1-2011

Identity-based strong designated verifier signature revisited

Qiong HUANG

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Duncan S. WONG

Willy SUSILO

Citation

HUANG, Qiong; YANG, Guomin; WONG, Duncan S.; and SUSILO, Willy. Identity-based strong designated verifier signature revisited. (2011). *Journal of Systems and Software*. 84, (1), 120-129.

Available at: https://ink.library.smu.edu.sg/sis_research/7353

The Journal of Systems and Software 84 (2011) 120–129

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Identity-based strong designated verifier signature revisited☆

Qiong Huang[a,*], Guomin Yang[b], Duncan S. Wong[a], Willy Susilo[c]

[a] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China
[b] Temasek Laboratories, National University of Singapore, Singapore
[c] School of Computer Science and Software Engineering, University of Wollongong, Northfields Avenue, Wollongong, Australia

ARTICLE INFO

Article history:
Received 2 February 2010
Received in revised form 18 July 2010
Accepted 29 August 2010
Available online 17 September 2010

Keywords:
Strong designated verifier signature
Non-delegatability
Non-transferability
Identity-based cryptography
Random oracle model
Signature scheme

ABSTRACT

Designated verifier signature (DVS) allows the signer to persuade a verifier the validity of a statement but prevent the verifier from transferring the conviction. Strong designated verifier signature (SDVS) is a variant of DVS, which only allows the verifier to privately check the validity of the signer's signature. In this work we observe that the unforgeability model considered in the existing identity-based SDVS schemes is not strong enough to capture practical attacks, and propose to consider another model which is shown to be strictly stronger than the old one. We then propose a new efficient construction of identity-based SDVS scheme, which is provably unforgeable under the newly proposed definition, based on the hardness of Computational Diffie–Hellman problem in the random oracle model. Our scheme is perfectly non-transferable in the sense that the signer and the designated verifier can produce identically distributed signatures on the same message. Besides, it is the *first* IBSDVS scheme that is *non-delegatable* with respect to (an identity-based variant of) the definition proposed by Lipmaa et al. (ICALP 2005).

1. Introduction

In Eurocrypt 1996, Jakobsson, Sako and Impagliazzo (Jakobsson et al., 1996) proposed the notion of designated verifier signature (DVS in short), which allows Alice, the signer, to prove the validity of a statement Θ to Bob, the verifier, in such a way that Bob is convinced about the validity of Θ, but different from ordinary signature schemes, he could not transfer the conviction to any third party. This is called *non-transferability*, which requires Bob being able to produce signatures that are indistinguishable from those generated by Alice. After receiving a signature on a message from Alice, Bob is ensured that Alice made the signature as he did not sign the message.

DVS allows any third party to tell from a signature that either Alice or Bob generated it. Let us consider the example given in Saeednia et al. (2003). A public institution initiates a call for tenders, asking some companies to propose their prices for a set of devices. The institution may require each of the companies to sign their offers. However, no company desires its offer to affect others' decisions, because a company may capture a competitor's signed offer on the transmission line to the initiator and prepares its offer accordingly to increase its chance of being selected. For this purpose, DVS can come to help. However, the signature could tell the company which captures it on the line before getting to the initiator that who the signer is, since now it is sure that the institution did not produce this signature. This problem was addressed by Jakobsson et al. (1996). They proposed the notion of *strong designated verifier signature* (SDVS), which allows only the designated verifier to privately verify a signature from the signer. Anyone who does not have the verifier's secret key could not tell who produced this signature. This notion has not been formalized until Laguillaumie et al.'s work, where the property *privacy of the signer* (PSI) is defined (Laguillaumie and Vergnaud, 2004).

Identity-based cryptography was introduced by Shamir (1984), aiming to solve the problems of certificate management in public key infrastrucute (PKI). In an identity-based scheme the public key of a user can be a string related its identity, such as email address, IP address and etc. There is a trusted party, named *public key generator* (PKG), which is responsible for the generation of user secret keys according to the user's identity. Anyone can secretly send a message to a user by encrypting the message under the user's identity, even before the user obtains its secret key from the PKG. In this paper we focus on SDVS in the identity-based setting. That is, we study *identity-based strong designated verifier signature* (IBSDVS).

1.1. Related work

Since the introduction of DVS, it (and its variants) have attracted the attention of many researchers. Steinfeld et al. (2003) proposed an extension of DVS, called *universal designated verifier signature*

☆ This work was supported by a grant from CityU (Project No. 7002585).

* Corresponding author.
E-mail addresses: csqhuang@gmail.com (Q. Huang), tslyg@nus.edu.sg (G. Yang), duncan@cityu.edu.hk (D.S. Wong), wsusilo@uow.edu.au (W. Susilo).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.08.057

(UDVS), in which the holder of a signature can designate any third party as the verifier for verifying the signature so that the verifier is convinced of that the holder indeed holds the signer's signature but in the meanwhile it cannot transfer this conviction to others. Many UDVS schemes have been proposed since then, e.g. Steinfeld et al. (2004), Zhang et al. (2005), Laguillaumie et al. (2006), Huang et al. (2006, 2007), and Vergnaud (2006). Later, Baek et al. (2005) proposed the notion of *universal designated verifier proof* to eliminate the verifier's need of creating and registering a public key.

Susilo et al. (2004) proposed an efficient IBSDVS scheme, whose unforgeability is based on Bilinear Diffie–Hellman (BDH) assumption. Huang et al. (2008) also proposed an IBSDVS scheme based on Diffie–Hellman key exchange, which has very short signatures, i.e. the signature is the hash value of the common secret key shared between the signer and the designated verifier. The security of their scheme relies on a stronger assumption, i.e. Gap Bilinear Diffie–Hellman (GBDH). Recently, Kang et al. (2009) proposed another IBSDVS scheme which is secure based BDH assumption. Security proofs of all these schemes are done in the random oracle model (Bellare and Rogaway, 1993).

Lipmaa, Wang and Bao considered a new type of attacks against DVS schemes, named *delegatability attacks*, in which Alice or Bob could release a derivative of their (common) secret key to any third party say Ted, so that Ted can use this derivative to produce signatures on any message for Bob on behalf of Alice. They proposed the notion of *non-delegatability*, which basically requires that if one produces a valid signature with respect to Alice and Bob, it must 'know' the secret key of either Alice or Bob. Many DVS schemes and variants have been shown to be vulnerable to delegatability attacks in Lipmaa et al. (2005). In the same year, Li et al. (2005) analyzed the security of other four designated verifier signature schemes and showed that they are also deletagable. Besides those schemes, it is also easy to show that the identity-based schemes proposed in Huang et al. (2008), Cao and Cao (2009), and Kang et al. (2009) are also vulnerable to this kind of attacks.

Recently, Zhang and Mao (2008) proposed an IBSDVS scheme which to the best of our knowledge, is the first one in the identity-based setting that is claimed to be non-delegatable. However, the proof of non-delegatability of their scheme does not strictly follow the definition proposed by Lipmaa et al. (2005). It is unknown if there is an algorithm which can extract the secret key of either Alice or Bob, given black-box oracle access to such a forger.

Laguillaumie and Vergnaud (2004) formalized the motivation for the introduction of SDVS, and proposed the notion of *privacy of signer's identity*, which says that if one does not have the verifier's secret key, it cannot distinguish the signatures produced by signer Alice for verifier Cindy from those by signer Bob for Cindy.

1.2. Our contributions

Our contributions in this paper are in twofold. First, we show that the model of unforgeability of IBSDVS considered in the existing work such as Susilo et al. (2004); Zhang and Mao (2008); Kang et al. (2009) is not strong enough to capture practical attacks. We propose to consider another model of unforgeability, and demonstrate that this model is strictly stronger than the old one by giving an example that is secure under the old model but insecure under the new one.

Second, we propose another efficient IBSDVS scheme, which is based on Gentry–Silverberg hierarchical identity-based encryption scheme (Gentry and Silverberg, 2002) and makes use of the standard technique of non-interactive zero-knowledge proof of knowledge obtained via Fiat–Shamir heuristic. Thus our scheme is secure in the random oracle model. The purpose of using a proof of knowledge is to obtain the non-delegatability. Though our scheme does not outperform other schemes such as Susilo et al. (2004); Zhang and Mao (2008); Huang et al. (2008); Kang et al. (2009), in terms of signature size, it is provably unforgeable with respect to the stronger model. The underlying assumption of unforgeability of our scheme is the widely studied CDH assumption, which is weaker than those assumptions used in Susilo et al. (2004); Zhang and Mao (2008); Huang et al. (2008); Kang et al. (2009). Moreover, our scheme is *non-delegatable*, and the proof follows the definition proposed by Lipmaa et al. (2005), i.e. there is an extractor which, given a forger algorithm, can extract the secret key of either the signer or the verifier in the black-box manner. Our construction of IBDVS also enjoys perfectly non-transferability in the sense that the signer's signatures can be perfectly simulated by the designated verifier. In addition, we show that our scheme supports the privacy of signer's identity as defined in Laguillaumie and Vergnaud (2004); Huang et al. (2008). As discussed in Sections 2.4 and 6.2, the privacy of signer's identity of our scheme is stronger than that of Huang et al.'s scheme Huang et al. (2008).

1.3. Paper organization

In the next section we give the definition of IBSDVS and its security model. Some mathematical background is given in Section 3. We show in Section 4 that there is a gap between the old definition of unforgeability and the new one considered here. Our IBSDVS scheme is proposed in Section 5. We also prove its security with respect to the given security definitions in the random oracle in Section 6, along with a comparison between our scheme and other existing schemes. The paper is concluded in Section 7.

2. Identity-based strong designated verifier signature

A strong designated verifier signature scheme (SDVS) Jakobsson et al. (1996) consists of four (probabilistic) polynomial-time algorithms, one for key generation, one for the signer to sign for a designated verifier, one for the designated verifier to simulate the signer's signature, and the other for the designated verifier to check the validity of a signature. Identity-based strong designated verifier signature (IBSDVS) is the analogy of SDVS in the identity-based setting. Below is the formal definition of it.

Definition 2.1 *(IBSDVS).* An identity-based strong designated verifier signature scheme consists of five (probabilistic) polynomial-time algorithms, described as below:

- Setup: The algorithm takes as input 1^k where k is the security parameter, and outputs a master key pair for the PKG, i.e. $(\mathtt{m}pk, \mathtt{m}sk) \leftarrow \mathsf{Setup}(1^k)$, where $\mathtt{m}pk$ is published, and $\mathtt{m}sk$ is kept secret by the PKG.
- Extract: The algorithm takes as input the master secret key $\mathtt{m}sk$ and an identity $\mathtt{i}d$ which can be a string of arbitrary length, and outputs the corresponding secret key $\mathtt{u}sk_{\mathtt{id}}$ for the user with identity $\mathtt{i}d$, i.e. $\mathtt{u}sk_{\mathtt{id}} \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{i}d)$.
- Sign: The algorithm takes as input the secret key of the signer $\mathtt{u}sk_S$, the identities of the signer and the designated verifier, i.e. $\mathtt{i}d_S$ and $\mathtt{i}d_V$, the master public key $\mathtt{m}pk$ and a message $M \in \{0, 1\}^*$, and outputs a signature σ, i.e. $\sigma \leftarrow \mathsf{Sign}(\mathtt{u}sk_S, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{m}pk, M)$.
- Ver: The algorithm takes as input a message M, the identities of the signer and the verifier, i.e. $\mathtt{i}d_S$ and $\mathtt{i}d_V$, the verifier's secret key $\mathtt{u}sk_V$, the master public key $\mathtt{m}pk$ and a purported signature σ, and outputs a bit b, which is 1 for acceptance or 0 for rejection, i.e. $b \leftarrow \mathsf{Ver}(M, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{u}sk_V, \mathtt{m}pk, \sigma)$.
- Sim: The algorithm takes as input the secret key of the verifier $\mathtt{u}sk_V$, the identities of the signer and the designated verifier, i.e. $\mathtt{i}d_S$ and $\mathtt{i}d_V$, the master public key $\mathtt{m}pk$ and a message M, and outputs a signature σ, i.e. $\sigma \leftarrow \mathsf{Sim}(\mathtt{u}sk_V, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{m}pk, M)$.

The *completeness* requires that for any $(\mathtt{m}pk, \mathtt{m}sk) \leftarrow \mathsf{Setup}(1^k)$, any $\mathtt{id}_S, \mathtt{id}_V \in \{0, 1\}^*$, $\mathtt{u}sk_S \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_S)$, $\mathtt{u}sk_V \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_V)$, any message $M \in \{0, 1\}^*$, let $\sigma \leftarrow \mathsf{Sign}(\mathtt{u}sk_S, \mathtt{id}_S, \mathtt{id}_V, \mathtt{m}pk, M)$ and $\sigma' \leftarrow \mathsf{Sim}(\mathtt{u}sk_V, \mathtt{id}_S, \mathtt{id}_V, \mathtt{m}pk, M)$, it holds that

$$\Pr[\mathsf{Ver}(M, \mathtt{id}_S, \mathtt{id}_V, \mathtt{usk}_V, \mathtt{mpk}, \sigma) = 1] = 1, \quad \text{and}$$
$$\Pr[\mathsf{Ver}(M, \mathtt{id}_S, \mathtt{id}_V, \mathtt{usk}_V, \mathtt{mpk}, \sigma') = 1] = 1$$

where the probabilities are taken over the random coins used in Setup, Extract, Sign and Sim, and the random choices of $\mathtt{id}_S$, $\mathtt{id}_V$ and M.

Besides the completeness, a non-delegatable IBSDVS scheme should additionally satisfy *unforgeability*, *non-transferability*, *privacy of signer's identity* and *non-delegatability*, which are defined in the following.

2.1. Unforgeability

Roughly speaking, unforgeability requires that any third party other than the signer and the designated verifier, cannot forge a signature on behalf of the signer with non-negligible probability. Formally, it is defined by the following game played between a game challenger C and a probabilistic polynomial-time adversary $\mathcal{A}$:

1. C generates a master key pair $(\mathtt{m}pk, \mathtt{m}sk)$, and invokes $\mathcal{A}$ on input $\mathtt{m}pk$.
2. The adversary can issue queries to the following oracles adaptively for polynomially many times:
   - $\mathcal{O}_{\mathsf{E}}$: Given a query $\mathtt{i}d$ from $\mathcal{A}$, the oracle computes $\mathtt{u}sk_{\mathtt{i}d} \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{i}d)$, and returns $\mathtt{u}sk_{\mathtt{i}d}$ to $\mathcal{A}$.
   - $\mathcal{O}_{\mathsf{Sign}}$: Given a query of the form $(\mathtt{id}_S, \mathtt{id}_V, M)$, the oracle returns a signature σ on M valid with respect to $\mathtt{id}_S$ and $\mathtt{id}_V$ back to $\mathcal{A}$.
   - $\mathcal{O}_{\mathsf{Sim}}$: Given a query of the form $(\mathtt{id}_S, \mathtt{id}_V, M)$, the oracle returns a signature σ on M valid with respect to $\mathtt{id}_S$ and $\mathtt{id}_V$ back to $\mathcal{A}$.
   - $\mathcal{O}_{\mathsf{Ver}}$: Given a query of the form $(\mathtt{id}_S, \mathtt{id}_V, M, \sigma)$, the oracle returns a bit b which is 1 if σ is a valid signature on M with respect to the signer $\mathtt{id}_S$ and the designated verifier $\mathtt{id}_V$, and 0 otherwise.
3. Finally, $\mathcal{A}$ outputs its forgery, $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, \sigma^*)$. It wins the game if
   (a) $1 \leftarrow \mathsf{Ver}(M^*, \mathtt{id}_S^*, \mathtt{id}_V^*, \mathtt{usk}_{V^*}, \mathtt{mpk}, \sigma^*)$;
   (b) $\mathcal{A}$ did not query $\mathcal{O}_{\mathsf{E}}$ on input $\mathtt{id}_S^*$ and $\mathtt{id}_V^*$; and
   (c) $\mathcal{A}$ did not query $\mathcal{O}_{\mathsf{Sign}}$ and $\mathcal{O}_{\mathsf{Sim}}$ on input $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*)$ and $(\mathtt{id}_V^*, \mathtt{id}_S^*, M^*)$.

Definition 2.2 *(Unforgeability).* An IBSDVS scheme is said to be $(t, q_{\mathsf{E}}, q_{\mathsf{Sign}}, q_{\mathsf{Sim}}, q_{\mathsf{Ver}}, \epsilon)$-unforgeable if there is no adversary $\mathcal{A}$ which runs in time at most t, issues at most q_{E} queries to $\mathcal{O}_{\mathsf{E}}$, at most q_{Sign} queries to $\mathcal{O}_{\mathsf{Sign}}$, at most q_{Sim} queries to $\mathcal{O}_{\mathsf{Sim}}$, and at most q_{Ver} queries to $\mathcal{O}_{\mathsf{Ver}}$, and wins the game with probability at least ϵ.

Remark 1: In the model of unforgeability of IBSDVS scheme considered in previous work such as Zhang and Mao (2008), Susilo et al. (2004), and Kang et al. (2009) there is a restriction that the adversary is not allowed to ask the target signer and verifier to sign any message or verify the validity of a signature. Thanks to the restriction, the old model of unforgeability is not strong enough to capture practical attacks in which the adversary asks the signer to sign a message for a verifier, or it eavesdrops the communication between the signer and the verifier. In Section 4 we show that the model of unforgeability we consider here is strictly stronger than the old one by giving an example scheme that is secure under the old model but insecure under ours.

2.2. Non-transferability

Non-transferability says that given a message-signature pair (M, σ) which is accepted by the designated verifier, it is infeasible for any probabilistic polynomial-time distinguisher to tell whether the message was signed by the signer or the designated verifier. Formally, we consider the following definition.

Definition 2.3 *(Non-Transferability).* An IBSDVS scheme is *non-transferable* if the signature output by the signer is *computationally indistinguishable* from that output by the designated verifier, i.e.

$$\{\mathsf{Sign}(\mathtt{usk}_S, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, M)\} \approx \{\mathsf{Sim}(\mathtt{usk}_V, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, M)\}$$

where the two ensembles are indexed by $(\mathtt{id}_S, \mathtt{id}_V, M)$. That is, for any probabilistic polynomial-time distinguisher $\mathcal{D}$, for any $(\mathtt{m}pk, \mathtt{m}sk) \leftarrow \mathsf{Setup}(1^k)$, any identities $\mathtt{id}_S, \mathtt{id}_V \in \{0, 1\}^*$, any message $M \in \{0, 1\}^*$, let $\mathtt{u}sk_S \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_S)$ and $\mathtt{u}sk_V \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_V)$, it holds that

$$\left| \Pr\left[\begin{array}{c} \sigma_0 \leftarrow \mathsf{Sign}(\mathtt{usk}_S, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, M), \sigma_1 \leftarrow \mathsf{Sim}(\mathtt{usk}_V, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, M) \\ b \leftarrow_{\$} \{0, 1\}, b' \leftarrow \mathcal{D}(\mathtt{mpk}, \mathtt{msk}, \mathtt{id}_S, \mathtt{id}_V, \sigma_b) \end{array} : b' = b \right] - \frac{1}{2} \right| < \epsilon(k)$$

where $\epsilon(k)$ is a negligible function[1] in the security parameter k, and the probability is taken over the randomness used in Setup, Extract, Sign and Sim, and the random coins consumed by $\mathcal{D}$.

If the two distributions are identical, we say that the IBSDVS scheme is *perfectly non-transferable*.

Remark 2: The definition of non-transferability above is actually very strong, in the sense that even the trusted authority (the PKG) cannot tell correctly that a signature is from the signer or from the designated verifier, with a probability non-negligibly larger than one-half. One can also define a weaker version of non-transferability, by restricting the distinguisher from obtaining the master secret key.

2.3. Non-delegatability

Intuitively, non-delegatability requires that to generate a valid signature on a message, one has to 'know' the secret key of the signer or the designated verifier. Formally, we consider the following definition, which is an extension of the definition given in Lipmaa et al. (2005) to the identity-based setting.

Definition 2.4 *(Non-delegatability).* Let $\kappa \in [0, 1]$ be the knowledge error. An IBSDVS scheme is (t, κ)-non-delegatable if there exists a black-box knowledge extractor $\mathcal{K}$ that, for every algorithm $\mathcal{F}$ that runs in time at most t, satisfies the following condition:

> For every $(\mathtt{m}pk, \mathtt{m}sk) \leftarrow \mathsf{Setup}(1^k)$, every $\mathtt{id}_S, \mathtt{id}_V \in \{0, 1\}^*$, every $\mathtt{u}sk_S \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_S)$, $\mathtt{u}sk_V \leftarrow \mathsf{Extract}(\mathtt{m}sk, \mathtt{id}_V)$, and every message $M \in \{0, 1\}^*$, if $\mathcal{F}$ produces a valid signature on M with respect to $\mathtt{id}_S, \mathtt{id}_V$ with probability $\epsilon > \kappa$, (denote this specific algorithm by $\mathcal{F}_{S,V,M}$), then on input M and on oracle access to $\mathcal{F}_{S,V,M}$, $\mathcal{K}$ produces either $\mathtt{u}sk_S$ or $\mathtt{u}sk_V$ in expected time $t/(\epsilon - \kappa)$, without counting the time to make oracle queries. Note that the probability of $\mathcal{F}$ is taken over the choice of its random coins and the choices of the random oracles.

[1] A function $f : \mathbb{N} \rightarrow \mathbb{N}$ is *negligible* in the security parameter k if for every polynomial $q(\cdot)$, there exists some $K \in \mathbb{N}$ such that for every $k > K$, $f(k) < 1/q(k)$.

2.4. Privacy of signer's identity

Privacy of signer's identity, first defined by Laguillaumie and Vergnaud (2004), is the formalization of the motivation for the introduction of strong designated verifier signature. Basically, it says that one cannot distinguish Alice's signature for Bob from Cindy's signature for Bob, if the distinguisher does not know Bob's secret key. Below we give a formal definition of privacy of signer's identity of IBSDVS schemes. Consider the following game played between the challenger C and a distinguisher $\mathcal{D}$.

1. C generates the master key pair $(\mathtt{m}pk, \mathtt{m}sk)$ and invokes $\mathcal{D}$ on input $\mathtt{m}pk$.
2. $\mathcal{D}$ issues queries adaptively for polynomially many times as in the unforgeability game.
3. $\mathcal{D}$ outputs two signer identities $\mathtt{id}^*_{S_0}, \mathtt{id}^*_{S_1}$ and a verifier identity $\mathtt{id}^*_V$, along with a message M^*. C then tosses a coin $b \in \{0, 1\}$, computes $\mathtt{usk}_{S^*_b} \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}^*_{S_b})$, and signs the message by $\sigma^* \leftarrow \mathsf{Sign}(\mathtt{usk}_{S^*_b}, \mathtt{id}^*_{S_b}, \mathtt{id}^*_V, \mathtt{mpk}, M^*)$. It returns σ^* back to $\mathcal{D}$.
4. $\mathcal{D}$ continues to issue queries as in Step 2. Finally it outputs b' and wins the game if $b' = b$ and
   (a) $\mathcal{D}$ did not query $\mathcal{O}_{\mathsf{E}}$ on input $\mathtt{id}^*_V$;
   (b) $\mathcal{D}$ did not query $\mathcal{O}_{\mathsf{Ver}}$ on input $(\mathtt{id}^*_{S_d}, \mathtt{id}^*_V, M^*, \sigma^*)$ for any $d \in \{0, 1\}$.

Definition 2.5 *(Privacy of signer's identity).* An IBSDVS scheme is said to be $(t, q_{\mathsf{E}}, q_{\mathsf{Sign}}, q_{\mathsf{Sim}}, q_{\mathsf{Ver}}, \epsilon)$*-PSI-secure* if there is no adversary $\mathcal{D}$ which runs in time at most t, issues at most q_{E} queries to $\mathcal{O}_{\mathsf{E}}$, at most q_{Sign} queries to $\mathcal{O}_{\mathsf{Sign}}$, at most q_{Sim} queries to $\mathcal{O}_{\mathsf{Sim}}$, and at most q_{Ver} queries to $\mathcal{O}_{\mathsf{Ver}}$, and wins the game above with probability that deviates from one-half by more than ϵ.

Remark 3: Our definition of privacy of signer's identity is actually stronger than the one considered in Huang et al. (2008). In their definition, the adversary cannot ask oracle $\mathcal{O}_{\mathsf{E}}$ for secret keys of $\mathtt{id}^*_{S_0}$ and $\mathtt{id}^*_{S_1}$, nor ask $\mathcal{O}_{\mathsf{Sign}}$ and $\mathcal{O}_{\mathsf{Sim}}$ on input $(\mathtt{id}^*_{S_d}, \mathtt{id}^*_V, M^*)$ and $(\mathtt{id}^*_V, \mathtt{id}^*_{S_d}, M^*)$ for any $d \in \{0, 1\}$; while in our definition it is allowed to do so. They have the restriction mainly because their signing algorithm is deterministic, and there is only one possible signature for each tuple $(\mathtt{i}d_S, \mathtt{i}d_V, M)$. While our signing algorithm is randomized, there are exponentially many possible signatures for each identity-message tuple.

Remark 4: We stress that if the IBSDVS scheme is provably secure in the random oracle model, all the adversaries in games of unforgeability, non-transferability, non-delegatability and privacy of signer's identity have access to the random oracles. The definitions of these security properties are modified accordingly to take into account the numbers of queries to the random oracles issued by the adversaries.

3. Mathematical background

In this section we review some number-theoretic assumptions that will be used in the proofs of our IBSDVS scheme.

(Admissible pairings): Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of large prime order p. The mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be an *admissible pairing*, if (1. *bilinearity*) $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$; (2. *non-degeneracy*) $\exists u, v \in \mathbb{G}$ such that $\hat{e}(u, v) \neq 1_T$, where 1_T is the identity element of $\mathbb{G}_T$; and (3. *computability*). there exists an efficient algorithm for computing $\hat{e}(u, v)$ for any $u, v \in \mathbb{G}$.

(CDH assumption): Let $\mathbb{G}$ be a cyclic group of prime order p, and g be a random generator of $\mathbb{G}$.

Definition 3.1 *(CDH assumption).* We say that the *CDH* assumption (t, ϵ) holds in $\mathbb{G}$ if there is no adversary $\mathcal{A}$ that runs in time at most t and

$$\Pr\left[a, b \leftarrow_{\$} \mathbb{Z}_p, Z \leftarrow \mathcal{A}(g, g^a, g^b) : Z = g^{ab}\right] > \epsilon$$

where the probability is taken over the random choices of $a, b \in \mathbb{G}$ and random coins consumed by $\mathcal{A}$.

(DBDH assumption): Let $\mathbb{G}, \mathbb{G}_T, \hat{e}$ be defined as above, and g be a random generator of $\mathbb{G}$.

Definition 3.2 *(DBDH assumption).* We say that the *DBDH* assumption (t, ϵ)-holds in the bilinear setting $(\mathbb{G}, \mathbb{G}_T, p, g, \hat{e})$, if there is no adversary $\mathcal{A}$ that runs in time at most t and

$$\Pr\left[a, b, c \leftarrow_{\$} \mathbb{Z}_p, Z_0 \leftarrow_{\$} \mathbb{G}_T, Z_1 \leftarrow \hat{e}(g, g)^{abc}, d \leftarrow_{\$} \{0, 1\}, d' \leftarrow \mathcal{A}(g, g^a, g^b, g^c, Z_d) : d' = d\right] > \epsilon$$

where the probability is taken over the random choices of $a, b, c \in \mathbb{Z}_p$, $d \in \{0, 1\}$, and $Z_0 \in \mathbb{G}_T$, and the random coins consumed by $\mathcal{A}$.

4. A gap between the unforgeability models

In previous work on IBSDVS, i.e. Zhang and Mao (2008); Kang et al. (2009); Susilo et al. (2004), it is commonly required in the game of unforgeability that the adversary cannot ask the oracles $\mathcal{O}_{\mathsf{Sign}}$ and $\mathcal{O}_{\mathsf{Sim}}$ for a signature on any message with respect to the target signer $\mathtt{id}^*_S$ and the target verifier $\mathtt{id}^*_V$. The proofs of unforgeability of these schemes heavily rely on this restriction. Specifically, they base the unforgeability of the schemes on the Bilinear Diffie–Hellman assumption.[2] The common method in the security reductions is to set the master public key to be $g_1 = g^a$, and control the output of the random oracles so that $\mathtt{usk}_{S^*} = \mathtt{H}(\mathtt{id}^*_S)^a = g^{ab}$ and $\mathtt{usk}_{V^*} = \mathtt{H}(\mathtt{id}^*_V)^a = g^{ac}$. The generation/verification of a signature will involve the computation of the common secret key shared between the signer and the verifier, i.e. $\hat{e}(\mathtt{u}sk_S, \mathtt{H}(\mathtt{i}d_V)) = \hat{e}(\mathtt{H}(\mathtt{i}d_S), \mathtt{u}sk_V)$. If the adversary succeeds in forging a signature with respect to $\mathtt{id}^*_S$ and $\mathtt{id}^*_V$, the BDH problem solver then can find the solution to the given BDH problem by computing the common secret key shared between $\mathtt{id}^*_S$ and $\mathtt{id}^*_V$ from the forged signature. Since the problem solver does not know $\hat{e}(g, g)^{abc}$ during the simulation, it could not sign any message nor verify the validity of a signature with respect to $\mathtt{id}^*_S$ and $\mathtt{id}^*_V$. That is where the restriction in the old unforgeability game comes from. Such a restriction reduces the level of unforgeability of IBSDVS schemes significantly. In this section we show that our new definition of unforgeability is strictly stronger than the old one considered in Zhang and Mao (2008); Kang et al. (2009); Susilo et al. (2004). Namely, we give an IBSDVS scheme that is secure under the old definition (with the restriction) but insecure under the new one. Here we take Kang–Boyd–Dawson IBSDVS scheme Kang et al. (2009) as an example. Before any further discussion, we briefly review their scheme as below:

Kang–Boyd–Dawson IBSDVS Scheme: Let $(\mathbb{G}, \mathbb{G}_T, g, p, \hat{e})$ be defined as in Section 3. Let $\mathtt{H}_1 : \{0, 1\}^* \rightarrow \mathbb{G}$ and $\mathtt{H}_2 : \{0, 1\}^* \rightarrow \mathbb{Z}^*_p$ be two cryptographic hash functions. The public key of the PKG is $(\mathbb{G}, \mathbb{G}_T, g, p, \hat{e}, \mathtt{H}_1, \mathtt{H}_2, g_1 = g^{\alpha})$, where $\alpha \leftarrow_{\$} \mathbb{Z}_p$ is the master secret key. The secret key of each identity $\mathtt{i}d$ is $\mathtt{u}sk_{\mathtt{id}} = \mathtt{H}_1(\mathtt{i}d)^{\alpha}$. To sign a message M for a designated verifier with identity $\mathtt{i}d_V$, the signer

[2] Briefly, the Bilinear Diffie–Hellman assumption states that given g, g^a, g^b, g^c for some random $a, b, c \in \mathbb{Z}_p$, it is infeasible to find $\hat{e}(g, g)^{abc}$.

with identity $\mathtt{i}d_S$ chooses at random $k \in \mathbb{Z}_p^*$ and computes

$$t = \hat{e}(g, \mathtt{H}_1(\mathtt{id}_V))^k, \quad T = g^k \cdot \mathtt{usk}_{\mathtt{id}_S}^{\mathtt{H}_2(M,t)}, \quad \sigma = \hat{e}(T, \mathtt{H}_1(\mathtt{id}_V)).$$

The signature on M is (σ, t), which can be verified by checking if

$$\sigma \overset{?}{=} t \cdot \hat{e}(\mathtt{H}_1(\mathtt{id}_S), \mathtt{usk}_{\mathtt{id}_V})^{\mathtt{H}_2(M,t)} \tag{1}$$

The designated verifier can simulate the signature by choosing at random $k \in \mathbb{Z}_p^*$, and computes

$$\hat{t} = \hat{e}(g, \mathtt{H}_1(\mathtt{id}_V))^k \quad \text{and} \quad \hat{\sigma} = \hat{t} \cdot \hat{e}(\mathtt{H}_1(\mathtt{id}_S), \mathtt{usk}_{\mathtt{id}_V})^{\mathtt{H}_2(M,\hat{t})}.$$

It was shown in Kang et al. (2009) that the scheme is unforgeable with the aforementioned restriction, based on the hardness of BDH problem.

The attack: If we consider the strengthened model of unforgeability (Def. 2.2), it is easy to show that Kang–Boyd–Dawson IBSDVS Scheme is forgeable. Consider Eq. (1) in the verification of a signature (σ, t). An adversary against the unforgeability can forge a signature on any message as follows:

1. Choose at random a message M from the message space and the target identities $\mathtt{id}_S^*$ and $\mathtt{id}_V^*$, and ask the oracle $\mathcal{O}_{\mathsf{Sign}}$ to sign M with respect to $\mathtt{id}_S^*$, $\mathtt{id}_V^*$, which returns a valid signature (σ, t).
2. Compute $SK_{S^*,V^*} = (\sigma/t)^{\mathtt{H}_2(M,t)^{-1}}$, which is the common secret key shared between $\mathtt{id}_S^*$ and $\mathtt{id}_V^*$. Since the group is of prime order p which is public, the computation of SK_{S^*,V^*} is feasible.
3. To sign a message M', choose at random $t' \in \mathbb{G}_T$ and compute $\sigma' = t' \cdot SK_{S^*,V^*}^{\mathtt{H}_2(M',t')}$. The signature on M' with respect to $\mathtt{id}_S^*$ and $\mathtt{id}_V^*$ is (σ', t').

The attack above can also be easily modified so that after eavesdropping only one signature on any message (not chosen by the adversary), the adversary then forges signatures on any message for the verifier on behalf of the signer. This attack shows that there is a big gap between the unforgeability of IBSDVS schemes with the aforementioned restriction and that defined in Def. 2.2. In the next section we propose an IBSDVS scheme that is provably unforgeable under Def. 2.2.

5. Our non-delegatable IBSDVS

5.1. The scheme

Our IBSDVS scheme is based on Gentry–Silverberg hierarchical identity-based encryption scheme Gentry and Silverberg (2002). Different from previous IBSDVS schemes, the generation of a signature in our scheme does not involve the computation of the common secret key shared between the signer and the verifier. Instead, the signer uses its own secret key to produce an identity-based designated verifier signature, and then somehow 'encrypts' the signature for the designated verifier so that it leaves a trapdoor for the simulator of unforgeability game to generate (or verify) a signature even without the secret keys of the signer and the verifier (in the random oracle model). Below is the detailed construction of our IBSDVS scheme. $\mathsf{Setup}(1^k)$: The PKG chooses two cyclic groups of prime order p of k bits, $\mathbb{G}$ and $\mathbb{G}_T$, a random generator g of $\mathbb{G}$, and an admissible pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. It selects at random $\alpha \leftarrow_{\$} \mathbb{Z}_p$, sets $g_1 = g^{\alpha}$, and selects three collision-resistant hash functions, $\mathtt{H}_1 : \{0, 1\}^* \to \mathbb{G}$, $\mathtt{H}_2 : \{0, 1\}^* \to \mathbb{G} \setminus \{1\}$, $\mathtt{H}_3 : (\{0, 1\}^*)^3 \times \mathbb{G}^3 \times \mathbb{G}_T^2 \to \mathbb{Z}_p$, $\mathtt{H}_4 : \mathbb{G} \times \mathbb{G}_T \to \mathbb{G}$ and $\mathtt{H}_5 : \mathbb{G} \to \mathbb{Z}_p$, which will be modeled as random oracles in the security proofs. The master public key is set to be $\mathtt{m}pk = (g, g_1, \mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_3, \mathtt{H}_4, \mathtt{H}_5)$, and the master secret key is $\mathtt{m}sk = \alpha$. $\mathsf{Extract}(\mathtt{m}sk, \mathtt{i}d)$: The secret key of a user with identity $\mathtt{i}d$ is set to be $\mathtt{u}sk_{\mathtt{i}d} = \mathtt{H}_1(\mathtt{i}d)^{\alpha}$. $\mathsf{Sign}(\mathtt{u}sk_S, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{m}pk, M)$: To sign a message M with respect to the designated verifier (with identity $\mathtt{i}d_V$), the signer (with identity $\mathtt{i}d_S$) does as follows:

1. Choose at random $r \leftarrow_{\$} \mathbb{Z}_p$. Set $\bar{S}_1 = \mathtt{usk}_S \cdot \mathtt{H}_2(M)^r$, and $s = \mathtt{H}_5(\bar{S}_1)$.
2. Set $S_2 = g^s$, $T = \hat{\mathrm{e}}(\mathtt{H}_1(\mathtt{i}d_V), g_1)^s$, and $S_1 = \bar{S}_1 \cdot \mathtt{H}_4(S_2, T)$.
3. Using r and hash function $\mathtt{H}_3$, compute the following proof of knowledge:

$$S_3 = PK\left\{\beta : \hat{e}(\mathtt{H}_2(M), g)^{\beta} = \frac{\hat{e}(\bar{S}_1, g)}{\hat{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)} \vee \hat{e}(\mathtt{H}_2(M), g)^{\beta} = \frac{\hat{e}(\bar{S}_1, g)}{\hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)}\right\}(\bar{M}) \tag{2}$$

   where $\bar{M} = (\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1)$. Readers can refer to Section 5.2 for the details of the generation and verification of (2).
4. Set $\sigma = (S_1, S_2, S_3)$.

$\mathsf{Ver}(M, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{u}sk_V, \mathtt{m}pk, \sigma)$: After receiving a signature $\sigma = (S_1, S_2, S_3)$ and a message M from the signer (with identity $\mathtt{i}d_S$), the verifier (with identity $\mathtt{i}d_V$ and secret key $\mathtt{u}sk_V$) does as the following:

1. Compute $T = \hat{e}(\mathtt{usk}_V, S_2)$, $\bar{S}_1 = S_1 / \mathtt{H}_4(S_2, T)$ and $s = \mathtt{H}_5(\bar{S}_1)$.
2. Check if $S_2 \overset{?}{=} g^s$, and check the validity of the proof of knowledge S_3 with respect to $\bar{S}_1$ and the 'message' $\bar{M} = (\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1)$. It rejects if either fails, and accepts otherwise.

$\mathsf{Sim}(\mathtt{u}sk_V, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{m}pk, M)$: To simulate a signature on M, the verifier does as the signer, except that $\bar{S}_1$ is computed as $\bar{S}_1 = \mathtt{usk}_V \cdot \mathtt{H}_2(M)^r$.

Efficiency: It is readily seen that the proposed IBSDVS scheme above is practically efficient. The master public key consists of two group elements of $\mathbb{G}$ and the description of five hash functions. The secret key of each user contains merely one element of $\mathbb{G}$. A signature consists of two elements of $\mathbb{G}$, two elements of $\mathbb{G}_T$ and three elements of $\mathbb{Z}_p$. The signing algorithm requires two exponentiations in $\mathbb{G}$, four exponentiations in $\mathbb{G}_T$ and three pairing evaluations. The verification of a signature requires one exponentiation in $\mathbb{G}$, four exponentiations in $\mathbb{G}_T$ and five pairing evaluations.

5.2. Details of generation and verification of (2)

To generate the proof of knowledge (2), the signer does as the following:

1. Choose $r_0, e_1, z_1 \leftarrow_{\$} \mathbb{Z}_p$.
2. Compute $R_0 = \hat{\mathrm{e}}(\mathtt{H}_2(M), g)^{r_0}$ and $R_1 = \hat{e}(\mathtt{H}_2(M), g)^{z_1} \cdot (\hat{e}(\bar{S}_1, g) / \hat{e}(H_1(\mathtt{id}_V), g_1))^{-e_1}$.
3. Set $e = \mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1)$, $e_0 = e - e_1$ and $z_0 = r_0 + \beta e_0$.

The proof of knowledge S_3 is set to be $S_3 = (R_0, e_0, z_0, R_1, z_1)$. [3]

A designated verifier with identity $\mathtt{i}d_V$ can produce an indistinguishable proofs of knowledge similarly. The difference is to replace the subscripts of the variables above with their complements.

To verify a proof of knowledge $S_3 = (R_0, e_0, z_0, R_1, z_1)$, the verifier does as the following:

[3] Actually, one can set $S_3 = (e_0, z_0, e_1, z_1)$ which has smaller size. However, for the sake of simplicity in the security proofs, we choose to include the R values in S_3.

1. Compute $e_1 = \mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1) - e_0$.
2. Accept if the two equations below hold, and reject otherwise.

$$\hat{e}(\mathtt{H}_2(M), g)^{z_0} \overset{?}{=} R_0 \cdot \left(\frac{\hat{e}(\bar{S}_1, g)}{\hat{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)} \right)^{e_0}, \tag{3}$$

$$\hat{e}(\mathtt{H}_2(M), g)^{z_1} \overset{?}{=} R_1 \cdot \left(\frac{\hat{e}(\bar{S}_1, g)}{\hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)} \right)^{e_1}. \tag{4}$$

The proof of knowledge can be simulated without the knowledge of β efficiently in the random oracle model. Namely, the simulator randomly selects $e_0, z_0, e_1, z_1 \leftarrow_\$ \mathbb{Z}_p$, computes

$$R_0 = \frac{\hat{e}(\mathtt{H}_2(M), g)^{z_0}}{(\hat{e}(\bar{S}_1, g)/\hat{e}(H_1(\mathtt{id}_V), g_1))^{e_0}} \quad \text{and}$$

$$R_1 = \frac{\hat{e}(\mathtt{H}_2(M), g)^{z_1}}{(\hat{e}(\bar{S}_1, g)/\hat{e}(H_1(\mathtt{id}_V), g_1))^{e_1}},$$

and then patches the random oracle $\mathtt{H}_3$ with $((\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1), e)$, i.e. setting $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1) = e$. It is easy to see that the simulated proof also passes the verification above, and the simulated proof is perfectly indistinguishable from a real proof generated by the signer or the designated verifier.

Moreover, given two valid tuples (R_0, e_0, z_0, R_1, z_1) and $(R_0, e'_0, z'_0, R_1, z'_1)$ and two different answers to the query $(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1)$ returned by the random oracle $\mathtt{H}_3$, say e and $e' \neq e$, there is an efficient algorithm which extracts the secret β from the two tuples.

If $e_0 \neq e'_0$. Let $R_0 = g^{r_0}$ for some $r_0 \in \mathbb{Z}_p$. From the two instances of Eq. (3) we have that $z_0 = r_0 + e_0\beta_0$ and $z'_0 = r_0 + e'_0\beta_0$. Then β_0 can be obtained by computing $\beta_0 = (z_0 - z'_0) \cdot (e_0 - e'_0)^{-1}$. It can be verified that $\hat{e}(\bar{S}_1, g) \cdot \hat{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)^{-1} = \hat{e}(\mathtt{H}_2(M), g)^{\beta_0}$. On the other hand, if $e - e_0 \neq e' - e'_0$, the extractor can extract another $\beta_1 \in \mathbb{Z}_p$ from (e_1, z_1, e'_1, z'_1) as above, such that $\hat{e}(\bar{S}_1, g) \cdot \hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)^{-1} = \hat{e}(\mathtt{H}_2(M), g)^{\beta_1}$.

6. Security analysis and comparison

6.1. Proofs

Theorem 6.1. *If CDH assumption (t, ϵ)-holds in $\mathbb{G}$, the IBDVS scheme above is $(t', q_{\mathtt{H}_1}, q_{\mathtt{H}_2}, q_{\mathtt{H}_3}, q_{\mathtt{H}_4}, q_{\mathtt{H}_5}, q_{\mathsf{E}}, q_{\mathsf{Sign}}, q_{\mathsf{Sim}}, q_{\mathsf{Ver}}, \epsilon')$-unforgeable, where $t' = \Theta(t)$ and $\epsilon' < 16q_{\mathtt{H}_1}^2 \sqrt{q_{\mathtt{H}_3}\sqrt{\epsilon}} + (2/p)$.*

Proof. Given an adversary $\mathcal{A}$ against the unforgeability of the IBSDVS scheme, we use it to build another algorithm $\mathcal{B}$ for solving the CDH problem. Given a random instance of CDH problem, $(g, g_1 = g^a, g_2 = g^b)$, $\mathcal{B}$ aims to find g^{ab}. It works as follows:

Setup: $\mathcal{B}$ invokes $\mathcal{A}$ on input $\mathtt{m}pk = (g, g_1)$. Note that the master secret key is $\mathtt{m}sk = \log_g g_1 = a$ that is unknown to $\mathcal{B}$. It selects at random two distinct indices i_S, i_V from $\{1, 2, \cdots, q_{\mathtt{H}_1}\}$, and then simulates oracles for $\mathcal{A}$ as below, which includes random oracles $\mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_3, \mathtt{H}_4, \mathtt{H}_5$, extraction oracle, signing oracle and verification oracle.

Query: $\mathcal{B}$ simulates the following oracles for $\mathcal{A}$ by maintaining five hash tables, $HT_1, HT_2, HT_3, HT_4, HT_5$. For simplicity, we assume that $\mathcal{A}$ does not issue repeated queries to the oracles,[4] and that $\mathcal{A}$ would not use a hash value before asking the corresponding random oracle on the input.

- $\mathtt{H}_1$ *Query*: Given the i-th query $\mathtt{id}_i$, $\mathcal{B}$ selects a distinct $w_i \leftarrow_\$ \mathbb{Z}_p$ at random. If $i \notin \{i_S, i_V\}$, $\mathcal{B}$ sets $\mathtt{H}_1(\mathtt{id}_i) = g^{w_i}$; otherwise, it sets $\mathtt{H}_1(\mathtt{id}_i) = g_2^{w_i}$. In either case, $\mathcal{B}$ stores the tuple $(\mathtt{id}_i, \mathtt{H}_1(\mathtt{id}_i), w_i)$ into table HT_1 and returns $\mathtt{H}_1(\mathtt{id}_i)$ back to $\mathcal{A}$.
- $\mathtt{H}_2$ *Query*: Given a query M, $\mathcal{B}$ randomly selects a distinct $m \in \mathbb{G}$, stores (M, m) into table HT_2, and returns m.
- $\mathtt{H}_3$ *Query*: Given a query $(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1)$, $\mathcal{B}$ chooses at random a distinct $e \leftarrow_\$ \mathbb{Z}_p$, and sets $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1) = e$. It returns e to $\mathcal{A}$ and stores $((\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1), e)$ in table HT_3.
- $\mathtt{H}_4$ *Query*: Given a query (S_2, T), $\mathcal{B}$ selects at random a distinct $\bar{T} \leftarrow_\$ \mathbb{G}$. It returns $\bar{T}$ and stores $((S_2, T), \bar{T})$ into table HT_4.
- $\mathtt{H}_5$ *Query*: Given a query $\bar{S}_1$, $\mathcal{B}$ chooses at random a distinct $s \leftarrow_\$ \mathbb{Z}_p$. It returns s and stores $(\bar{S}_1, s)$ into table HT_5.
- $\mathsf{Extract}$ *Query*: Given an identity $\mathtt{id}$, $\mathcal{B}$ retrieves the tuple $(\mathtt{id}, \mathtt{H}_1(\mathtt{id}), c, w)$ from table HT_1. If $\mathtt{id} \notin \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$, $\mathcal{B}$ computes the user secret key as $\mathtt{usk_{id}} = g_1^w$ and returns it to $\mathcal{A}$; otherwise, it aborts.
- $\mathsf{Sign}/\mathsf{Sim}$ *Query*: Thanks to the perfect non-transferability, it suffices to consider how to answer Sign queries. Given a query $(\mathtt{id}_S, \mathtt{id}_V, M)$, $\mathcal{B}$ retrieves the tuple $(\mathtt{i}d_S, \mathtt{H}_1(\mathtt{i}d_S), c_S, t_S)$ from HT_1. We consider the following cases:
  - $\mathtt{id}_S \notin \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$: $\mathcal{B}$ generates the user secret key of $\mathtt{i}d_S$ as in the simulation of $\mathsf{Extract}$ oracle, and then computes the signature σ by running the Sign algorithm on input $(\mathtt{u}sk_S, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{m}pk, M)$.
  - $\mathtt{id}_V \notin \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$: $\mathcal{B}$ generates the user secret key of $\mathtt{i}d_V$, and prepares the answer as above using $\mathtt{u}sk_V$.
  - $\{\mathtt{id}_S, \mathtt{id}_V\} = \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$: $\mathcal{B}$ selects at random $\bar{S}_1 \leftarrow_\$ \mathbb{G}$. Note that there exists some $r \in \mathbb{Z}_p$ (unknown to $\mathcal{B}$) such that $\bar{S}_1 = \mathtt{H}(\mathtt{id}_S)^a \cdot \mathtt{H}_2(M)^r$. $\mathcal{B}$ then calls the random oracle $\mathtt{H}_5$ on input $\bar{S}_1$, and gets s. It computes $T = \hat{\mathrm{e}}(\mathtt{H}_1(\mathtt{i}d_V), g_1)^s$ and $S_2 = g^s$, asks the random oracle $\mathtt{H}_4$ on input (S_2, T) and obtains the answer $\bar{T} = \mathtt{H}_4(S_2, T)$. It then sets $S_1 = \bar{S}_1 \cdot \bar{T}$, and calls the simulator of the proof of knowledge to produce S_3 in the way specified in Section 5.2. In case there is any collision when patching the oracles $\mathtt{H}_3, \mathtt{H}_4, \mathtt{H}_5$, $\mathcal{B}$ aborts. This event occurs only with probability at most $(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})(3(q_{\mathsf{Sign}} + q_{\mathsf{Sim}}) + q_{\mathtt{H}_3} + q_{\mathtt{H}_4} + q_{\mathtt{H}_5})/p$ in the whole simulation. Denote this upper bound by γ_1.
- Ver *Query*: Given a query $(M, \mathtt{i}d_S, \mathtt{i}d_V, \sigma)$ where $\sigma = (S_1, S_2, S_3)$, we consider the following two cases:
  - $\mathtt{id}_V \notin \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$: $\mathcal{B}$ generates the user secret key of $\mathtt{i}d_V$, runs the verification algorithm on input $(M, \mathtt{i}d_S, \mathtt{i}d_V, \mathtt{u}sk_V, \mathtt{m}pk, \sigma)$ and returns the decision bit.
  - $\mathtt{id}_V \in \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$: $\mathcal{B}$ looks up the table HT_5: for each tuple $(\bar{S}_1, s)$ in HT_5, $\mathcal{B}$ checks if $S_2 = g^s$. If such a tuple is found, $\mathcal{B}$ stops the look-up. Note that according to the simulation of $\mathtt{H}_5$, with probability at most $(q_{\mathtt{H}_5} + q_{\mathsf{Sign}} + q_{\mathsf{Sim}})/p$, there are more than one satisfactory tuples. Denote the bound by γ_2. $\mathcal{B}$ computes $T = \hat{\mathrm{e}}(\mathtt{H}_1(\mathtt{i}d_V), g_1)^s$, asks the random oracle $\mathtt{H}_4$ on input (S_2, T) and obtains the answer $\bar{T}$. It then checks if $\bar{S}_1 \overset{?}{=} S_1/\bar{T}$. If not, $\mathcal{B}$ returns 0 indicating that the signature is invalid with respect to $\mathtt{i}d_S, \mathtt{i}d_V$; otherwise, $\mathcal{B}$ checks the validity of the proof of knowledge S_3 with respect to the 'message' $(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1)$. If invalid, $\mathcal{B}$ returns 0; otherwise it returns 1.

  If there is no such a tuple in HT_5, $\mathcal{B}$ simply returns 0. In this case, instead of asking $\mathtt{H}_5$ for s, $\mathcal{A}$ chose s by itself. Let $\bar{S}_1$ be $S_1/\mathtt{H}_4(S_2, \hat{\mathrm{e}}(\mathtt{H}_1(\mathtt{i}d_V), g_1)^s)$. Since the output of oracle $\mathtt{H}_5$ is random, with probability at most $1/p$, the s and $\bar{S}_1$ chosen by $\mathcal{A}$ satisfies that $\mathtt{H}_5(\bar{S}_1) = s$. Hence, with probability at least $(1 - 1/p)$, the signature submitted by $\mathcal{A}$ is invalid with respect to $\mathtt{i}d_S, \mathtt{i}d_V$.

Forge: Finally, $\mathcal{A}$ outputs its forgery, $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, \sigma^*)$ where $\sigma^* = (S_1^*, S_2^*, S_3^*)$, and $S_3^* = (R_0^*, e_0^*, z_0^*, R_1^*, e_1^*)$. Assume that the forgery is valid. If $\{\mathtt{id}_S^*, \mathtt{id}_V^*\} \neq \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$, $\mathcal{B}$ aborts. The valid-

[4] If a query was asked before, $\mathcal{B}$ retrieves the answer from its memory and returns it.

ity of σ^* guarantees that with probability at least $(1-1/p)$ there exist a tuple $((\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, S_2^*, \bar{S}_1^*, R_0^*, R_1^*), e^*)$ in HT_3, a tuple $((S_2^*, T^*), \bar{T}^*)$ in HT_4 and a tuple $(\bar{S}_1^*, s^*)$ in HT_5 such that S_2^* in the first tuple is equal to that in the second tuple, $\bar{S}_1^*$ in the first tuple is equal to that in the third tuple, $g^{s^*} = S_2^*$ and $S_1^*/\bar{T}^* = \bar{S}_1^*$. $\mathcal{B}$ rewinds $\mathcal{A}$ to the status of querying oracle $\mathtt{H}_3$ on input $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, S_2^*, \bar{S}_1^*, R_0^*, R_1^*)$. It chooses at random $e'^* \neq e^* \in \mathbb{Z}_p$, and answers $\mathcal{A}$ with e'^*. It then continues to simulate oracles for $\mathcal{A}$ as in **Query** phase. Suppose that, $\mathcal{A}$ outputs a successful forgery once again, say $(\mathtt{id}'^*_S, \mathtt{id}'^*_V, M'^*, \sigma'^*)$ where $\sigma'^* = (S'^*_1, S'^*_2, S'^*_3)$ and $S'^*_3 = (R'^*_0, e'^*_0, z'^*_0, R'^*_1, z'^*_1)$. If $(\mathtt{id}'^*_S, \mathtt{id}'^*_V, M'^*, S'^*_1, S'^*_2, \bar{S}_1^{*'}, R'^*_0, R'^*_1) \neq (\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, S_2^*, \bar{S}_1^*, R_0^*, R_1^*)$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ calls the extractor (see Section 5.2) to extract the secret randomness r^* from $(S_1^*, S_2^*, \bar{S}_1^*, (S_3^*, S'^*_3))$. Retrieve the tuples $(\mathtt{id}_S^*, \mathtt{H}_1(\mathtt{id}_S^*), w_S)$ and $(\mathtt{id}_V^*, \mathtt{H}_1(\mathtt{id}_V^*), w_V)$ from HT_1.

- If $\hat{e}(\mathtt{H}_2(M^*), g)^{r^*} = \hat{e}(\bar{S}_1^*, g)/\hat{e}(\mathtt{H}_1(\mathtt{id}_S^*), g_1)$, we have that $\bar{S}_1^* = \mathtt{H}_1(\mathtt{id}_S^*)^a \cdot \mathtt{H}_2(M^*)^{r^*}$. Recall that $\mathtt{H}_1(\mathtt{id}_S^*) = g_2^{w_S} = g^{bw_S}$. Hence, $\mathcal{B}$ recovers g^{ab} from $\bar{S}_1^*$ by computing $g^{ab} = \left(\bar{S}_1^*/\mathtt{H}_2(M^*)^{r^*}\right)^{1/w_S}$.
- If $\hat{e}(\mathtt{H}_2(M^*), g)^{r^*} = \hat{e}(\bar{S}_1^*, g)/\hat{e}(\mathtt{H}_1(\mathtt{id}_V^*), g_1)$, we have that $\bar{S}_1^* = \mathtt{H}_1(\mathtt{id}_V^*)^a \cdot \mathtt{H}_2(M^*)^{r^*}$. Recall that $\mathtt{H}_1(\mathtt{id}_V^*) = g_2^{w_V} = g^{bw_V}$. Hence, $\mathcal{B}$ recovers g^{ab} from $\bar{S}_1^*$ by computing $g^{ab} = \left(\bar{S}_1^*/\mathtt{H}_2(M^*)^{r^*}\right)^{1/w_V}$.

In either case $\mathcal{B}$ solves the given CDH problem.

Probability analysis: There are some cases in the first run of $\mathcal{A}$ in which $\mathcal{B}$ aborts.

1. A collision occurs when patching the oracles $\mathtt{H}_3$, $\mathtt{H}_4$, $\mathtt{H}_5$ in simulating oracles Sign and Sim. As analyzed above, the event does not occur with probability at least $1-\gamma_1$.
2. $\mathcal{A}$ issues a query to Extract oracle on input $\mathtt{id}_{i_S}$ or $\mathtt{id}_{i_V}$, or $\{\mathtt{id}_S^*, \mathtt{id}_V^*\} \neq \{\mathtt{id}_{i_S}, \mathtt{id}_{i_V}\}$ in the forgery phase. According to the game specification, $\mathcal{A}$ could not issue queries to Extract oracle on input $\mathtt{id}_S^*$, $\mathtt{id}_V^*$ in order to win the game. Hence, this event does not occur with probability at least $1/\binom{q_{\mathtt{H}_1}}{2} = 2/q_{\mathtt{H}_1}(q_{\mathtt{H}_1}-1) > 2/q_{\mathtt{H}_1}^2$, due to the random choices of i_S, i_V.
3. $\mathcal{B}$ believes that $\mathcal{A}$'s forgery is invalid. This does not occur with probability at least $(1-1/p)\epsilon'$, where the factor $1-1/p$ is due to the same reason in the second case in answering a Ver query.

Conditioned on that $\mathcal{B}$ does not abort, the simulation of the oracles by $\mathcal{B}$ is perfect, except that the simulation of Ver oracle is at most $(\gamma_2 + q_{\mathsf{Ver}}/p)$ different from a real oracle, where q_{Ver}/p comes from that $\mathcal{A}$ guesses the output of $\mathtt{H}_5$ correctly in the q_{Ver} queries to oracle Ver (the second case in the simulation of oracle Ver). Therefore, in the first run of $\mathcal{A}$, algorithm $\mathcal{B}$ does not abort with probability at least

$$\varepsilon \geq \frac{2}{q_{\mathtt{H}_1}^2}(1-\gamma_1)\left(1-\gamma_2-\frac{q_{\mathsf{Ver}}}{p}\right)\left(\left(1-\frac{1}{p}\right)\epsilon' - \frac{1}{p}\right)$$

where the last $1/p$ comes from that $\mathcal{A}$ guesses the value $\mathtt{H}_3(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, S_2^*, \bar{S}_1^*, R_0^*, R_1^*)$ correctly. A similar analysis with that in Pointcheval and Stern (2000); Boneh et al. (2004) shows that with probability at least $\epsilon \geq \varepsilon^2/16q_{\mathtt{H}_3}$, $\mathcal{A}$ outputs a successful forgery in the second run that satisfies the aforementioned conditions (given in the **Forge** phase), which, together with the valid forgery in the first run, enables $\mathcal{B}$ to solve the given CDH problem. Thus we have that,

$$\begin{aligned}
\epsilon \geq \frac{\varepsilon^2}{16q_{\mathtt{H}_3}} &\geq \frac{1}{4q_{\mathtt{H}_3}q_{\mathtt{H}_1}^4}(1-\gamma_1)^2\left(1-\gamma_2-\frac{q_{\mathsf{Ver}}}{p}\right)^2\left(\left(1-\frac{1}{p}\right)\epsilon' - \frac{1}{p}\right)^2 \\
&\overset{(*)}{>} \frac{1}{64q_{\mathtt{H}_3}q_{\mathtt{H}_1}^4}\left(\left(1-\frac{1}{p}\right)\epsilon' - \frac{1}{p}\right)^2 \\
\Rightarrow \epsilon' &< \left(8q_{\mathtt{H}_1}^2\sqrt{q_{\mathtt{H}_3}}\sqrt{\epsilon} + \frac{1}{p}\right)\left(1-\frac{1}{p}\right)^{-1} < 16q_{\mathtt{H}_1}^2\sqrt{q_{\mathtt{H}_3}}\sqrt{\epsilon} + \frac{2}{p}
\end{aligned}$$

where the inequality (*) follows from the fact that $\gamma_1 < 1/2$ and $\gamma_2 + q_{\mathsf{Ver}}/p < 1/2$. This completes the proof.

Theorem 6.2. *The proposed IBSDVS scheme is perfectly non-transferable* (see Def. 2.3).

Proof. Recall that a signature (S_1, S_2, S_3) from the signer satisfies that

$$S_1 = \mathtt{H}_1(\mathtt{id}_S)^\alpha \cdot \mathtt{H}_2(M)^r \cdot \mathtt{H}_4(g^s, \hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)^s), S_2 = g^s,$$

and a signature (S'_1, S'_2, S'_3) from the designated verifier satisfies that

$$S'_1 = \mathtt{H}_1(\mathtt{id}_V)^\alpha \cdot \mathtt{H}_2(M)^{r'} \cdot \mathtt{H}_4(g^{s'}, \hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)^{s'}), S'_2 = g^{s'}.$$

where $s = \mathtt{H}_5(\mathtt{H}_1(\mathtt{id}_S)^\alpha \cdot \mathtt{H}_2(M)^r)$ and $s' = \mathtt{H}_5(\mathtt{H}_1(\mathtt{id}_V)^\alpha \mathtt{H}_2(M)^{r'})$. Thus, (S_1, S_2) (resp. (S'_1, S'_2)) is fully determined by $\bar{S}_1 = \mathtt{H}_1(\mathtt{id}_S)^\alpha \cdot \mathtt{H}_2(M)^r$ (resp. $\bar{S}'_1 = \mathtt{H}_1(\mathtt{id}_V)^\alpha \cdot \mathtt{H}_2(M)^{r'}$). Therefore, it is equivalent to consider the indistinguishability between $(\bar{S}_1, S_3)$ and $(\bar{S}'_1, S'_3)$.

Since the group $\mathbb{G}$ is of prime order, $\mathtt{H}_2(M)$ is a generator of $\mathbb{G}$, and thus $\mathtt{H}_2(M)^r$ perfectly hides the user secret key. Therefore, $\mathtt{usk}_S \cdot \mathtt{H}_2(M)^r$ and $\mathtt{usk}_V \cdot \mathtt{H}_2(M)^{r'}$ are identically distributed. Given an $\bar{S}_1$, there exist $r, r' \in \mathbb{Z}_p$ such that $\bar{S}_1 = \mathtt{usk}_S \cdot \mathtt{H}_2(M)^r = \mathtt{usk}_V \cdot \mathtt{H}_2(M)^{r'}$. On the other hand, the proof of knowledge S_3 is perfectly witness indistinguishable, thus revealing no information about the randomness r. Hence, $(\bar{S}_1, S_3)$ and $(\bar{S}'_1, S'_3)$ are identically distributed. In a consequence, the signature $\sigma = (S_1, S_2, S_3)$ is information-theoretically hiding. □

Theorem 6.3. *Assume that for some identities* $\mathtt{id}_S, \mathtt{id}_V \in \{0, 1\}^*$ *and some message* $M \in \{0, 1\}^*$, *the algorithm* $\mathcal{F}$ *can produce valid signatures in time t and with probability* ϵ. *Then the IBSDVS scheme is* $(64t/\epsilon, 1/p)$*-non-delegatable* (see Def. 2.4) *in the random oracle model.*

Proof. Assume that $\epsilon > \kappa = 1/p$, where $1/p$ is the probability that $\mathcal{F}$ guesses correctly the hash value without asking the random oracle $\mathtt{H}_3$. There is an extractor $\mathcal{K}$ that, on input σ and black-box oracle access to algorithm $\mathcal{F}$, extracts the secret key of either the signer or the designated verifier.

Let $\mathcal{F}_{S,V,M}$ be a forger with input $(\mathtt{id}_S, \mathtt{id}_V, M)$. Consider two runs of $\mathcal{F}_{S,V,M}$ on the same random input to $\mathcal{F}_{S,V,M}$. In both runs, $\mathcal{K}$ executes $\mathcal{F}_{S,V,M}$ step-by-step and simulates the random oracles for $\mathcal{A}$, except that $\mathcal{K}$ returns different random values (e versus e') as the answer to the hash query $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, S_2, \bar{S}_1, R_0, R_1)$. Since $S_1, S_2, \bar{S}_1, R_0, R_1$ are in the input to the hash function, their values must be unchanged in both of the two runs. If both signatures, i.e. $(S_1, S_2, S_3 = (R_0, e_0, z_0, R_1, z_1))$ and $(S_1, S_2, S'_3 = (R_0, e'_0, z'_0, R_1, z'_1))$, are valid, one can extract the user secret key as follows:

1. Find in table HT_5 (for the simulation of oracle $\mathtt{H}_5$) the tuple $(\bar{S}_1, s)$ such that $g^s = S_2$ and $\bar{S}_1$ is equal to that in the input to oracle $\mathtt{H}_3$, and thus equal to $S_1/\mathtt{H}_4(S_2, \hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)^s)$. As discussed in the proof of Theorem 6.1, the validity of the signature guarantees that with probability at least $(1-1/p)(1-(q_{\mathtt{H}_5} + q_{\mathsf{Sign}} + q_{\mathsf{Sim}})/p)$ there is exactly one such tuple in the table.
2. Call the extractor of the proof of knowledge (described in Section 5.2) to extract the randomness r from (S_3, S'_3).

Table 1
Comparison between our scheme and other existing IBSDVS schemes.

Scheme	Signature-size	Non-Trans	Non-Dele	**UF**	**PSI**	**RO**	**Assump**
Huang et al. (2008)	$\mathbb{Z}_p^*$	Perfect	×	Full	√	√	GBDH
Kang et al. (2009)	$2\mathbb{G}_T$	Perfect	×	Weak	?	√	BDH
Susilo et al. (2004)	$1\mathbb{G} + 1\mathbb{Z}_p + 1\mathbb{Z}_p^*$	Perfect	?	Weak	?	√	BDH
Zhang and Mao (2008)	$3\mathbb{G}$	Perfect	?	Weak	?	√	BDH
Ours	$2\mathbb{G} + 2\mathbb{G}_T + 3\mathbb{Z}_p$	Perfect	√	Full	√	√	CDH

If $\hat{e}(\mathtt{H}_2(M), g)^r = \hat{e}(\bar{S}_1, g)/\hat{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)$, one can find $\mathtt{usk}_S = \bar{S}_1/\mathtt{H}_2(M)^r$. If $\hat{e}(\mathtt{H}_2(M), g)^r = \hat{e}(\bar{S}_1, g)/\hat{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)$, one can find $\mathtt{usk}_V = \bar{S}_1/\mathtt{H}_2(M)^r$.

Now assume that Rewind is an algorithm that given oracle access to $\mathcal{F}_{S,V,M}$, in time T_R produces two different valid signatures $(S_1, S_2, S_3 = (R_0, e_0, z_0, R_1, z_1))$ and $(S'_1, S'_2, S'_3 = (R'_0, e'_0, z'_0, R'_1, z'_1))$ on M with respect to $\mathtt{i}d_S$, $\mathtt{i}d_V$, such that $(S_1, S_2, \bar{S}_1, R_0, R_1) = (S'_1, S'_2, \bar{S}'_1, R'_0, R'_1)$. Then one can compute $\mathtt{u}sk_S$ or $\mathtt{u}sk_V$ with probability $(1 - 1/p)(1 - (q_{\mathtt{H}_5} + q_{\mathsf{Sign}} + q_{\mathsf{Sim}})/p)$. Thus, given that algorithm Rewind runs in expected time $56/((1 - 1/p)(1 - (q_{\mathtt{H}_5} + q_{\mathsf{Sign}} + q_{\mathsf{Sim}})/p)\epsilon) < 56/(7\epsilon/8) = 64/\epsilon$, we have proven the theorem.

The algorithm Rewind works as the following. We are given an algorithm $\mathcal{F}_{S,V,M}$ which returns a valid signature with probability at least ϵ, where the probability is taken over the random coins used by $\mathcal{F}_{S,V,M}$ and the random outputs of $\mathtt{H}_3$ (and $\mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_4, \mathtt{H}_5$). Let **H** be a matrix with a row for each possible set of random coins for $\mathcal{F}_{S,V,M}$ and the random outputs of the oracles $\mathtt{H}_1$, $\mathtt{H}_2$, $\mathtt{H}_4$ and $\mathtt{H}_5$, and one column for each possible $\mathtt{H}_3$ value e. Write 1 in an entry if $\mathcal{F}_{S,V,M}$ outputs a valid signature with corresponding random choices and the $\mathtt{H}_3$ value, and 0 otherwise. Using $\mathcal{F}_{S,V,M}$ as a black box, we can probe any entry in **H**, and the goal is to find two 1's in the same row. Note that ϵ equals the fraction of 1-entries in the matrix **H**. Using an algorithm from Damgård and Fujisaki (2002), Rewind can find such 1-entries in time $56/((1 - 1/p)(1 - (q_{\mathtt{H}_5} + q_{\mathsf{Sign}} + q_{\mathsf{Sim}})/p)\epsilon) < 64/\epsilon$. □

Theorem 6.4. *If DBDH assumption (t, ϵ)-holds, our IBSDVS scheme is $(t', q_{\mathtt{H}_1}, q_{\mathtt{H}_2}, q_{\mathtt{H}_3}, q_{\mathtt{H}_4}, q_{\mathtt{H}_5}, q_{\mathsf{E}}, q_{\mathsf{Sign}}, q_{\mathsf{Sim}}, q_{\mathsf{Ver}}, \epsilon')$-PSI-secure, where $t' \approx t$ and $\epsilon' \leq 2q_{\mathtt{H}_1}\epsilon + (1/p)$.*

Proof. Let $\mathcal{D}$ be a distinguisher for the PSI-security of our IBSDVS scheme. We use it to construct another algorithm D for breaking DBDH assumption. Given a random instance of DBDH problem, i.e. (g, g^a, g^b, g^c, Z), D invokes $\mathcal{D}$ on input $(g, g_1 = g^a)$, and then simulates oracles for it.

In the simulation of $\mathtt{H}_1$, D chooses at random $i \in \{1, \cdots, q_{\mathtt{H}_1}\}$. For the j-th $(j \neq i)$ query $\mathtt{id}_j$, D randomly selects $w_j \in \mathbb{Z}_p$, stores $(\mathtt{id}_j, g^{w_j}, w_j)$ in a hash table HT_1, and returns g^{w_j} to $\mathcal{D}$. For the i-th query $\mathtt{i}d_i$, D stores $(\mathtt{i}d_i, g^b, \perp)$ in HT_1, and returns g^b. Here $\mathtt{i}d_i$ is the guess of $\mathtt{id}_V^*$ that will be chosen by $\mathcal{D}$ as the target verifier.

Oracles $\mathtt{H}_2$, $\mathtt{H}_3$, $\mathtt{H}_4$, $\mathtt{H}_5$ can be simulated naturally. Namely, for a query, D randomly chooses its answer from the corresponding range of the function, and returns it. The query-answer pairs are stored in separate tables, i.e. HT_2, HT_3, HT_4, HT_5.

For the extraction oracle $\mathcal{O}_{\mathsf{E}}$, given an identity $\mathtt{id}$, if it is the i-th query to $\mathtt{H}_1$, D aborts and outputs a random bit; otherwise, let $(\mathtt{id}, g^w, w)$ be the entry in HT_1, D returns $\mathtt{usk}_{\mathtt{id}} = g_1^w$ to $\mathcal{D}$.

For the signing oracle $\mathcal{O}_{\mathsf{Sign}}$ and the simulation oracle $\mathcal{O}_{\mathsf{Sim}}$, given a query $(\mathtt{id}_S, \mathtt{id}_V, M)$, if $\mathtt{id}_S$ (resp. $\mathtt{id}_V$) is not the i-th query to $\mathtt{H}_1$, D computes $\mathtt{u}sk_S$ (resp. $\mathtt{u}sk_V$) as in the simulation of $\mathcal{O}_{\mathsf{E}}$, and then generates the signature σ on M by following the Sign (resp. Sim) algorithm. Since our scheme is perfectly non-transferable, the two oracles are perfectly simulated. Since $\mathtt{id}_S \neq \mathtt{id}_V$, it must hold that either $\mathtt{id}_S \neq \mathtt{id}_i$ or $\mathtt{id}_V \neq \mathtt{id}_i$.

For the verification oracle $\mathcal{O}_{\mathsf{Ver}}$, given a query $(M, \mathtt{id}_S, \mathtt{id}_V, \sigma)$ where $\sigma = (S_1, S_2, S_3)$, if $\mathtt{id}_V$ is not the i-th query to $\mathtt{H}_1$, D simply calls the Ver algorithm to give the answer as it knows $\mathtt{u}sk_V$; otherwise, it returns the answer in the same way as in the proof of unforgeability, which causes at most q_{Ver}/p difference from a real oracle.

At some time $\mathcal{D}$ outputs two signer identities $\mathtt{id}_{S_0}^*$, $\mathtt{id}_{S_1}^*$, a verifier identity $\mathtt{id}_V^*$ and a message M^*. If $\mathtt{id}_V^*$ is not the i-th query to $\mathtt{H}_1$, D aborts and returns a random bit. Otherwise, it tosses a random coin b, computes $\mathtt{usk}_{S_b}^*$, and produces σ^* on M^* for $\mathtt{id}_V^*$ using $\mathtt{usk}_{S_b}^*$ as below:

- Choose at random $r \in \mathbb{Z}_p$, and compute $\bar{S}_1^* = \mathtt{usk}_{S_b}^* \cdot \mathtt{H}_2(M^*)^r$.
- Set $S_2^* = g^c$, and $T^* = Z$, and $S_1^* = \bar{S}_1^* \cdot \mathtt{H}_4(S_2^*, T^*)$.
- Use r to generate S_3^* by following the generation of the proof of knowledge described in Section 5.2.

It returns $\sigma^* = (S_1^*, S_2^*, S_3^*)$ back to $\mathcal{D}$, which then continues to issue queries. D responses to the queries as before, except that if $\mathcal{D}$ queries $\mathtt{H}_5$ on input $\bar{S}_1^*$ or queries $\mathtt{H}_4$ on input (S_2^*, T^*), D aborts and outputs 1. Finally $\mathcal{D}$ outputs a bit b'. If $\mathcal{D}$ wins the game, D outputs 1; otherwise, it outputs 0.

Probability analysis: First of all, since i is randomly chosen from $\{1, \cdots, q_{\mathtt{H}_1}\}$, the probability that $\mathcal{D}$ did not query $\mathcal{O}_{\mathsf{E}}$ on input the i-th identity and $\mathtt{id}_V^*$ is exactly the i-th identity queried to $\mathtt{H}_1$, is $1/q_{\mathtt{H}_1}$. If D's guess of i is wrong, it outputs 1 with probability 1/2 no matter $Z = \hat{e}(g, g)^{abc}$ or not. So we do not need to consider it in the rest of the analysis. Conditioned on that D's guess of i is correct, we the analyze the probability that D outputs 1.

- $Z = \hat{e}(g, g)^{abc}$: In this case, the challenge signature is identically distributed as a real one. Let ε be the probability that $\mathcal{D}$ asks $\mathtt{H}_5$ on input $\bar{S}_1^*$ or asks $\mathtt{H}_4$ on input (S_2^*, T^*). According to the simulation above, D outputs 1 in this case with probability at least $\varepsilon + (1 - \varepsilon)(\epsilon' + 1/2)$.
- $Z \leftarrow_{\$} \mathbb{G}_T$: In this case, since Z is a random element of $\mathbb{G}_T$, $\mathcal{D}$ has no information about the value of $\mathtt{H}_4(S_2^*, Z)$ and S_1^* perfectly hides the bit b. Moreover, since r is randomly chosen from $\mathbb{Z}_p$, it becomes that $\bar{S}_1^*$ is also a random element of $\mathbb{G}$. Thus, the probability that it asks $\mathtt{H}_5$ on input $\bar{S}_1^*$ or asks $\mathtt{H}_4$ on input (S_2^*, Z) is at most $2/p$. Hence, the probability that D outputs 1 in this case is at most $2/p + (1 - 2/p)/2$.

Let d be the bit D outputs. We then have that

$$\begin{aligned}
\epsilon &= \left|\Pr[d = 1 \wedge Z = \hat{e}(g,g)^{abc}] - \Pr[d = 1 \wedge Z \leftarrow_{\$} \mathbb{G}_T]\right| \\
&= \frac{1}{2}\left|\Pr[d = 1 | Z = \hat{e}(g,g)^{abc}] - \Pr[d = 1 | Z \leftarrow_{\$} \mathbb{G}_T]\right| \\
&\geq \frac{1}{2}\frac{1}{q_{\mathtt{H}_1}}\left|\left(\varepsilon + (1-\varepsilon)\left(\epsilon' + \frac{1}{2}\right)\right) - \left(\frac{2}{p} + \left(1 - \frac{2}{p}\right)\frac{1}{2}\right)\right| \\
&= \frac{1}{2q_{\mathtt{H}_1}}\left|\frac{1}{2}\varepsilon + (1-\varepsilon)\epsilon' - \frac{1}{p}\right| \\
&= \frac{1}{2q_{\mathtt{H}_1}}\left|\epsilon' + \left(\frac{1}{2} - \epsilon'\right)\varepsilon - \frac{1}{p}\right| \\
&\geq \frac{1}{2q_{\mathtt{H}_1}}\left(\epsilon' - \frac{1}{p}\right) \quad \text{since}\, \epsilon' \leq \frac{1}{2} \\
\Rightarrow \epsilon' &\leq 2q_{\mathtt{H}_1}\epsilon + \frac{1}{p}
\end{aligned}$$

This completes the proof.

6.2. Comparison

In Table 1 we give a comparison of our scheme with those existing IBSDVS schemes, where **Non-Trans** indicates the level of non-transferability; **Non-Dele** indicates if the scheme is non-delegatable under the definition of Lipmaa et al. (2005); **UF** indicates if the unforgeability of the scheme is 'full' (in the sense of Def. 2.2), or 'weak' (in the sense of the definition discussed in Section 4); **PSI** indicates if the scheme supports privacy of signer's identity; **RO** indicates if the security of the scheme is in the random oracle model; and **Assump** indicates the assumption used in the proof of unforgeability of the scheme. Note that the question mark '?' in column **Non-Dele** means that it is unknown whether the scheme can be proved to be (non-)delegatable strictly under the definition in Lipmaa et al. (2005), and '?' in column **PSI** means that it is unknown if the scheme supports privacy of signer's identity.

To the best of our knowledge, before our work, the scheme proposed by Huang et al. (2008) is the only IBSDVS scheme in the literature that is unforgeable under Def. 2.2 (without the restriction). However, the unforgeability of their scheme relies on a very strong assumption, named GBDH assumption,[5] which has an interactive oracle for answering the problem-solver's queries, while that of our scheme is a very weak and widely studied assumption, CDH. In addition, Huang et al.'s scheme is delegatable, while ours is non-delegatable. Besides, as discussed in Remark 3, the privacy of signer's identity of our scheme is also stronger than that of Huang et al. (2008). In terms of signature size, our scheme has longer signature than that of Huang et al. (2008). However, since our scheme relies on weaker assumption, it seems reasonable and unavoidable to have larger signature size.

7. Conclusion

In this paper we proposed a new model of unforgeability of IBSDVS schemes, and showed that it is strictly stronger than the old one considered in previous work. We then proposed a new efficient construction of IBSDVS scheme, which is provably unforgeable under the newly proposed model, based on CDH assumption in the random oracle model. Our scheme enjoys perfectly non-transferability, and is the first IBSDVS scheme that is provably non-delegatable. We also showed that the scheme supports privacy of signer's identity.

[5] Gap Bilinear Diffie–Hellman (GBDH) assumption says that given g, g^a, g^b, g^c for random $a, b, c \in \mathbb{Z}_p$, and an oracle that takes as input g^d, g^e, g^f, Z where $d, e, f \in \mathbb{Z}_p$ and returns 1 if $Z = \hat{e}(g,g)^{def}$ and 0 otherwise, it is infeasible to find $\hat{e}(g,g)^{abc}$.

References

Baek, J., Safavi-Naini, R., Susilo, W., 2005. Universal designated verifier signature proof (or how to efficiently prove knowledge of a signature). In: Advances in Cryptology—ASIACRYPT 2005, vol. 3788. Lecture Notes in Computer Science, Springer, pp. 644–661.

Bellare, M., Rogaway, P., 1993. Random oracles are practical: a paradigm for designing efficient protocols. In: ACM Conference on Computer and Communications Security, ACM, pp. 62–73.

Boneh, D., Boyen, X., Shacham, H., 2004. Short group signatures. In: Advances in Cryptology—CRYPTO 2004, vol. 3152. Lecture Notes in Computer Science, Springer, pp. 41–55.

Cao, F., Cao, Z., 2009. An identity based universal designated verifier signature scheme secure in the standard model. The Journal of Systems and Software 82 (4), 643–649.

Damgård, I., Fujisaki, I., 2002. An integer commitment scheme based on groups with hidden order. In: Advances in Cryptology—ASIACRYPT 2002, vol. 2501. Lecture Notes in Computer Science, Springer, pp. 125–142.

Gentry, C., Silverberg, A., 2002. Hierarchical id-based cryptography. In: Advances in Cryptology—ASIACRYPT 2002, vol. 2501. Lecture Notes in Computer Science, Springer, pp. 548–566.

Huang, X., Susilo, W., Mu, Y., Wu, W., 2006. Universal designated verifier signature without delegatability. In: Proceedings of 8 th International Conference on Information and Communications Security, ICICS 2006, vol. 4307. Lecture Notes in Computer Science, Springer, pp. 479–498.

Huang, X., Susilo, W., Mu, Y., Wu, W., 2007. Secure universal designated verifier signature without random oracles. International Journal of Information Security 7 (3), 171–183.

Huang, X., Susilo, W., Mu, Y., Zhang, F., 2008. Short designated verifier signature scheme and its identity-based variant. International Journal of Network Security 6 (1), 82–93.

Jakobsson, M., Sako, K., Impagliazzo, R. 1996. Designated verifier proofs and their applications. In: Advances in Cryptology—EUROCRYPT 96, vol. 1070. Lecture Notes in Computer Science, Springer, pp. 143–154.

Kang, B., Boyd, C., Dawson, E., 2009. A novel identity-based strong designated verifier signature scheme. The Journal of Systems and Software 82 (2), 270–273.

Laguillaumie, F., Vergnaud, D., 2004. Designated verifier signatures: anonymity and efficient construction from any bilinear map. In: Proceedings of 4 th International Conference on Security in Communication Networks, SCN 2004, vol. 3352. Lecture Notes in Computer Science, Springer, pp. 105–119.

Laguillaumie, F., Libert, B., Quisquater, J.-J., 2006. Universal designated verifier signatures without random oracles or non-black box assumptions. In: Proceedings of 5 th International Conference on Security and Cryptography for Networks, SCN 2006, vol. 4116. Lecture Notes in Computer Science, Springer, pp. 63–77.

Li, Y., Lipmaa, H., Pei, D., 2005. On delegatability of four designated verifier signatures. In: Proceedings of 7 th International Conference on Information and Communications Security, ICICS 2005, vol. 3783. Lecture Notes in Computer Science, Springer, pp. 61–71.

Lipmaa, H., Wang, G., Bao, F., 2005. Designated verifier signature schemes: Attacks new security notions and a new construction. In: Proceedings of 32 th International Colloquium on Automata, Languages and Programming, ICALP 2005, vol. 3580. Lecture Notes in Computer Science, Springer, pp. 459–471.

Pointcheval, D., Stern, J., 2000. Security arguments for digital signatures and blind signatures. Journal of Cryptology 13 (3), 361–396.

Saeednia, S., Kremer, S., Markowitch, O., 2003. An efficient strong designated verifier signature scheme. In: Proceedings of International Conference on Information Security and Cryptology, ICISC 2003, vol. 2971. Lecture Notes in Computer Science, Springer, pp. 40–54.

Shamir, A., 1984. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology—CRYPTO 84, pp. 47–53.

Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J., 2003. Universal designated-verifier signatures. In: Advances in Cryptology—ASIACRYPT 2003, vol. 2894. Lecture Notes in Computer Science, Springer, pp. 523–542.

Steinfeld, R., Wang, H., Pieprzyk, J., 2004. Efficient extension of standard Schnorr/RSA signatures into universal designated-verifier signatures. In: Proceedings of Public Key Cryptography 2004, vol. 2947. Lecture Notes in Computer Science, Springer, pp. 86–100.

Susilo, W., Zhang, F., Mu, Y., 2004. Identity-based strong designated verifier signature schemes. In: Proceedings of 9 th Australasian Conference on Information Security and Privacy, ACISP 2004, vol. 3108. Lecture Notes in Computer Science, Springer, pp. 313–324.

Vergnaud, D., 2006. New extensions of pairing-based signatures into universal designated verifier signatures. In: Proceedings of 33 th International Colloquium on Automata, Languages andProgramming, ICALP 2006, vol. 4052. Lecture Notes in Computer Science, Springer, pp. 58–69.

Zhang, J., Mao, J., 2008. A novel id-based designated verifier signature scheme. Information Sciences 178 (3), 766–773.

Zhang, R., Furukawa, J., Imai, H., 2005. Short signature and universal designated verifier signature without random oracles. In: Proceedings of 3 rd International Conference on Applied Cryptography and Network Security, ACNS 2005, vol. 3531. Lecture Notes in Computer Science, Springer, pp. 483–498.

Qiong Huang got his B.S. and M.S. degrees from Fudan University in 2003 and 2006, respectively, and obtained Ph.D. degree from City University of Hong Kong in 2010. Now he is a senior research associate at Department of Computer Science, City University of Hong Kong. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis.

Guomin Yang received his PhD degree from the Computer Science Department at City University of Hong Kong in 2009. He is currently a research scientist in the Temasek Laboratories at National University of Singapore. His research interests are cryptography and network security.

Duncan S. Wong received the B.Eng. degree from the University of Hong Kong in 1994, the M.Phil. degree from the Chinese University of Hong Kong in 1998, and the Ph.D. degree from Northeastern University, Boston, MA, U.S.A. in 2002. He is currently an associate professor in the Department of Computer Science at the City University of Hong Kong.

Willy Susilo received the B.S. degree in the in Computer Science from Universitas Surabaya, Indonesia, in 1994, and the M.Comp.Sc. and Ph.D. degrees in Computer Science from University of Wollongong, Australia in 1996 and 2001, respectively. He is currently a Professor in the School of Computer Science and Software Engineering, University of Wollongong, Australia. He is also holding the prestigious ARC Future Fellowship awarded by the Australian Research Council. Prior to this role, he was the Head of school of School of Computer Science and Software Engineering and the Deputy Director of ICT Research Institute. He is the Director of Centre for Computer and Information Security Research. His research areas include cryptography, information security, computer security and network security, in particular the design of digital signature schemes.