cheating) verifier, actively interacts with prover clones in sequence. After the last execution of the identification protocol is over, it starts to impersonate the prover to others.

(Impersonation under Active Attack): An imp-aa adversary $\mathcal{A}$ is a pair of PPT algorithms $(\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ acts as V^* and $\mathcal{A}_2$ acts as P^*. Let st denote the state of information. The active attack is initialized by first calling `Setup`, $\texttt{KeyGen}^T$ and $\texttt{KeyGen}^P$ to generate public/secret key pairs (pk_T, sk_T) and (pk_P, sk_P) for the trusted authority and the prover respectively. Taking public keys pk_T and pk_P as input, the adversary $\mathcal{A}$ then performs its attack in the following two phases:

- Phase 1. (Learning Phase) Given input pk_T, pk_P, the adversary $\mathcal{A}_1$ is allowed to interact with P's clones sequentially. When each of P's clones interacts with $\mathcal{A}_1$, it is initialized with (pk_P, sk_P), pk_T and fresh random coins. Later, $\mathcal{A}_1$ outputs st to be passed onto $\mathcal{A}_2$. This completes phase 1.

- Phase 2. (Impersonation Phase) At the beginning of phase 2, V is initialized with the public keys pk_T, pk_P, while the adversary $\mathcal{A}_2$ is given st. Then $\mathcal{A}_2$ tries to impersonate P to V. At the end of this phase, V outputs a decision bit b, indicating `accept` or `reject`.

The adversary $\mathcal{A}$ is said to be *successful* in the attack if V outputs 1 at the end of Phase 2. Formally, we consider the following experiment:

```
$\mathrm{Expt}_{\mathcal{A}}^{\mathsf{imp-aa}}(k)$:
  $\mathsf{param} \leftarrow \mathtt{Setup}(1^k)$
  $(pk_T, sk_T) \leftarrow \mathtt{KeyGen}^T(\mathsf{param})$
  $(pk_P, sk_P) \leftarrow \mathtt{KeyGen}^P(\mathsf{param})$
  $(\perp, st) \leftarrow \mathcal{A}_1^{P(sk_P)}(pk_T, sk_T, pk_P)$
  $(\perp, b) \leftarrow \langle \mathcal{A}_2(sk_T, st), V \rangle (pk_T, pk_P)$
  return $b$
```

where an oracle call to $P(sk_P)$ results in an execution of the identification protocol with the prover P and a transcript tr is returned.

Definition 5 (Security against Impersonation under Active Attack). *We say an escrowed deniable identification scheme EDID is* secure against impersonation under active attack*, if there is no PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *such that the probability* $\Pr[\mathrm{Expt}_{\mathcal{A}}^{\mathsf{imp-aa}}(k) = 1]$ *is* negligible *in k.*

Note that in the definition above the adversary can be the trusted authority. That is, even TA cannot impersonate the prover in an active attack.

3.4 Transferability

Intuitively, the notion of *transferability* in escrowed deniability identification schemes is aimed to reveal the transcript confirmation or evidence that proves the validity of the prover of the transcript. The idea is that a verifier is provided with evidence for a case in dispute to prove to another party who would like to be convinced of the validity of the transcript. To complete this idea, a trusted authority is involved to process an opening (transferring) algorithm. Unlike the deniability property in a general identification scheme (ie., zero-knowledge protocol based identification schemes), the verifier can now convince another party that the transcript of an identification scheme is actually due to an interaction with the claimed prover with the help of or evidence from a trusted party. In the experiment below, the adversary is modeled as a malicious verifier who tries to convince any third party to accept the transcript without the help of the trusted authority. Hence, the trusted authority is viewed as an opening oracle $\mathcal{O}_{EDID_{Open}}$ who answers queries for opening the chosen transcript. We provide a formal definition of *transferability* as follows:

Let V^* be any verifier strategy (honest or malicious). Let (pk, sk) be the prover's public key and private key generated by the key generation algorithm of the identification scheme, and let (pk_T, sk_T) be the TA's public key and private key, respectively, generated by the key generation algorithm of the identification scheme. Let $tr \leftarrow \langle P(sk), V^* \rangle (pk)$ be the transcript of an interaction between P and V^*, and let $\sigma \leftarrow \langle TA(sk_{TA}), V^* \rangle (pk)$ be the confirmation evidence σ of an interaction between TA and V^*. Let `Verf` be the verifier's decision algorithm which takes a transcript tr and its confirmation evidence σ as inputs and outputs 1 or 0, which indicate '`accept`' or '`reject`', respectively. Let S be a probabilistic polynomial-time algorithm. We consider the following experiment:

```
$\mathrm{Expt}_{\mathcal{A}}^{\mathsf{tran}}(k)$:
    $(pk, pk_T, sk, sk_T) \leftarrow \mathtt{KeyGen}(1^k)$
    $(\perp, st) \leftarrow \mathcal{A}_1^{\mathcal{O}_{EDID_{ID}}, \mathcal{O}_{EDID_{Open}}}(pk, pk_T)$
    $(tr^*, \sigma^*) \leftarrow \mathcal{A}_2(st)$
    If $(tr^*, \sigma^*)$ has been queried to $\mathcal{O}_{EDID_{ID}}, \mathcal{O}_{EDID_{Open}}$ then $\perp$,
    otherwise, in the transfer protocol(or any other protocol for
    transferring the proof $(tr^*, \sigma^*)$),
    $(\perp, b) \leftarrow \langle \mathcal{A}_2(tr^*, \sigma^*), AnyV \rangle (pk, pk_T)$
    Return $b$
```

Adversary $\mathcal{A}$ is said to be *successful* in the attack if $AnyV$ outputs $b = \mathtt{accept}$.

Definition 6 (Security against Transferability Attack). *An identification scheme* $\mathsf{ID} = (\mathtt{KeyGen}$*,* P*,* $V)$ *is said to be secure against transferability attack if there is no probabilistic polynomial-time* tran *adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *such that the probability* $\Pr[\mathrm{Expt}_{\mathcal{A}}^{\mathsf{tran}}(k) = 1]$ *is* negligible *in* k*.*

```
Oracle $\mathcal{O}_{EDID_{ID}}$:
    $tr = (a, b, z) \leftarrow \langle \mathcal{O}_{EDID_{ID}}(sk_P), V \rangle (pk_T, pk_P)$
    Return $tr$
```

```
Oracle $\mathcal{O}_{EDID_{Open}}(tr)$:
    $\sigma \leftarrow \mathtt{Open}(sk_T, pk_T, tr)$
    $b \leftarrow \mathtt{Verf}(tr, \sigma, pk_T)$
    Return $\sigma$ iff $b$ = accept
    Otherwise, return $\perp$
```

```
Oracle $\mathcal{O}_H(str)$:
    $m \leftarrow H(str)$
    Return $m$
```

Figure 1: Oracle for adversary attacking transferability of escrowed deniability identification scheme

4 Our Construction

4.1 High Level Idea

Before presenting our construction of escrowed deniable identification schemes, we will first describe our idea and intuition behind our construction. Let TA's pair of public/secret keys be (pk_T, sk_T). Firstly, P generates a commitment $\mathtt{Cmt}$, and V replies with a random challenge c. Then, P signs both $\mathtt{Cmt}$ and c to obtain σ. Next, P will verifiably encrypt σ using the TA's public key pk_T. That is, $\hat{\sigma} \leftarrow VE_{pk_T}(\sigma)$. Then, P sends $\hat{\sigma}$ to V. V can check the validity of $\hat{\sigma}$ with respect to pk_T and pk_P, but V cannot transfer this conviction to anyone else (due to the *indistinguishability* property of the verifiable encryption used). When mischievous behavior occurs, TA converts the transcript and makes it undeniable. TA can decrypt $\hat{\sigma}$ using sk_T to obtain σ, and since σ is a regular digital signature generated by P, it is undeniable.

4.2 The Construction

In this section, we present our scheme based on the idea outlined above. The construction uses a Boneh-Boyen short signature scheme and verifiable encryption scheme due to Boneh et al. [6, 7]. We incorporate the technique in [13] to construct our EDID scheme in the standard model. The scheme works as follows.

1. $\texttt{Setup}$: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be two multiplicative cyclic groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$ for some large prime p. g, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is a bilinear pairing. Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be a collision-resistant hash function. The system parameter param then consists of $(\mathbb{G}_1, \mathbb{G}_T, \hat{e}, p, g, g_1, g_2, H)$.
2. $\texttt{KeyGen}^T$: Given the public parameter param, $\texttt{KeyGen}^T$ Select random numbers $x, y \in \mathbb{Z}_p; W \in \mathbb{G}_1$ and compute $V = W^y$, $U = V^x$. The public key and private key of the trusted authority are $pk_T = (U, V, W)$ and $sk_T = (x, y)$ respectively.
3. $\texttt{KeyGen}^P$: Given the public parameter param, $\texttt{KeyGen}^P$ selects a random number $s \in \mathbb{Z}_p$ and compute $S_P = g_1^s$. The public key and private key of the prover are $pk_P = S_P$ and $sk_P = s$ respectively.
4. Identification protocol: The protocol comprises two parts. The first part is a 4-round zero-knowledge proof protocol of the Schnorr Identification, in which the prover P proves to the verifier V that he knows the secret key s which is the discrete logarithm of the public key S_P to base g_1. In the second part of the identification protocol, prover P generates a $BB04$ short signature σ on the 4-round Schnorr Identification transcript he just carried out with the verifier. P then computes $\hat{\sigma}$, which is the verifiable encryption of σ under the TA's public key, and sends it to the verifier. Finally, P proves, in an interactive manner, to the verifier that $\hat{\sigma}$ is correctly formed. Following the description above, the protocol will be more than four rounds. Optimization of the round efficiency of the protocol can be done by setting σ to be the signature on the first two rounds of the 4-round Schnorr Identification protocol and conducting the proof-of-correctness of $\hat{\sigma}$ in parallel with the Schnorr Identification with the verifier. The resulting protocol remains four rounds and it is shown as follows.
   (a) 1^{st} Round (V to P). (Commitment of Challenge.) V randomly generates $c, d \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, computes $\mathbb{T} = g_1^c g_2^d$ and sends $\mathbb{T}$ to P.
   (b) 2^{nd} Round (P to V).
      i. P randomly generates $r_s \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, computes $\texttt{T} = g_1^{r_s}$. Now, P runs the $\texttt{KeyGen}$ of $BB04$ signature for the one time public key pk_{OT} and the one time secret key sk_{OT}. However, P can simply use some common parameter from param such as $p, \hat{e}$ for $BB04$ signature. Hence, on input param, P randomly selects $g_a, g_b \in \mathbb{G}_1$; $\alpha, \eta \in \mathbb{Z}_p$ and computes the one time public key $pk_{OT} = (g_a, g_b, \mathcal{U} = g_b^{\alpha}, \mathcal{V} = g_b^{\eta}, \mathcal{Z} = \hat{e}(g_a, g_b))$ and the one time secret key $sk_{OT} = (\alpha, \eta)$.
      ii. P randomly selects $a, b \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $E_1 = U^a$ and $E_2 = V^b$. Then, P randomly generates $r_a, r_b \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $A_1 = U^{r_a}$ and $A_2 = V^{r_b}$.
      iii. Let $m = H(pk_{OT})$. Now, P computes a signature $\sigma = g^{\frac{1}{s+m}}$. Then, P computes $E_3 = \sigma W^{a+b}$ and $A_3 = \hat{e}(W, S_P g_1^m)^{r_a + r_b}$. Parse $\hat{A}$ as (A_1, A_2, A_3) and $\hat{E}$ as (E_1, E_2, E_3).
      iv. Let $\bar{m} = H(\texttt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$. On input pk_{OT}, sk_{OT}, P randomly chooses $\kappa \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and computes $\bar{\sigma} = g_a^{\frac{1}{\alpha + \kappa \cdot \eta + \bar{m}}}$. Then P sends $(\texttt{T}, pk_{OT}, \bar{\sigma}, \kappa, \hat{E}, \hat{A})$ to V.
   (c) 3^{rd} Round (V to P). (Challenge.) V sends c, d to P.
   (d) 4^{th} Round (P to V). (Response.) P checks if $T \overset{?}{=} g_1^c g_2^d$. Output $\perp$ is the check fails. Otherwise compute $z_s = r_s - cs$, $z_a = r_a - ca$ and $z_b = r_b - cb$. Set $\hat{Z}$ as (z_s, z_a, z_b) and send $\hat{Z}$ to V.
   (e) (Verification.) V computes $\bar{m} = H(\texttt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$ and $m = H(pk_{OT})$ and outputs $\texttt{accept}$ if the following holds:

$$T_1 \overset{?}{=} S_P^c g_1^{z_s}, \ \hat{e}(g_a, g_b) \overset{?}{=} \hat{e}(\sigma, \mathcal{U} \cdot \mathcal{V}^{\kappa} \cdot g_2^{\bar{m}}), \ \ A_1 \overset{?}{=} E_1^c U^{z_a},$$

$$A_2 \overset{?}{=} E_2^c V^{z_b},\ A_3 \overset{?}{=} \left(\frac{\hat{e}(E_3, S_P g_1^m)}{\hat{e}(g, g_1)}\right)^c \hat{e}(W, S_P g_1^m)^{z_a+z_b}.$$

Output `reject` otherwise.

5. Open protocol: A protocol can be denoted by $OP = ($ `Open`, `Verf`$)$, where `Open` and `Verf` are PPT algorithms used in the protocol detailed in Figure 2.
6. Transfer protocol: A protocol can be denoted by $TP = ($ `Cmt`, `Ch`, `Rsp`, `Check` $)$, where `Cmt`, `Ch`, `Rsp` and `Check` are PPT algorithms used in the following protocol, where the verifier V proves that a transcript denoted as tr is indeed generated by P to any third party verifier. This protocol is illustrated in Figure 2.

5 Security Analysis

In this section, we provide security proofs for our proposed EDID schemes, which include deniability, security against impersonation and transferability (escrowed deniability). For a brief representation, we first define the following notations, which we will use throughout the rest of this section.

5.1 Deniability

We provide the proof that the identification protocol and the transfer protocol in our EDID scheme are zero knowledge protocol. First, the completeness of the identification protocol and the transfer protocol in EDID scheme are straight forward, hence, it will be omitted. Secondly, the zero knowledge proof of the identification protocol and the transfer protocol in EDID scheme are stated as the following theorems.

Theorem 2. *The identification protocol in our identification scheme EDID is deniable.*

Proof. Let $\mathcal{S}$ be a simulator and V^* be any verifier. Given the public keys $pk_T = (U, V, W)$, and $pk_P = S_P = g_1^s$, algorithm $\mathcal{S}$ simulates transcripts as follows:

1. First, $\mathcal{S}$ receives T from V^*, and then computes its response as follows:
   - $\mathcal{S}$ first selects random generators $\mathtt{T}', A_1', A_2', A_3', \sigma' \overset{\$}{\leftarrow} \mathbb{G}_1$.
   - Then, $\mathcal{S}$ run `KeyGen` of $BB04$ signature scheme and obtain $pk_{OT}' = (g_{a'}, g_{b'}, \mathcal{U} = g_{b'}^{\alpha'}, \mathcal{V} = g_{b'}^{\eta'}, \mathcal{Z}' = \hat{e}(g_{a'}, g_{b'}))$ and $sk_{OT}' = (\alpha', \eta')$.
   - Next, $\mathcal{S}$ chooses integers $a', b', \kappa' \overset{\$}{\leftarrow} \mathbb{Z}_p^*$ and compute $E_1' = U^{a'}$, $E_2' = U^{b'}$. $\mathcal{S}$ computes $m' = H(pk_{OT}')$.
   - $\mathcal{S}$ computes $E_3' = \sigma' W^{a+b}$ and $\bar{m}' = H(\mathtt{T}', \mathbb{T}, pk_P, E_1', E_2', E_3', A_1', A_2', A_3')$. Then compute $\bar{\sigma}' = g_{a'}^{\frac{1}{\alpha' + \kappa' \cdot \eta' + \bar{m}'}}$.
   - $\mathcal{S}$ responds $\mathcal{A}$ with $(\mathtt{T}', pk_{OT}', \bar{\sigma}', \kappa', E_1', E_2', E_3', A_1', A_2', A_3')$.
2. After V^* replies with c' and d', $\mathcal{S}$ checks the validity of (c', d') with respect to $\mathbb{T}$ and then rewinds V^* to previous state and computes a new response as follows:
   - Run `KeyGen` of $BB04$ signature scheme and obtain $pk_{OT} = (g_a, g_b, \mathcal{U} = g_b^{\alpha}, \mathcal{V} = g_b^{\eta}, \mathcal{Z} = \hat{e}(g_a, g_b))$ and $sk_{OT} = (\alpha, \eta)$.
   - Select $\kappa, z_s, a, b, z_a, z_b \overset{\$}{\leftarrow} \mathbb{Z}_p^*$; $E_3 \overset{\$}{\leftarrow} \mathbb{G}_1$; $E_1 = U^a$; $E_2 = V^b$; $T_1 = S_P^c g_1^{z_s}$; $A_1 = E_1^{c'} U^{z_a}$; $A_2 = E_2^c V^{z_b}$.
   - Then, compute $m = H(pk_{OT})$, $A_3 = (\frac{\hat{e}(E_3, S_P g_1^m)}{\hat{e}(g, g_1)})^c \hat{e}(W, S_P g_1^m)^{z_a+z_b}$.

Open

TA		**V**
	$\xleftarrow{tr}$	$tr \overset{\mathtt{def}}{=} (\mathbb{T}, \mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, \hat{A}, \hat{E}, c, d, \hat{Z})$
$\bar{m} \leftarrow H(\mathbb{T}, \mathtt{T}, pk_P, E_1,$ $E_2, E_3, A_1, A_2, A_3);$ $m \leftarrow H(pk_{OT});$ iff not $(T_1 \overset{?}{=} S_P^c g_1^{z_s} \wedge$ $\hat{e}(g_a, g_b) \overset{?}{=} \hat{e}(\sigma, \mathcal{U} \cdot \mathcal{V}^{\kappa} \cdot g_2^{\bar{m}}) \wedge$ $A_1 \overset{?}{=} E_1^c U^{z_a} \ \wedge \ A_2 \overset{?}{=} E_2^c V^{z_b}$ $\wedge\ A_3 \overset{?}{=} \big(\frac{\hat{e}(E_3, S_P g_1^m)}{\hat{e}(g, g_1)}\big)^c$ $\hat{e}(W, S_P g_1^m)^{z_a + z_b})$ then $\perp$. Otherwise, $\sigma \leftarrow E_3/(E_1^{1/xy} E_2^{1/x});$ iff $\hat{e}(g, g_1) \overset{?}{=} \hat{e}(\sigma, S_P g_1^m)$ then $\mathtt{Open}(sk_{TA}) \overset{\mathtt{def}}{=} \sigma$, else $\perp$	$\xrightarrow{\sigma}$	
		$\mathtt{Verf}$: iff $\hat{e}(g, g_1) \overset{?}{=} \hat{e}(\sigma, S_P g_1^m)$ then $\mathtt{Verf} = \mathtt{accept}$,else $\mathtt{Verf} = \mathtt{reject}$

Transfer

V		**Any third party**
	$\xleftarrow{\mathfrak{T}}$	$c', d' \overset{\$}{\leftarrow} \mathbb{Z}_p^*; \mathfrak{T} = g_1^{c'} g_2^{d'}$
$a', r_{a'} \overset{\$}{\leftarrow} \mathbb{Z}_p^*;$ $D_1 \leftarrow \sigma g_2^{a'};$ $D_2 \leftarrow \hat{e}(g_2, S_P g_1^m)^{r_{a'}};$ $\mathtt{Cmt}(pk_P, tr, r_{a'}, a') \overset{\mathtt{def}}{=} (D_1, D_2)$	$\xrightarrow{tr, D_1, D_2}$	
	$\xleftarrow{c', d'}$	$\mathtt{Ch} \overset{\mathtt{def}}{=} (c', d')$
iff $(\mathfrak{T} \overset{?}{=} g_1^{c'} g_2^{d'})$ then $z \leftarrow r_{a'} + c'a';$ $\mathtt{Rsp}(pk_P, sk_P, \mathtt{Cmt}, \mathtt{Ch}) \overset{\mathtt{def}}{=} z,$ else $\perp$	$\xrightarrow{z}$	
		$\mathtt{Check}$: $\bar{m} \leftarrow H(\mathbb{T}, \mathtt{T}, pk_P, E_1, E_2,$ $E_3, A_1, A_2, A_3)$ and $m \leftarrow H(pk_{OT})$. iff $(\hat{e}(D_1, S_P g_1^m)/\hat{e}(g, g_1))^{c'} \overset{?}{=}$ $D_2^{-1} \hat{e}(g_2, S_P g_1^m)^z$ $\wedge\ \hat{e}(g_a, g_b) \overset{?}{=} \hat{e}(\sigma, \mathcal{U} \cdot \mathcal{V}^{\kappa} \cdot g_2^{\bar{m}})$, then the transcript tr was generated by P

Figure 2: Open & Transfer Protocols

- Compute $\bar{m} = H(\mathtt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$, $\bar{\sigma} = g_a^{\frac{1}{\alpha+\kappa\cdot\eta+\bar{m}}}$.
- Return $(\mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, \hat{E}, \hat{A})$ to V^*.

3. Finally, upon the receiving of c and d from V^*, $\mathcal{S}$ aborts if $c \neq c'$ or $d \neq d'$. Otherwise, $\mathcal{S}$ replies with (z_s, z_a, z_b).

From the structure of protocol and the simulation process, the difference between the distributions of the real transcripts $\mathcal{T} = \{tr\}$ and the simulated transcripts $\hat{\mathcal{T}} = \{\widehat{tr}\}$ only lies in the event that the $c \neq c'$ or $d \neq d'$. We denote by INEQ this event. We can see that the probability that event INEQ happens only with negligible probability. If this is not the case, we can compute the discrete logarithm of g_2 with respect to the base g_1. If $c \neq c'$, it turns out that $d \neq d'$ as well. Since $g_1^c g_2^d = T = g_1^{c'} g_2^{d'}$, we get that $g_2 = g_1^{(c-c')/(d'-d)}$. As the group order p is known, we can get that $\log_{g_1} g_2 = (c-c')/(d'-d) \bmod p$. This contradicts the discrete logarithm assumption and hence, we can run this experiment to solve the discrete logarithm problem by setting $g_1 = h$ and $g_2 = h^a$, where h, h^a are instances of the discrete logarithm problem. Therefore, we conclude that for any PPT algorithm $\mathcal{D}$,

$$\Big|\Pr[\mathcal{D}(pk_T, pk_P, tr) = 1] - \Pr[\mathcal{D}(pk_T, pk_P, \hat{tr}) = 1]\Big| \leq \Pr[\text{INEQ}],$$

which is also negligible in k. □

Theorem 3. *The transfer protocol in our identification scheme* EDID = ($\mathtt{Setup}$, $\mathtt{KeyGen}$, P, V, TA, $AnyV$) *is zero knowledge protocol.*

Proof. Let $\mathcal{S}$ be a simulator and $\mathcal{A}$ plays a role of any arbitrary verifier V^*. Let tr be a transcript that $\mathcal{S}$ want to prove a procession of a proof of transcript tr, which is a signature σ. $\mathcal{S}$ simulates transcripts as follows:

1. First, $\mathcal{S}$ receives $\mathbb{T}$ from $\mathcal{A}$. Upon the access to the random tape used by V^*, $\mathcal{S}$ obtain c' and d', where $T = g_1^{c'} g_2^{d'}$.
2. Then, $\mathcal{S}$ computes a response as follows:
   - Select $z \xleftarrow{\$} \mathbb{Z}_p^*$ and $D_1 \xleftarrow{\$} \mathbb{G}_1$.
   - Compute $m = H(pk_{OT}, \bar{\sigma})$. Then compute
     $D_2 = \big(\frac{\hat{e}(g,g_1)}{\hat{e}(D_1, S_P g_1^m)}\big)^{c'} \hat{e}(g_2, S_P g_1^m)^z$.
   - Then $\mathcal{S}$ returns (D_1, D_2) to $\mathcal{A}$.
3. Upon the receiving of c and d, $\mathcal{S}$ aborts if $c \neq c'$ or $d \neq d'$. Otherwise, $\mathcal{S}$ replies with z.

Let tr_t be a transcript of the real transfer transcripts and $\widehat{tr_t}$ be a transcript of the simulated transfer transcripts. From the structure of protocol and the simulation process, the difference between the distributions of the real transcripts $\mathcal{T} = \{tr_t\}$ and the simulated transcripts $\hat{\mathcal{T}} = \{\widehat{tr_t}\}$ only happen when the $c \neq c'$ or $d \neq d'$ in the reveal of challenge step. Since both $c, d \xleftarrow{\$} \mathbb{Z}_p$, the probability that $c \neq c'$ or $d \neq d'$ but $T = g_1^{c'} g_2^{d'} = g_1^c g_2^d$ is equal to $1/p$. Therefore, the distance between the probability distribution of $\mathcal{T}$ and $\hat{\mathcal{T}}$ caused by this reason is no more than $1/2^k$ where k is a security parameter and $k \approx |p|$.
Therefore, the distance between the probability distribution of $\mathcal{T}$ and $\hat{\mathcal{T}}$ is

$$\Big|\Pr[\langle P(sk), V^*\rangle(pk) = 1] - \Pr[S^{V^*}(pk) = 1]\Big| < 1/2^k,$$

which is *negligible* for sufficiently large k. □

5.2 Security Analysis for Impersonation

The following is the security analysis of our escrowed deniability identification scheme against impersonation under active attacks. Before this, we recall that a confirmation evidence generated in the 2^{nd} round of the protocol is indeed a Boneh-Boyen basic short signature.

Theorem 4. *Our identification scheme EDID is secure against impersonation under active attacks in the standard model, if the q-DL assumption holds.*

Proof. Suppose that there exists a PPT imp-aa adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ for EDID scheme such that the probability $\Pr[\mathrm{Expt}_{\mathcal{A}}^{\mathsf{imp-aa}}(k) = 1]$ is non-negligible. Then we show that there exists a PPT adversary $\mathcal{F}$ for solving the q-DL problem using $\mathcal{A}$ as a subroutine. $\mathcal{F}$ is given $g, g^s, g^{s^2}, \cdots, g^{s^q} \in \mathbb{G}_1$ as input. $\mathcal{F}$ computes g_1 and S_P in the same way with the proof of Lemma 1 in [6]. $\mathcal{F}$ then sets $g = g_1^{\gamma}$ and $g_2 = g^{\beta}$, where $\gamma, \beta \xleftarrow{\$} \mathbb{Z}_p$. Let $OT = \{pk_{OT,1}, sk_{OT,1}, ..., pK_{OT,q_H}, sk_{OT,q_H}\}$ be the list of pre-computed one-time public keys and secret keys. Let $LM = \{m_1 = H(pk_{OT,1}), ..., m_{q_H} = H(pk_{OT,q_H})\}$ be the list of hash value of the one time public keys.

Identification queries $\mathcal{O}_P$: On input a call, $\mathcal{F}$ simulates the prover as follows:

1. Obtain $\mathbb{T}$ from $\mathcal{A}_1$.
2. First, select random generators $\mathtt{T}', A_1', A_2', A_3' \xleftarrow{\$} \mathbb{G}_1$. Second, Choose integers $a', b' \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $E_1' = U^{a'}$; $E_2' = U^{b'}$. Let $m \xleftarrow{\$} LM$. Next, compute $\sigma = g_1^{1/(s+m)}$ with respect to the proof of Lemma 1 in [6]. Then compute $E_3' = \sigma W^{a+b}$. Obtain $pk_{OT} \in HM$, where $m = H(pk_{OT})$ and then select $\kappa' \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $\bar{m}' = H(\mathtt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$; $\bar{\sigma}' = g_a^{\frac{1}{\alpha+\kappa'\cdot\eta+\bar{m}'}}$. Finally, $\mathcal{O}_P$ returns $(\mathtt{T}', pk_{OT}, \bar{\sigma}', \kappa', E_1', E_2', E_3', A_1', A_2', A_3')$.
3. $\mathcal{A}_1$ replies with c, d.
4. $\mathcal{F}$ checks the validility of c, d with respect to $\mathbb{T}$. If not valid, $\mathcal{F}$ aborts the current execution. Otherwise, it rewinds $\mathcal{A}_1$ to the second step and then computes as follows:
   (a) $z_s, a, b, z_a, z_b \xleftarrow{\$} \mathbb{Z}_p^*$; $E_1 = U^a$; $E_2 = V^b$; $\mathtt{T} = S_P^c g_1^{z_s}$; $A_1 = E_1^c U^{z_a}$; $A_2 = E_2^c V^{z_b}$.
   (b) Set $E_3 = E_3'$ and compute $A_3 = \big(\frac{\hat{e}(E_3, S_P g_1^m)}{\hat{e}(g, g_1)}\big)^c \hat{e}(W, S_P g_1^m)^{z_a+z_b}$.
   (c) Select a random integer $\kappa \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $\bar{m} = H(\mathtt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$; $\bar{\sigma} = g_a^{\frac{1}{\alpha+\kappa\cdot\eta+\bar{m}}}$.
   (d) Finally, $\mathcal{O}_P$ returns $(\mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, E_1, E_2, E_3, A_1, A_2, A_3)$ to $\mathcal{A}$
5. $\mathcal{A}$ replies with c', d'.
6. Check the validation of c', d' with $\mathbb{T}$ and check whether $c = c'$ and $d = d'$. If the above does not hold, $\mathcal{F}$ aborts the current execution. Otherwise, it returns $\hat{Z} = (z_s, z_a, z_b)$ to $\mathcal{A}$.
7. Finally, $\mathcal{F}$ records a transcript tr and a signature σ.

Now, $\mathcal{F}$ runs the eimp-aa experiment with $\mathcal{A}$. First, in the *Learning Phase*, the entire parameter is first initialized. The public/secret key pair of provers is initially set to $pk_P = S_P$ and $sk_P = s$, Then the TA public/secret key pair is generated by running $(pk_T, sk_T) \leftarrow \mathtt{KeyGen}(1^k)$. In this phase, $\mathcal{A}$ plays a role of $\mathcal{A}_1$. $\mathcal{A}_1$ is given with pk_P, pk_T, sk_T and the access to $\mathcal{O}_P$. At the end of this phase, $\mathcal{A}_1$ outputs a state of information st and passes it to $\mathcal{A}_2$.

Now we move to the *Impersonation Phase*. On input st from the *Learning Phase*, $\mathcal{A}_2$ runs the identification protocol to convince V (played by $\mathcal{F}$) to accept $\mathcal{A}_2$ as P. Note that the public parameter for $\mathcal{A}_2$ can be obtained from $\mathcal{A}_1$ in the previous phase. $\mathcal{A}_2$ then interacts with $\mathcal{F}$ as the following protocol:

- $(\mathcal{F} \to \mathcal{A}_2)$ Select random integers $c, d \in \mathbb{Z}_p^*$ and compute $\mathbb{T} = g_1^c g_2^d$. $\mathcal{F}$ sends $\mathbb{T}$ to $\mathcal{A}_2$.
- $(\mathcal{A}_2 \to \mathcal{F})$ Reply with $(\mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, E_1, E_2, E_3, A_1, A_2, A_3)$.
- $(\mathcal{F} \to \mathcal{A}_2)$ Respond with c, d
- $(\mathcal{A}_2 \to \mathcal{F})$ Return $\hat{Z} = (z_s, z_a, z_b)$

$\mathcal{A}$ wins the game if a transcript $tr = (\mathbb{T}, \mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, E_1, E_2, E_3, A_1, A_2, A_3, c, d, \hat{Z})$ from above protocol passes the validation. Due to the fact that $\mathcal{F}$ possessed γ, β, which are secret keys to solve the relationship among g, g_1 and g_2, with a overwhelming probability, $\mathcal{F}$ then rewinds $\mathcal{A}_2$ to the third step and replies $\mathcal{A}_2$ with $c', d' \in \mathbb{Z}_p^*$ such that $c' \neq c$ and $d' \neq d$. $\mathcal{A}_2$ responds with $\hat{Z}' = (z_s', z_a', z_b')$ and let $tr_2 = (\mathbb{T}, \mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, E_1, E_2, E_3, A_1,\ A_2, A_3, c', d', \hat{Z}')$ denote the second transcript. From these two transcripts, $\mathcal{F}$ compute s as the answer to q-DL problem.

Next, we conclude the success probability that $\mathcal{A}$ successes the impersonation. There are two events that trigger $\mathcal{F}$ to abort. We will show that $\mathcal{F}$ will abort the simulation with negligible probability. These two events are in the steps 4 and 6 of the identification queries. The first event is that $c \neq c'; d \neq d'$ but $\mathbb{T} = g_1^c g_2^d = g_1^{c'} g_2^{d'}$. If this event occurs, then $\mathcal{A}_1$ can be used to solve the discrete logarithm problem where, on input g, g^x, find x. We simply set $g_1 = g; g_2 = g^x$ then, upon receiving c, d, c' and d', we compute $x = (c - c')/(d' - d)$. Hence, this event occurs with negligible probability underlying that the discrete logarithm problem is hard. The last event is when c, d is dishonestly generated and is consequently not valid. Since the correctness of $\mathbb{T} = g_1^c g_2^d$ holds with negligible probability of error with regard to the first event, we can conclude that if $\mathcal{A}_1$ is correctly interacting with the identification queries, then the second event will not occur. Therefore, we claim that $\mathcal{F}$ solves the q-DL problem with non-negligible probability by using $\mathcal{A}$.

To conclude, the above shows that if there exists an adversary that breaks the impersonation of an EDID scheme under active attack, then we can use this adversary to solve the q-DL problem with non-negligible probability. Conversely, if the q-DL problem holds, then the EDID scheme is secure against impersonation under active attack. □

5.3 Security Analysis for Transferability

Theorem 5. *Our identification scheme* EDID $= (\mathtt{Setup}, \mathtt{KeyGen}, P, V, TA, AnyV)$ *is secure against transferability attack if only the q-SDH problem hold under in the standard model.*

Proof. From the proof in Theorem 2 and Theorem 3, the identification protocol and the transfer protocol in our EDID scheme (computational) are zero knowledge protocol. Based on the above statement, we can exclude a PPT tran adversary that can distinguish the simulated transcripts out of the actual transcripts. We also exclude an adversary that, without help from the trusted third party, uses any mean (or any protocol) to convince another party that the adversary has actually interacted with the prover to generate a transcript. If there exists an adversary as described above then that adversary in fact seems to be an adversary against deniability where the proof has been provided in a proof of Theorem 2. Hence, the rest of the proof will show that, assumed that there exists a PPT tran adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a PPT algorithm $\mathcal{F}$ using $\mathcal{A}$ to solve the q-SDH problem. Start with $\mathcal{F}$ constructing queries and setting up public parameters as follow:

Parameter setup: First, $\mathcal{F}$ is given $g, g^s, g^{s^2}, \cdots, g^{s^q} \in \mathbb{G}_1$ as input. Then, $\mathcal{F}$ computes g_1 and S_P in the same way with the proof of Lemma 1 in [6]. $\mathcal{F}$ then sets $g = g_1^\gamma$ and $g_2 = g^\beta$, where $\gamma, \beta \xleftarrow{\$} \mathbb{Z}_p$. Next, $\mathcal{F}$ runs $\mathtt{KeyGen}^T$ to get the public key and private key of the trusted authority, which are $pk_T = (U, V, W)$ and $sk_T = (x, y)$ respectively. Let $OT = \{pk_{OT,1}, sk_{OT,1}, ...,$ $pK_{OT,q_H-1},\ sk_{OT,q_H-1},\ pk_{OT}^*, sk_{OT}^*\}$ be the list of pre-compute one-time public keys and secret keys. Let $LM = \{m_1 = H(pk_{OT,1}), ..., m_{q_H-1} = H(pk_{OT,q_H-1}), m^* = H(pk^*_{OT})\}$ be the list of hash value of the one time public keys.

Identification queries $\mathcal{O}_P$: For every request of transcript to $\mathcal{O}_P$ excepted when $m = m_*$, $\mathcal{F}$ constructs identification queries in the same way as the proof in Theorem 4. For a case of m_*, $\mathcal{F}$ changes the procedure of the identification queries and processes as follows:

1. Obtain $\mathbb{T}$ from $\mathcal{A}_1$.
2. First, select random generators $\mathtt{T}', A'_1, A'_2, A'_3, E'_3 \overset{\$}{\leftarrow} \mathbb{G}_1$. Second, choose integers $a', b' \overset{\$}{\leftarrow} \mathbb{Z}^*_p$ and compute $E'_1 = U^{a'}$; $E'_2 = U^{b'}$. Obtain $pk^*_{OT} \in HM$ and then select $\kappa' \overset{\$}{\leftarrow} \mathbb{Z}^*_p$ and compute $\bar{m}' = H(\mathtt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$; $\bar{\sigma}' = g_a^{\frac{1}{\alpha+\kappa'\cdot\eta+\bar{m}'}}$. Finally, $\mathcal{O}_P$ returns $(\mathtt{T}', pk_{OT}, \bar{\sigma}', \kappa', E'_1, E'_2, E'_3, A'_1, A'_2, A'_3)$.
3. $\mathcal{A}_1$ replies with c, d.
4. $\mathcal{F}$ checks the validility of c, d with respect to $\mathbb{T}$. If not valid, $\mathcal{F}$ aborts the current execution. Otherwise, it rewinds $\mathcal{A}_1$ to the second step and then computes as follows:
   (a) $z_s, a, b, z_a, z_b \overset{\$}{\leftarrow} \mathbb{Z}^*_p$; $E_3 \overset{\$}{\leftarrow} \mathbb{G}_1$
   $E_1 = U^a$; $E_2 = V^b$; $\mathtt{T} = S^c_P g_1^{z_s}$; $A_1 = E^c_1 U^{z_a}$; $A_2 = E^c_2 V^{z_b}$.
   (b) Compute $A_3 = \big(\frac{\hat{e}(E_3, S_P g_1^m)}{\hat{e}(g, g_1)}\big)^c \hat{e}(W, S_P g_1^m)^{z_a+z_b}$.
   (c) Select a random integer $\kappa \overset{\$}{\leftarrow} \mathbb{Z}^*_p$ and compute $\bar{m} = H(\mathtt{T}, \mathbb{T}, pk_P, E_1, E_2, E_3, A_1, A_2, A_3)$; $\bar{\sigma} = g_a^{\frac{1}{\alpha+\kappa\cdot\eta+\bar{m}}}$.
   (d) Finally, $\mathcal{O}_P$ returns $(\mathtt{T}, pk_{OT}, \bar{\sigma}, \kappa, E_1, E_2, E_3, A_1, A_2, A_3)$ to $\mathcal{A}$
5. $\mathcal{A}$ replies with c', d'.
6. Check the validation of c', d' with $\mathbb{T}$ and check whether $c = c'$ and $d = d'$. If the above does not hold, $\mathcal{F}$ aborts the current execution. Otherwise, it returns $\hat{Z} = (z_s, z_a, z_b)$ to $\mathcal{A}$.
7. Finally, $\mathcal{F}$ records a transcript tr^* and a signature σ^*.

Open queries $\mathcal{O}_O$: $\mathcal{F}$ constructs the open queries as follows:

- If tr is in the list of queried transcript then return an associated signature σ excepted $m = m^*$ return $\perp$.
- Otherwise, decrypt tr with sk_T and obtain σ and then check whether $\hat{e}(g, g_1) = \hat{e}(\sigma, S_P g_1^m)$. If yes then return σ. If not return $\perp$.

Let $\mathsf{param} = (\hat{e}, p, g, g_1, g_2)$ and $pk_S = S_P = g_1^s$. The private parameters for $\mathcal{F}$ are sk_{TA}, γ and β. Now, $\mathcal{F}$ simulates the *Learning Phase* by running $\mathcal{A}$ on input (param,pk_P, pk_T,$\mathcal{O}_P$, $\mathcal{O}_O$), and then operates as follows:

- On the request of tr, $\mathcal{A}_1$ runs the identification protocol with $\mathcal{O}_P$ to obtain tr.
- On the request to open tr, $\mathcal{A}_1$ arbitrarily sends tr to $\mathcal{O}_O$. $\mathcal{O}_O$ returns a signature σ on message m from tr, if tr and σ are valid and tr is generated by $\mathcal{O}_O$. Otherwise, it returns a failure.

At the end of this phase, $\mathcal{A}_1$ outputs a state of information st and passes it on to $\mathcal{A}_2$.
Now, move to the the *Convincing Phase*, where $\mathcal{F}$ plays a role as $AnyV$. Note that the public parameter for $\mathcal{A}_2$ can be obtained from $\mathcal{A}_1$ in the previous phase. $\mathcal{A}_2$ then interacts with $AnyV$ as the following protocol:

- $(AnyV \rightarrow \mathcal{A}_2)$ Select random integers $c, d \in \mathbb{Z}^*_p$ and compute $\mathfrak{T} = g_1^c g_2^d$. $\mathcal{F}$ sends $\mathfrak{T}$ to $\mathcal{A}_2$.
- $(\mathcal{A}_2 \rightarrow AnyV)$ Run $\{tr, D_1, D_2\} \leftarrow \mathcal{A}_2(st, pk_P, pk_T, \mathsf{param})$ and reply with (tr, D_1, D_2).
- $(AnyV \rightarrow \mathcal{A}_2)$ Respond with c, d
- $(\mathcal{A}_2 \rightarrow AnyV)$ Return $z \leftarrow \mathcal{A}_2(st, pk_P, pk_T, \mathsf{param}, tr, D_1, D_2, c, d)$;

$\mathcal{A}$ wins the game if a transfer transcript $tr_t = (\mathfrak{T}, tr, D_1, D_2, c, d, z)$ from above protocol passes the validation. Due to the fact that $\mathcal{F}$ possessed γ, β, which are secret keys to solve the relationship among g, g_1 and g_2, with a overwhelming probability, $\mathcal{F}$ then rewinds $\mathcal{A}_2$ to the third step and replies $\mathcal{A}_2$ with $c', d' \in \mathbb{Z}_p^*$ such that $c' \neq c$ and $d' \neq d$. Then $\mathcal{A}_2$ responds with z' and let $tr'_t = (\mathfrak{T}, tr, D_1, D_2, c', d', z')$ denote the second transfer transcript.

Let q_H, q_O be a number of queries that $\mathcal{A}$ makes to identification queries and open queries, respectively. Within the probability $\frac{1}{q_H}$, $\mathcal{A}$ processes the second phase with m_*. If $\mathcal{A}$ wins the game with m_*, then, from the above two transcripts, $\mathcal{F}$ computes a signature σ^* on message m_* as the answer to q-SDH problem.

There are certain events that cause $\mathcal{F}$ to abort the simulation. We will show that such events happen with negligible probability or that some are expected to occur with non-negligible probability. First, in the open queries, the first event(E_1) is that $\mathcal{F}$ aborts the simulation when $m = m_*$. This event is already expected to happen within the probability $(1-1/(q_H))^{q_O} \geq 1/e$ where e is the natural logarithm. The other event(E_2) is also in the open queries when tr is not in the list of transcript produced by $\mathcal{O}_P$ but it passes the verification. This means that $\mathcal{A}$ can produce a valid transcript. Then $\mathcal{A}$ can indeed be used to break the impersonation of our EDID scheme. Hence, from the proof in Theorem 4, the probability that $\mathcal{F}$ aborts in this event is negligible. For the events above, the probability that $\mathcal{F}$ does not abort the simulation is non-negligible, where $\Pr[E_1] + \Pr[E_2] \approx 1/e$. Hence, we claim that the probability that $\mathcal{A}$ wins the game is non-negligible if the probability of solving the q-SDH problem is non-negligible.

6 Conclusion

We introduced a new notion called *escrowed deniability* in an identification scheme. This notion bridges the gap between deniability and non-deniability in the identification scheme. We have also provided a concrete scheme that satisfies this new notion. The security of our identification scheme provides for both impersonation and transferability (*escrowed deniability*). Proof of these was also presented. In short, we believe the *escrowed deniability* property is an essential feature for identification schemes where the need for incorporation and disaffirmation is crucial.

References

[1] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre, *From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security.*, EUROCRYPT (Lars R. Knudsen, ed.), Lecture Notes in Computer Science, vol. 2332, Springer, 2002, pp. 418–433.

[2] N. Asokan, Victor Shoup, and Michael Waidner, *Optimistic fair exchange of digital signatures (extended abstract)*, EUROCRYPT, 1998, pp. 591–606.

[3] Mihir Bellare, Marc Fischlin, Shafi Goldwasser, and Silvio Micali, *Identification protocols secure against reset attacks.*, EUROCRYPT (Birgit Pfitzmann, ed.), Lecture Notes in Computer Science, vol. 2045, Springer, 2001, pp. 495–511.

[4] Mihir Bellare and Adriana Palacio, *GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks.*, CRYPTO (Moti Yung, ed.), Lecture Notes in Computer Science, vol. 2442, Springer, 2002, pp. 162–177.

[5] Dan Boneh, *The decision diffie-hellman problem*, ANTS (Joe Buhler, ed.), Lecture Notes in Computer Science, vol. 1423, Springer, 1998, pp. 48–63.

[6] Dan Boneh and Xavier Boyen, *Short signatures without random oracles.*, EUROCRYPT (Christian Cachin and Jan Camenisch, eds.), Lecture Notes in Computer Science, vol. 3027, Springer, 2004, pp. 56–73.

[7] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps*, EUROCRYPT (Eli Biham, ed.), Lecture Notes in Computer Science, vol. 2656, Springer, 2003, pp. 416–432.

[8] Sherman S. M. Chow, Willy Susilo, and Tsz Hon Yuen, *Escrowed linkability of ring signatures and its applications*, VIETCRYPT (Phong Q. Nguyen, ed.), Lecture Notes in Computer Science, vol. 4341, Springer, 2006, pp. 175–192.

[9] Cynthia Dwork, Moni Naor, and Amit Sahai, *Concurrent zero-knowledge*, STOC, 1998, pp. 409–418.

[10] Uriel Feige, Amos Fiat, and Adi Shamir, *Zero-knowledge proofs of identity.*, J. Cryptology **1** (1988), no. 2, 77–94.

[11] Amos Fiat and Adi Shamir, *How to prove yourself: Practical solutions to identification and signature problems.*, CRYPTO (Andrew M. Odlyzko, ed.), Lecture Notes in Computer Science, vol. 263, Springer, 1986, pp. 186–194.

[12] Louis C. Guillou and Jean-Jacques Quisquater, *A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory.*, EUROCRYPT, 1988, pp. 123–128.

[13] Qiong Huang, Duncan S. Wong, Jin Li, and Yiming Zhao, *Generic transformation from weakly to strongly unforgeable signatures*, J. Comput. Sci. Technol. **23** (2008), no. 2, 240–252.

[14] Qiong Huang, Guomin Yang, Duncan S. Wong, and Willy Susilo, *Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles*, RSA Conference 2008, Cryptographers' Track (CT-RSA 2008), Lecture Notes in Computer Science 4964 (2008), 106 – 120.

[15] Myungsun Kim and Kwangjo Kim, *A new identification scheme based on gap diffie-hellman problem.*, The 2002 Symposium on Cryptography and Information Security, 2002.

[16] Wenbo Mao, *Verifiable escrowed signature*, ACISP (Vijay Varadharajan, Josef Pieprzyk, and Yi Mu, eds.), Lecture Notes in Computer Science, vol. 1270, Springer, 1997, pp. 240–248.

[17] Kazuo Ohta and Tatsuaki Okamoto, *A modification of the fiat-shamir scheme.*, CRYPTO (Shafi Goldwasser, ed.), Lecture Notes in Computer Science, vol. 403, Springer, 1988, pp. 232–243.

[18] Tatsuaki Okamoto, *Provably secure and practical identification schemes and corresponding signature schemes.*, CRYPTO (Ernest F. Brickell, ed.), Lecture Notes in Computer Science, vol. 740, Springer, 1992, pp. 31–53.

[19] Rafael Pass, *On deniability in the common reference string and random oracle model*, CRYPTO (Dan Boneh, ed.), Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 316–337.

[20] Mario Di Raimondo and Rosario Gennaro, *New approaches for deniable authentication*, ACM Conference on Computer and Communications Security (Vijay Atluri, Catherine Meadows, and Ari Juels, eds.), ACM, 2005, pp. 112–121.

[21] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk, *Deniable authentication and key exchange*, ACM Conference on Computer and Communications Security (Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, eds.), ACM, 2006, pp. 400–409.

[22] Claus-Peter Schnorr, *Efficient identification and signatures for smart cards*, CRYPTO (Gilles Brassard, ed.), Lecture Notes in Computer Science, vol. 435, Springer, 1989, pp. 239–252.

[23] Victor Shoup, *On the security of a practical identification scheme.*, J. Cryptology **12** (1999), no. 4, 247–260.

[24] Pairat Thorncharoensri, Qiong Huang, Man Ho Au, Willy Susilo, Guomin Yang, Yi Mu, and Duncan S. Wong, *The need for deniability in identification schemes*, In Short Presentation Track in 9th International Workshop on Information Security Applications (WISA2008), September 2008.

[25] Gang Yao, Guilin Wang, and Yong Wang, *An improved identification scheme*, Coding, Cryptography and Combinatorics (Birkhauser Verlag, Basel, Switzerland), Progress in Computer Science and Applied Logic, vol. 23, Birkhauser Verlag, 2004.

Authors

Pairat Thorncharoensri received a master of computer science by coursework and a master of internet technology by coursework from Wollongong University in 2004 and 2003 respectively. He received a bachelor of electrical engineering from King Mongkut's Institute of Technology North Bangkok, Thailand in 1998. Currently, He is a PhD student in Wollongong University under the supervision of Dr. Willy Susilo and Dr. Yi Mu. His current research interests include network security, computer security, and cryptography.

Qiong Huang got his B.S. and M.S. degrees from Fudan University in 2003 and 2006. After graduation, he worked as a research associate in City University of Hong Kong. Now he is a PhD student in City University of Hong Kong, under the supervision of Dr. Duncan S. Wong. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis.

Willy Susilo obtained his Bachelor Degree in Computer Science from Universitas Surabaya, Indonesia with a "Summa Cum Laude" predicate. He received his Master Degree in Computer Science and Doctor of Philosophy from University of Wollongong in 1996 and 2001, resp. His main research interest include cryptography and computer security, in particular the design of signature schemes. He was appointed as a Professor and Head of School of Computer Science and Software Engineering (SCSSE) in 2009. Prior to this role, he was the deputy director of ICT Research Institute and the Academic Program Director for UoW (Singapore). He is the director of Centre for Computer and Information Security Research (CCISR).

Man Ho Au obtained his PhD from the Faculty of Informatics at University of Wollongong in the year 2009. His academic supervisors are Dr. Willy Susilo and Dr. Yi Mu. He is currently an associate research fellow at UoW. His research interests include public key cryptography and network security. In particular, he is interested in privacy-preserving cryptographic systems including anonymous credential systems and electronic cash systems.

Yi Mu received his PhD from the Australian National University in 1994. He currently is an associate professor in School of Computer Science and Software Engineering and the director of Centre for Computer and Information Security Research, University of Wollongong. Prior to joining University of Wollongong, he was a senior lecturer in the Department of Computing, Macquarie University. He also worked in Department of Computing and IT, University of Western Sydney as a lecturer. His current research interest includes cryptography, network security, electronic payment, access control, and computer security. He is Editor-in-Chief of International Journal of Applied Cryptography and serves as Editor or Guest Editor for many international Journals. He has served in program committees for a number of international security conferences He is a senior member of the IEEE and a member of the IACR.

Duncan Wong received his B.Eng. degree in Electrical & Electronic Engineering with first class honors from the University of Hong Kong in 1994, M.Phil. degree in Information Engineering from the Chinese University of Hong Kong in 1998 and Ph.D. degree in Computer Science from Northeastern University, Boston, MA, USA in 2002. After graduation, he has been a visiting assistant professor at the Chinese University of Hong Kong for one year before joining City University of Hong Kong in September 2003. He is now an assistant professor in the Department of Computer Science. His primary research interest is applied cryptography; in particular, cryptographic protocols, encryption and signature schemes, and anonymous systems.