Non-delegatable Identity-based Designated Verifier Signature

Qiong Huang[*] Duncan S. Wong[*] Willy Susilo[†]

July 22, 2009

Abstract

Designated verifier signature is a cryptographic primitive which allows a signer to convince a designated verifier of the validity of a statement but in the meanwhile prevents the verifier from transferring this conviction to any third party. In this work we present an identity-based designated verifier signature scheme that supports non-delegatability, and prove its security in the random oracle model, based on computational Diffie-Hellman assumption. Our scheme is perfectly non-transferable, and its non-delegatability follows the original definition proposed by Lipmaa et al. [LWB05].

Keywords. designated verifier signature, non-delegatability, non-transferability, random oracle model, signature scheme

[*] Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong S.A.R., China. Emails: csqhuang@student.cityu.edu.hk, duncan@cityu.edu.hk.

[†] School of Computer Science and Software Engineering, University of Wollongong, Northfields Avenue, New South Wales 2522, Australia. Email: wsusilo@uow.edu.au.

Contents

1 Introduction

Designated verifier signature (DVS in short), introduced by Jakobsson, Sako and Impagliazzo [JSI96], aims to allow an entity say, Alice, to prove that she has signed a document Θ to a specific entity say, Bob, in such a way that Bob is convinced about the fact but, unlike conventional digital signatures, he could not transfer this conviction to any third party. This property is called *non-transferability*, which is accomplished by empowering Bob the ability of producing signatures indistinguishable from those generated by Alice. After receiving a signature from Alice, Bob is sure about that Alice made the signature as he didn't do so. However, any third party only believes that either Alice or Bob is the signer of the signature. Designated verifier signature has applications in e-voting [JSI96], deniable authentication [WS09] and etc.

1.1 Related Work

Since the introduction of DVS [JSI96], there have been a lot of work on it and its variants. Jakobsson et al. [JSI96] proposed a stronger version of DVS, *strong designated verifier signature* (SDVS), in which only the verifier can verify the validity of a signature designated to him since the verification requires the secret key of the designated verifier. Steinfeld et al. [SBWP03] proposed the notion of *universal designated verifier signature* (UDVS), in which the holder of a signature can designate any third party as the designated verifier for checking the validity of the signature, but in the meanwhile, the designated verifier still could not convince others the source of the signature. Laguillaumie et al. studied other variants of designated verifier signatures [LV04b, LV04a], i.e. multi-designated verifiers signatures and etc. Later, Zhang et al. [ZFI05] proposed a UDVS scheme secure without random oracles based on Boneh-Boyen short signature [BB04]. Independently, Laguillaumie et al. [LLQ06] and Huang et al. [HSMW07] proposed (almost) the same UDVS schemes based on Waters signature [Wat05], which are also secure without random oracles. Vergnaud [Ver06] gave another two constructions of UDVS, one based on Boneh-Boyen short signature [BB04] and secure without random oracles but requiring a strong assumption named *knowledge-of-exponent assumption* [Dam92], and the other based on Boneh-Lynn-Shacham signature [BLS04] and secure in the random oracle model [BR93]. Recently Yu et al. [YXZL09] gave a construction of universal designated verifier proxy signature scheme without random oracles, which is essentially an extension of the schemes in [LLQ06, HSMW07].

Besides the aforementioned designated verifier signature schemes and variants in the conventional public key infrastructure (PKI) setting, another interesting and practically useful variant is *identity-based designated verifier signature* (IBDVS in short), which is a combination of DVS and identity-based cryptography [Sha84]. Susilo et al. [SZM04] studied DVS schemes in the identity-based setting and proposed an identity-based SDVS scheme based on bilinear Diffie-Hellman (BDH) assumption. Huang et al. [HSMZ08] also proposed a strong DVS scheme and an identity-based SDVS scheme based on Diffie-Hellman key exchange, which has very short signature size. Recently, Kang et al. [KBD09] proposed another identity-based SDVS scheme which is secure based BDH assumption. Cao et al. [CC09] proposed the first identity-based (universal) designated verifier signature scheme that is secure without the random oracles. Their scheme is based on Paterson-Schuldt identity-based signature scheme [PS06], which in turn is based Waters signature scheme [Wat05]. In essence, their scheme is the two-user version of the identity-based ring signature scheme proposed by Au et al. [ALYW06].

Lipmaa, Wang and Bao considered a new type of attacks against DVS schemes, i.e. *delegatability attack*, in which Alice or Bob could release a derivative of their secret key to any third party say Teddy, so that Teddy can produce signatures on behalf of Alice using this derivative. 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 have been shown to be vulnerable to delegatability attacks in [LWB05]. Besides those scheme, it is also easy to show that the identity-based schemes proposed in [HSMZ08, CC09, KBD09] are also vulnerable to this kind of attacks.

In 2006, Huang et al. [HSMW06] proposed a UDVS scheme which supports non-delegatability. However, their scheme is in the PKI setting. Recently, Zhang et al. [ZM08] proposed an identity-based SDVS 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 follow the definition proposed by Lipmaa et al. [LWB05]. What they actually proved is that if there is an algorithm which produces a signature with respect to the signer and the designated verifier without the signer or the verifier's secret key, there is another algorithm which solves the computational Diffie-Hellman (CDH) problem. However, it is unknown if there is an algorithm which can extract the secret key of either the signer or the verifier, given black-box oracle access to such a forger.

1.2 Our Work

In this work we propose another non-delegatable identity-based designated verifier signature scheme, which is based on Gentry et al.'s hierarchical identity-based encryption scheme [GS02]. Though our scheme does not outperform other schemes like [ZM08, CC09] in terms of signature size, the non-delegatability of our proposal *strictly* follows the original definition proposed by Lipmaa et al. [LWB05], 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. In addition, we prove that our scheme is unforgeable in the random oracle model assuming the hardness of CDH problem which is a widely used and well studied number-theoretic assumption. 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.

1.3 Paper Organization

In the next section we review the definition of IBDVS and its security model. Some mathematical background is given in Sec. 3. Our IBDVS scheme is then proposed in Sec. 4. We also prove its security with respect to the given security definitions in the random oracle in Sec. 5, along with a comparison between our scheme and other existing schemes. The paper is concluded in Sec. 6.

2 Identity-based Designated Verifier Signature

A designated verifier signature scheme [JSI96] consists of four (probabilistic) polynomial-time algorithms, one for key generation, one for the signer to sign with respect to a designated verifier, one for the designated verifier to simulate the signer's signature, and the other for verification. Identity-based designated verifier signature (IBDVS) is the analogy of DVS in the identity-based setting. Below is the formal definition of it.

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

- Setup*: The algorithm takes as input a security parameter* 1^k*, and outputs a master key pair for the PKG, i.e.* $(\texttt{mpk}, \texttt{msk}) \leftarrow \mathsf{Setup}(1^k)$*, where* $\texttt{mpk}$ *is published, and* $\texttt{msk}$ *is kept secret by the PKG.*

- $\mathsf{Extract}$*: The algorithm takes as input the master secret key* $\mathtt{msk}$ *and an identity* $\mathtt{id}$ *which can be a string of arbitrary length, and outputs the corresponding secret key* $\mathtt{usk_{id}}$ *for the user with identity* $\mathtt{id}$*, i.e.* $\mathtt{usk_{id}} \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id})$*.*

- Sign*: The algorithm takes as input the secret key of the signer* $\mathtt{usk}_S$*, the identity of the designated verifier* $\mathtt{id}_V$*, the master public key* $\mathtt{mpk}$ *and a message* $M \in \{0,1\}^*$*, and outputs a signature* σ*, i.e.* $\sigma \leftarrow \mathsf{Sign}(\mathtt{usk}_S, \mathtt{id}_V, \mathtt{mpk}, M)$*.*

- Ver*: The algorithm takes as input a message* M*, the identities of the signer and the verifier, i.e.* $\mathtt{id}_S, \mathtt{id}_V$*, the master public key* $\mathtt{mpk}$ *and a purported signature* σ*, and outputs a bit* b*, which is 1 for acceptance or 0 for rejection, i.e.* $b \leftarrow \mathsf{Ver}(M, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, \sigma)$*.*

- Sim*: The algorithm takes as the secret key of the verifier* $\mathtt{usk}_V$*, the identity of the signer* $\mathtt{id}_V$*, the master public key* $\mathtt{mpk}$ *and a message* M*, and outputs a signature* σ*, i.e.* $\sigma \leftarrow \mathsf{Sim}(\mathtt{usk}_V, \mathtt{id}_S, \mathtt{mpk}, M)$*.*

The *completeness* requires that for any $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{Setup}(1^k)$, any $\mathtt{id}_S, \mathtt{id}_V \in \{0,1\}^*$, $\mathtt{usk}_S \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_S)$, $\mathtt{usk}_V \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_V)$, any message $M \in \{0,1\}^*$, it holds that

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

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, $\mathsf{G^u}$, played between a game challenger C and a probabilistic polynomial-time adversary $\mathcal{A}$:

1. C runs the Setup algorithm to generate a master key pair $(\mathtt{mpk}, \mathtt{msk})$, and invokes $\mathcal{A}$ on input $\mathtt{mpk}$.

2. In this phase, the adversary can issue queries to the following oracles, for polynomial times:

   - $\mathcal{O}_\mathsf{E}$: Given a query $\mathtt{id}$ from $\mathcal{A}$, the oracle computes $\mathtt{usk_{id}} \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id})$, and returns $\mathtt{usk_{id}}$ to $\mathcal{A}$.
   - $\mathcal{O}_\mathsf{Sign}$: Given a query of the form $(\mathtt{id}_S, \mathtt{id}_V, M)$, the oracle first computes the secret key of $\mathtt{id}_S$ as $\mathtt{usk}_S \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_S)$, and signs M by computing $\sigma \leftarrow \mathsf{Sign}(\mathtt{usk}_S, \mathtt{id}_V, \mathtt{mpk}, M)$. It returns σ back to $\mathcal{A}$.
   - $\mathcal{O}_\mathsf{Sim}$: Given a query of the form $(\mathtt{id}_S, \mathtt{id}_V, M)$, the oracle first computes the secret key of $\mathtt{id}_V$ as $\mathtt{usk}_V \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_V)$, and signs M by computing $\sigma \leftarrow \mathsf{Sim}(\mathtt{usk}_V, \mathtt{id}_S, \mathtt{mpk}, M)$. It returns σ back to $\mathcal{A}$.

3. Finally, $\mathcal{A}$ outputs its forgery, $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, \sigma^*)$. It wins the game if

   (a) $1 \leftarrow \mathsf{Ver}(M^*, \mathtt{id}_S^*, \mathtt{id}_V^*, \mathtt{mpk}, \sigma^*)$;

   (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 $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*)$.

Definition 2.2 (Unforgeability)**.** *An IBDVS scheme is said to be* $(T, q_\mathsf{E}, q_\mathsf{Sign}, q_\mathsf{Sim}, \epsilon)$-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 wins the game with probability at lease* ϵ*.*

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, if the distinguisher does not know the signer's secret key. Formally, we consider the following definition.

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

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

That is, for any probabilistic polynomial-time distinguisher $\mathcal{D}$, for any $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{Setup}(1^k)$*, any identities* $\mathtt{id}_S, \mathtt{id}_V \in \{0,1\}^*$*, any message $M \in \{0,1\}^*$, let* $\mathtt{usk}_S \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_S)$ *and* $\mathtt{usk}_V \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_V)$*, it holds that*

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

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 IBDVS scheme is perfectly non-transferable.

Remark 1 : 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 much 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 [LWB05] to the identity-based setting.

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

> *For every* $(\mathtt{mpk}, \mathtt{msk}) \leftarrow \mathsf{Setup}(1^k)$*, every* $\mathtt{id}_S, \mathtt{id}_V \in \{0,1\}^*$*, every* $\mathtt{usk}_S \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_S)$*,* $\mathtt{usk}_V \leftarrow \mathsf{Extract}(\mathtt{msk}, \mathtt{id}_V)$*, 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 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* $\mathtt{usk}_S$ *or* $\mathtt{usk}_V$ *in expected time $T \cdot (\epsilon - \kappa)^{-1}$, 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.*

Remark 2 : We stress that if the IBDVS scheme is provably secure in the random oracle model, all the adversaries in games of unforgeability, non-transferability and non-delegatability have access to the random oracles. The definitions of the three security properties are modified accordingly to take into account the numbers of queries to the random oracles issued by the adversaries.

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

3 Mathematical Background

(Admissible Pairings): Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of large prime order p. The mapping $\mathsf{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be an *admissible pairing*, if

- *Bilinearity*: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $\mathsf{e}(u^a, v^b) = \mathsf{e}(u, v)^{ab}$;
- *Non-degeneracy*: $\exists u, v \in \mathbb{G}$ such that $\mathsf{e}(u, v) \neq 1_T$, where 1_T is the identity element of $\mathbb{G}_T$; and
- *Computability*: there exists an efficient algorithm for computing $\mathsf{e}(u, v)$ 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 $\mathbb{G}$. The computational Diffie-Hellman (CDH) problem is as follows:

Given g, g^a, g^b for some random $a, b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, compute g^{ab}.

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

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

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

4 Our Non-delegatable IBDVS

In this section we propose an identity-based designated verifier signature scheme which is *non-delegatable*. Before proposing the scheme, we first briefly discuss the difficulty in constructing an IBDVS scheme.

To the best of our knowledge, all the identity-based (strong) designated verifier signature schemes use bilinear pairings. These schemes either use a common secret key shared between the signer and the designated verifier to produce a signature, i.e. [HSMZ08, KBD09, CC09], thus impossible for one to extract the user secret key from a signature, or use too many blind factors to hide the user secret key, i.e. [SZM04, ZM08], thus infeasible for one to recover the key.

Based on the observation, we employ a different method in constructing IBDVS schemes. Our scheme is based Gentry-Silverberg HIBE scheme [GS02], in which there is only one blind factor for hiding the user secret key. A signature of user with identity $\mathtt{id}$ on message M is $\sigma = (S_1, S_2) = (\mathtt{H}_1(\mathtt{id})^\alpha \cdot \mathtt{H}_2(M)^r, g^r)$, where $\mathtt{H}_1(\mathtt{id})^\alpha$ is the user secret key. A signature of user $\mathtt{id}$ is verified as $\mathsf{e}(S_1, g) \stackrel{?}{=} \mathsf{e}(\mathtt{H}_1(\mathtt{id}), g^\alpha) \cdot \mathsf{e}(\mathtt{H}_2(M), S_2)$ where g^α is the master public key. If we do not include $S_2 = g^r$ in the signature, but instead set S_2 to be a non-interactive proof of knowledge of the randomness r showing that S_1 is binding to either the signer or the designated verifier, the signature becomes a designated verifier signature. Moreover, given an adversary which forges a signature, we can run the extractor of the proof of knowledge to extract the randomness r from S_2, and then get the secret key by removing the factor $\mathtt{H}_2(M)^r$.

4.1 The Scheme

Our construction of IBDVS works as follows:

- * $\mathsf{Setup}(1^k)$: 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 $\mathtt{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. It selects at random $\alpha \overset{\$}{\leftarrow} \mathbb{Z}_p$, sets $g_1 = g^\alpha$, and selects three collision-resistant hash functions, $\mathtt{H}_1 : \{0,1\}^* \to \mathbb{G}$, $\mathtt{H}_2 : \{0,1\}^* \to \mathbb{G}\backslash\{1\}$ and $\mathtt{H}_3 : (\{0,1\}^*)^3 \times \mathbb{G} \times \mathbb{G}_T^2 \to \mathbb{Z}_p$, which will be modeled as random oracles in the security proofs. The master public key is set to be $\mathtt{mpk} = (g, g_1, \mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_3)$, and the master secret key is $\mathtt{msk} = \alpha$.

- * $\mathsf{Extract}(\mathtt{msk}, \mathtt{id})$: The secret key of a user with identity $\mathtt{id}$ is set to be $\mathtt{usk}_{\mathtt{id}} = \mathtt{H}_1(\mathtt{id})^\alpha$.

- * $\mathsf{Sign}(\mathtt{usk}_S, \mathtt{id}_V, \mathtt{mpk}, M)$: To sign a message M with respect to the designated verifier (with identity $\mathtt{id}_V$), the signer (with identity $\mathtt{id}_S$) does as follows:

  1. Choose at random $r \overset{\$}{\leftarrow} \mathbb{Z}_p$.
  2. Set $S_1 = \mathtt{usk}_S \cdot \mathtt{H}_2(M)^r$. Using r and hash function $\mathtt{H}_3$, compute the following proof of knowledge:

$$S_2 = PK\left\{\beta \;:\; \mathtt{e}(\mathtt{H}_2(M), g)^\beta = \frac{\mathtt{e}(S_1, g)}{\mathtt{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)} \bigvee \mathtt{e}(\mathtt{H}_2(M), g)^\beta = \frac{\mathtt{e}(S_1, g)}{\mathtt{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)}\right\}(\bar{M}) \tag{1}$$

  where $\bar{M} = (\mathtt{id}_S, \mathtt{id}_V, M, S_1)$. Set $\sigma = (S_1, S_2)$. In Sec. 4.2 we give the details in the generation and verification of S_2, which includes two elements of $\mathbb{G}_T$ and three elements of $\mathbb{Z}_p$.

- * $\mathsf{Ver}(M, \mathtt{id}_S, \mathtt{id}_V, \mathtt{mpk}, \sigma)$: After receiving a signature $\sigma = (S_1, S_2)$ and a message M from the signer (with identity $\mathtt{id}_S$), the verifier (with identity $\mathtt{id}_V$) checks the validity of the proof of knowledge S_2 with respect to S_1. It accepts if the proof of knowledge is valid, and rejects otherwise.

- * $\mathsf{Sim}(\mathtt{usk}_V, \mathtt{id}_S, \mathtt{mpk}, M)$: To simulate a signature on M, the verifier does as the signer, except that S_1 is computed as $S_1 = \mathtt{usk}_V \cdot \mathtt{H}_2(M)^r$.

It's easy to see that the scheme is complete. Details can be found in Sec. 4.2.

Efficiency: In our IBDVS scheme a signature comprises of one element of $\mathbb{G}$, two elements of $\mathbb{G}_T$ and three elements of $\mathbb{Z}_p$. The signing algorithm and the simulation algorithm involves three pairing evaluations, one exponentiation in $\mathbb{G}$ and three exponentiations in $\mathbb{G}_T$. The verification algorithm involves four pairing evaluations and four exponentiations in $\mathbb{G}_T$.

4.2 Details of Generation and Verification of (1)

To generate (1), the signer does as the following:

1. Choose $r_0, e_1, z_1 \overset{\$}{\leftarrow} \mathbb{Z}_p$.

2. Set $R_0 = \mathtt{e}(\mathtt{H}_2(M), g)^{r_0}$ and

$$R_1 = \frac{\mathtt{e}(\mathtt{H}_2(M), g)^{z_1}}{(\mathtt{e}(S_1, g)/\mathtt{e}(H_1(\mathtt{id}_V), g_1))^{e_1}}$$

3. Set $e = \mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, R_0, R_1)$.

4. Set $e_0 = e - e_1$, $z_0 = r_0 + \beta e_0$.

The proof of knowledge S_2 is set to be $S_2 = (R_0, e_0, z_0, R_1, z_1)$[2].

A designated verifier with identity $\mathtt{id}_V$ 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 $S_2 = (R_0, e_0, z_0, R_1, z_1)$, the verifier does as the following:

1. Compute $e_1 = \mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, R_0, R_1) - e_0$.
2. Check if

$$\mathsf{e}(\mathtt{H}_2(M), g)^{z_0} \overset{?}{=} R_0 \cdot \left(\frac{\mathsf{e}(S_1, g)}{\mathsf{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)}\right)^{e_0} \quad (2)$$

$$\mathsf{e}(\mathtt{H}_2(M), g)^{z_1} \overset{?}{=} R_1 \cdot \left(\frac{\mathsf{e}(S_1, g)}{\mathsf{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)}\right)^{e_1} \quad (3)$$

It accepts if both of the equations above hold, and rejects otherwise.

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

$$R_0 = \frac{\mathsf{e}(\mathtt{H}_2(M), g)^{z_0}}{(\mathsf{e}(S_1, g)/\mathsf{e}(H_1(\mathtt{id}_V), g_1))^{e_0}} \quad \text{and} \quad R_1 = \frac{\mathsf{e}(\mathtt{H}_2(M), g)^{z_1}}{(\mathsf{e}(S_1, g)/\mathsf{e}(H_1(\mathtt{id}_V), g_1))^{e_1}}$$

and then patches the random oracle $\mathtt{H}_3$ with $((\mathtt{id}_S, \mathtt{id}_V, M, R_0, R_1), e)$, i.e. setting $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, R_0, R_1) = e$. It's 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 $(\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1)$ 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. (2) we have that

$$z_0 = r_0 + e_0\beta_0 \quad \text{and} \quad z'_0 = r_0 + e'_0\beta_0$$

Then β_0 can be obtained by computing

$$\beta_0 = \frac{z_0 - z'_0}{e_0 - e'_0}$$

It can be verified that

$$\frac{\mathsf{e}(S_1, g)}{\mathsf{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)} = \mathsf{e}(\mathtt{H}_2(M), g)^{\beta_0}$$

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

$$\frac{\mathsf{e}(S_1, g)}{\mathsf{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)} = \mathsf{e}(\mathtt{H}_2(M), g)^{\beta_1}$$

[2] Actually, one can set $S_2 = (e_0, z_0, e_1, z_1)$ which has smaller size. However, for the sake of the simplicity in the security proofs, we choose to include the R values in S_2.

5 Security Proofs

Informally, since the group $\mathbb{G}$ is of prime order p, $\mathtt{H}_2(M)^r$ generates the whole group. Therefore, $\mathtt{H}_1(\mathtt{id}_S)^\alpha$ is perfectly hidden by $\mathtt{H}_2(M)^r$. That is, the distribution of $\mathtt{H}_1(\mathtt{id}_S)^\alpha\mathtt{H}_2(M)^r$ is identical to that of $\mathtt{H}_1(\mathtt{id}_V)^\alpha\mathtt{H}_2(M)^r$. In addition, the proof of knowledge S_2 is perfectly witness indistinguishable. In a consequence, the signature produced by the signer is perfectly indistinguishable from that by the verifier.

To see the non-delegatability, we can construct an extractor which controls the output of the random oracle $\mathtt{H}_2$. The validity of a signature indicates that either the secret key of $\mathtt{id}_S$ or that of $\mathtt{id}_V$ is contained in S_1. If an adversary outputs a valid signature with respect to $\mathtt{id}_S, \mathtt{id}_V$, the extractor can first extract the witness r encapsulated in S_2 by rewinding the adversary to some previous status, and then remove the factor $\mathtt{H}_2(M)^r$ from S_1.

Theorem 5.1. *If CDH assumption (T, ϵ) holds in $\mathbb{G}$, the IBDVS scheme above is $(T', q_{\mathtt{H}_1}, q_{\mathtt{H}_2}, q_{\mathtt{H}_3}, q_{\mathsf{E}}, q_{\mathsf{Sign}}, q_{\mathsf{Sim}}, \epsilon')$-unforgeable, where*

$$T' = \Theta(T), \qquad \epsilon' < \frac{10\mathfrak{e}^2 q_{\mathsf{E}}^2 \sqrt{q_{\mathtt{H}_3}}}{9} \cdot \sqrt{\epsilon}$$

and $\mathfrak{e}$ is the natural logarithm.

Proof. Given an adversary $\mathcal{A}$ against the unforgeability of the IBDVS scheme with success probability ϵ', we use it to build another algorithm $\mathcal{B}$ for solving the CDH problem with success probability ϵ. 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 : $\mathcal{B}$ chooses three collision-resistant hash functions $\mathtt{H}_1, \mathtt{H}_2$ and $\mathtt{H}_3$ as required by the scheme, and invokes the adversary $\mathcal{A}$ on input $\mathtt{mpk} = (g, g_1, \mathtt{H}_1, \mathtt{H}_2, \mathtt{H}_3)$. Note that the master secret key $\mathtt{msk} = \log_g g_1 = a$ is unknown to $\mathcal{B}$.

Qeury : $\mathcal{B}$ simulates the following oracles for $\mathcal{A}$ by maintaining three hash tables, HT_1, HT_2 and HT_3.

* $\mathtt{H}_1$ *Query*: Given a query $\mathtt{id}$, if there is an entry starting with $\mathtt{id}$ in table HT_1, $\mathcal{B}$ retrieves the corresponding value $\mathtt{H}_1(\mathtt{id})$ from HT_1, and returns it. Otherwise, $\mathcal{B}$ tosses a coin c so that $\Pr[c = 1] = \delta$ which will be determined later. If $c = 0$, $\mathcal{B}$ chooses at random $t \xleftarrow{\$} \mathbb{Z}_p$, and sets $\mathtt{H}_1(\mathtt{id}) = g_2^t$; otherwise, it chooses at random $t \xleftarrow{\$} \mathbb{Z}_p$, and sets $\mathtt{H}_1(\mathtt{id}) = g^t$. In either, $\mathcal{B}$ stores the tuple $(\mathtt{id}, \mathtt{H}_1(\mathtt{id}), c, t)$ into table HT_1, and returns $\mathtt{H}_1(\mathtt{id})$ back to $\mathcal{A}$.

* $\mathtt{H}_2$ *Query*: Given an input M, if there is an entry starting with it in table HT_2, $\mathcal{B}$ retrieves the corresponding answer $\mathtt{H}_2(M)$ from the table and returns it. Otherwise, $\mathcal{B}$ returns $m \leftarrow \mathtt{H}_2(M)$. It stores $(M, \mathtt{H}_2(M))$ into table HT_2, and returns the hash value.

* $\mathtt{H}_3$ *Query*: Given an input $(\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1)$, if there is an entry starting with it in table HT_3, $\mathcal{B}$ retrieves the corresponding answer $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1)$ from the table and returns it. Otherwise, $\mathcal{B}$ chooses at random $e \xleftarrow{\$} \mathbb{Z}_p$, and sets $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, R_0, R_1) = e$. It stores $((\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1), \mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1))$ into table HT_3, and returns the hash value.

* $\mathsf{Extract}$ *Query*: Given an identity $\mathtt{id}$, $\mathcal{B}$ retrieves the corresponding tuple $(\mathtt{id}, \mathtt{H}_1(\mathtt{id}), c, t)$ from HT_1. If $c = 1$, $\mathcal{B}$ computes the user secret key $\mathtt{usk}_{\mathtt{id}} = g_1^t$ and returns it to $\mathcal{A}$. If $c = 0$, $\mathcal{B}$ aborts.

- Sign *Query*: Given a query $(\mathtt{id}_S, \mathtt{id}_V, M)$, $\mathcal{B}$ retrieves the tuple $(\mathtt{id}_S, \mathtt{H}_1(\mathtt{id}_S), c, t)$ from HT_1. We distinguish two cases:
  - If $c = 1$, $\mathcal{B}$ generates the user secret key of $\mathtt{id}_S$ as in the simulation of $\mathsf{Extract}$ oracle, and then computes the signature σ by running the Sign algorithm on input $(\mathtt{usk}_S, \mathtt{id}_V, \mathtt{mpk}, M)$.
  - If $c = 0$, $\mathcal{B}$ randomly selects $S_1 \xleftarrow{\$} \mathbb{G}$. Note that there exists some r (unknown to $\mathcal{B}$) such that $S_1 = \mathtt{H}_1(\mathtt{id}_S)^a \cdot \mathtt{H}_2(M)^r$. $\mathcal{B}$ then simulates the proof of knowledge S_2 in the way specified in Sec. 4.2. In case there is a collision when patching the oracle $\mathtt{H}_3$, $\mathcal{B}$ aborts. This event occurs only with probability at most $((q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3})/p$. $\mathcal{B}$ returns $\sigma = (S_1, S_2)$ to $\mathcal{A}$.
- Sim *Query*: This kind of queries can be answered by $\mathcal{B}$ in a similar way with that above. The difference is that $\mathcal{B}$ generates the signature from the point of the designated verifier.

Forge : Finally, $\mathcal{A}$ outputs its forgery, $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, \sigma^*)$ where $\sigma^* = (S_1^*, S_2^* = (R_0^*, e_0^*, z_0^*, R_1^*, z_1^*))$. Suppose that $\mathcal{A}$ wins the game. (Otherwise $\mathcal{B}$ aborts.) $\mathcal{B}$ retrieves the two tuples $(\mathtt{id}_S^*, \mathtt{H}_1(\mathtt{id}_S^*), c_S, t_S)$ and $(\mathtt{id}_V^*, \mathtt{H}_1(\mathtt{id}_V^*), c_V, t_V)$ from HT_1. If $c_S = 1$ or $c_V = 1$, $\mathcal{B}$ aborts. It also retrieves the tuple $(M^*, \mathtt{H}_2(M^*))$ from HT_2, and the tuple $((\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, R_0^*, R_1^*), \mathtt{H}_3(\mathtt{id}_S^*, \mathtt{id}_V^*, S_1^*, M^*, R_0^*, R_1^*))$ from HT_3. Let $e^* = \mathtt{H}_3(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, R_0^*, R_1^*)$. Next, $\mathcal{B}$ rewinds $\mathcal{A}$ to the status of querying oracle $\mathtt{H}_3$ on input $(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, R_0^*, R_1^*)$. It chooses at random $e'^* \neq e^* \in \mathbb{Z}_p$ and answers $\mathcal{A}$ with e'^*. $\mathcal{B}$ then continues to simulates oracles as above for $\mathcal{A}$. Suppose that again, $\mathcal{A}$ outputs a successful forgery, say $(\mathtt{id}'^*_S, \mathtt{id}'^*_V, M'^*, \sigma'^*)$ where $\sigma'^* = (S'^*_1, S'^*_2 = (R'^*_0, e'^*_0, z'^*_0, R'^*_1, z'^*_1))$. If $(\mathtt{id}'^*_S, \mathtt{id}'^*_V, M'^*, S'^*_1, R'^*_0, R'^*_1) \neq (\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, R_0^*, R_1^*)$, $\mathcal{B}$ aborts. Otherwise, it runs the extractor (described in Sec. 4.2) to extract the secret randomness r^* from (S_2^*, S'^*_2).

- If $\mathsf{e}(\mathtt{H}_2(M^*), g)^{r^*} = \mathsf{e}(S_1^*, g)/\mathsf{e}(\mathtt{H}_1(\mathtt{id}_S^*), g_1)$, we have that $S_1^* = \mathtt{H}_1(\mathtt{id}_S^*)^a \cdot \mathtt{H}_2(M^*)^{r^*}$. Recall that $\mathtt{H}_1(\mathtt{id}_S^*) = g_2^{t_S}$. $\mathcal{B}$ then can recover g^{ab} from S_1^* by computing

$$g^{ab} = \left(\frac{S_1^*}{\mathtt{H}_2(M^*)^{r^*}}\right)^{\frac{1}{t_S}}$$

- If $\mathsf{e}(\mathtt{H}_2(M^*), g)^{r^*} = \mathsf{e}(S_1^*, g)/\mathsf{e}(\mathtt{H}_1(\mathtt{id}_V^*), g_1)$, we have that $S_1^* = \mathtt{H}_1(\mathtt{id}_V^*)^a \cdot \mathtt{H}_2(M^*)^{r^*}$. Recall that $\mathtt{H}_1(\mathtt{id}_V^*) = g_2^{t_V}$. $\mathcal{B}$ then can recover g^{ab} from S_1^* by computing

$$g^{ab} = \left(\frac{S_1^*}{\mathtt{H}_2(M^*)^{r^*}}\right)^{\frac{1}{t_V}}$$

In either case $\mathcal{B}$ obtains the solution to the given instance of CDH problem.

Probability Analysis: In the process of solving the CDH problem above, there are some cases in which $\mathcal{B}$ aborts.

1. A collision occurs when patching the oracle $\mathtt{H}_3$. This does not happen with probability at least $1 - ((q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3})/p$.

2. $\mathcal{A}$ issues an $\mathsf{Extract}$ query on input an identity $\mathtt{id}$ whose corresponding c value (stored in HT_1) is 0. This event does not happen with probability $\delta^{q_{\mathsf{E}}}$.

3. Conditioned on that $\mathcal{B}$ does not abort in the simulation of oracles, $\mathcal{A}$ fails in outputting its forgery. This event does not happen with probability ϵ' due to the perfect simulation of the oracles.

4. Conditioned on that $\mathcal{B}$ does not abort in the simulation of oracles and $\mathcal{A}$ succeeds in outputting its forgery, either of the two identities, i.e. $\mathtt{id}_S^*, \mathtt{id}_V^*$, has the corresponding c value being 1. This does not happen with probability $(1-\delta)^2$.

Therefore, in the first run of $\mathcal{A}$, $\mathcal{B}$ does not abort with probability at least

$$\varepsilon \geq \left(1 - \frac{(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3}}{p}\right) \cdot \delta^{q_{\mathsf{E}}} \cdot (1-\delta)^2 \cdot \left(\epsilon' - \frac{1}{p}\right)$$

where $1/p$ stems from that $\mathcal{A}$ obtains $\mathtt{H}_3(\mathtt{id}_S^*, \mathtt{id}_V^*, M^*, S_1^*, R_0^*, R_1^*)$ without querying the oracle $\mathtt{H}_3$. A similar analysis with that in [PS00, BBS04] shows that with probability at least

$$\epsilon \geq \frac{\varepsilon^2}{16 q_{\mathtt{H}_3}} = \frac{\left(1 - \frac{(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3}}{p}\right)^2 \cdot (\delta^{q_{\mathsf{E}}} \cdot (1-\delta)^2)^2 \cdot \left(\epsilon' - \frac{1}{p}\right)^2}{16 q_{\mathtt{H}_3}}$$

$\mathcal{A}$ outputs a successful forgery that satisfies the aforementioned conditions, which, together with the successful output in the first run, enables $\mathcal{B}$ to solve the given CDH problem. This probability is maximized when $\delta = \frac{q_{\mathsf{E}}}{q_{\mathsf{E}}+2}$. Thus, we get that

$$\begin{aligned} \epsilon &\geq \left(1 - \frac{(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3}}{p}\right)^2 \cdot \frac{\left(\left(1 - \frac{2}{q_{\mathsf{E}}+2}\right)^{q_{\mathsf{E}}} \cdot \left(\frac{2}{q_{\mathsf{E}}+2}\right)^2\right)^2 \cdot \left(\epsilon' - \frac{1}{p}\right)^2}{16 q_{\mathtt{H}_3}} \\ &\approx \left(1 - \frac{(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3}}{p}\right)^2 \cdot \frac{1}{q_{\mathtt{H}_3} \cdot q_{\mathsf{E}}^4 \cdot \mathfrak{e}^4} \cdot \left(\epsilon' - \frac{1}{p}\right)^2 \end{aligned}$$

where $\mathfrak{e}$ is the natural logarithm. Hence,

$$\epsilon' \leq \frac{\mathfrak{e}^2 \cdot q_{\mathsf{E}}^2 \cdot \sqrt{q_{\mathtt{H}_3}}}{\left(1 - \frac{(q_{\mathsf{Sign}} + q_{\mathsf{Sim}})^2 + (q_{\mathsf{Sign}} + q_{\mathsf{Sim}})q_{\mathtt{H}_3}}{p}\right)} \cdot \sqrt{\epsilon} + \frac{1}{p} < \frac{10 \mathfrak{e}^2 q_{\mathsf{E}}^2 \sqrt{q_{\mathtt{H}_3}}}{9} \cdot \sqrt{\epsilon}$$

This completes the proof. □

Remark 3 : In the proof above, hash functions $\mathtt{H}_1$ and $\mathtt{H}_3$ are modeled as programmable random oracles, while $\mathtt{H}_2$ is modeled as a non-programmable random oracle.

Theorem 5.2. *The IBDVS scheme is perfectly non-transferable (see Def. 2.3).*

Proof. Note that the first component in a signature is of the form $S_1 = \mathtt{usk} \cdot \mathtt{H}_2(M)^r$ for some r randomly chosen from $\mathbb{Z}_p$, where $\mathtt{usk}$ is either $\mathtt{usk}_S$ or $\mathtt{usk}_V$. 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 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 S_1, there exist $r, r' \in \mathbb{Z}_p$ such that $S_1 = \mathtt{usk}_S \cdot \mathtt{H}_2(M)^r = \mathtt{usk}_V \cdot \mathtt{H}_2(M)^{r'}$. On the other hand, the proof of knowledge S_2 is perfectly witness indistinguishable, thus revealing no information about the randomness r. In a consequence, the signature $\sigma = (S_1, S_2)$ is information-theoretically hiding. □

Theorem 5.3. *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 IBDVS scheme is* $(56T/\epsilon, 1/p)$*-non-delegatable (see Def. 2.4) in the random oracle model.*

Scheme	**Type**	**Signature-Size**	**Non-Trans**	**Non-Dele**	**RO**	**Assump**
Ours	IBDVS	$1\mathbb{G} + 2\mathbb{G}_T + 3\mathbb{Z}_p$	perfect	$\surd$	$\surd$	CDH
[CC09]	IBUDVS	$4\mathbb{G}$	perfect	$\times$	$\times$	CDH
[HSMZ08]	IBSDVS	$\mathbb{Z}_p^*$	perfect	$\times$	$\surd$	Gap-BDH
[KBD09]	IBSDVS	$2\mathbb{G}_T$	perfect	$\times$	$\surd$	BDH
[SZM04]	IBSDVS	$1\mathbb{G} + 1\mathbb{Z}_p + 1\mathbb{Z}_p^*$	perfect	?	$\surd$	BDH
[ZM08]	IBSDVS	$3\mathbb{G}$	perfect	?	$\surd$	BDH

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

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, except that $\mathcal{K}$ returns different random values (e versus e') as the answer to the hash query $\mathtt{H}_3(\mathtt{id}_S, \mathtt{id}_V, M, S_1, R_0, R_1)$. Since S_1, R_0, R_1 are in the input to the hash function, their values must be equal in both runs. If both signatures, i.e. $(S_1, S_2 = (R_0, e_0, z_0, R_1, z_1))$ and $(S_1, S_2' = (R_0, e_0', z_0', R_1, z_1'))$, are valid, one can call the extractor of the proof of knowledge (described in Sec. 4.2) to extract the randomness r from (S_2, S_2'). If $\mathsf{e}(\mathtt{H}_2(M), g)^r = \mathsf{e}(S_1, g)/\mathsf{e}(\mathtt{H}_1(\mathtt{id}_S), g_1)$, one can find $\mathtt{usk}_S = S_1/\mathtt{H}_2(M)^r$. If $\mathsf{e}(\mathtt{H}_2(M), g)^r = \mathsf{e}(S_1, g)/\mathsf{e}(\mathtt{H}_1(\mathtt{id}_V), g_1)$, one can find $\mathtt{usk}_V = S_1/\mathtt{H}_2(M)^r$.

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 $(S_1, S_2 = (R_0, e_0, z_0, R_1, z_1))$ and $(S_1', S_2' = (R_0', e_0', z_0', R_1', z_1'))$ on M with respect to $\mathtt{id}_S, \mathtt{id}_V$, such that $(S_1, R_0, R_1) = (S_1', R_0', R_1')$. Then one can compute $\mathtt{usk}_S$ or $\mathtt{usk}_V$ with probability 1. Thus, given that algorithm Rewind runs in expected time $56/\epsilon$, 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 $\mathtt{H}_1, \mathtt{H}_2$). Let $\mathbf{H}$ be a matrix with a row for each possible set of random coins for $\mathcal{F}_{S,V,M}$, 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 $\mathbf{H}$, 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 $\mathbf{H}$. Using an algorithm from [DF02], Rewind can find such 1-entries in time $56/\epsilon$. □

Disavowability: Since our IBDVS is perfectly non-transferable, given a signature, the signer is unable to disavow that it is the real signer, though it is possible for the signer to confirm the fact.

Comparison: In Table 1 we give a comparison of our scheme with those existing identity-based DVS schemes, where **Non-Trans** indicates the level of non-transferability, **Non-Dele** indicates if the scheme is non-delegatable under the definition of [LWB05], **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 the **Non-Dele** column means that it is unknown whether the scheme can be proved to be (non-)delegatable strictly under the definition in [LWB05].

6 Conclusion

In this work we proposed a new efficient non-delegatable identity-based designated verifier signature scheme. The scheme was proved to be unforgeable based on CDH assumption in the random oracle model, and be perfectly non-transferable. Though our scheme has slightly larger signature size than previous works, it is the first *identity-based DVS* scheme whose non-delegatability strictly follows the definition proposed by Lipmaa et al. [LWB05].

References

[ALYW06] Man Ho Au, Joseph K. Liu, Tsz Hon Yuen, and Duncan S. Wong. Id-based ring signature scheme secure in the standard model. In *Proceedings of 1st International Workshop on Security, IWSEC 2006*, volume 4266 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2006.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[CC09] Feng Cao and Zhenfu Cao. An identity based universal designated verifier signature scheme secure in the standard model. *The Journal of Systems and Software*, 82(4):643–649, 2009.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Advances in Cryptology - CRYPTO 91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, 1992.

[DF02] Ivan Damgård and Eiichiro Fujisaki. An integer commitment scheme based on groups with hidden order. In *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2002.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[HSMW06] Xinyi Huang, Willy Susilo, Yi Mu, and Wei Wu. Universal designated verifier signature without delegatability. In *Proceedings of 8th International Conference on Information and Communications Security, ICICS 2006*, volume 4307 of *Lecture Notes in Computer Science*, pages 479–498. Springer, 2006.

[HSMW07] Xinyi Huang, Willy Susilo, Yi Mu, and Wei Wu. Secure universal designated verifier signature without random oracles. *International Journal of Information Security*, 7(3):171–183, 2007.

[HSMZ08] Xinyi Huang, Willy Susilo, Yi Mu, and Futai Zhang. Short designated verifier signature scheme and its identity-based variant. *International Journal of Network Security*, 6(1):82–93, 2008.

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology - EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 143 – 154. Springer, 1996.

[KBD09] Baoyuan Kang, Colin Boyd, and Ed Dawson. A novel identity based strong designated verifier signature scheme. *The Journal of Systems and Software*, 82(2):270–273, 2009.

[LLQ06] Fabien Laguillaumie, Benoit Libert, and Jean-Jacques Quisquater. Universal designated verifier signatures without random oracles or non-black box assumptions. In *Proceedings of 5th International Conference on Security and Cryptography for Networks, SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2006.

[LV04a] Fabien Laguillaumie and Damien Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In *Proceedings of 4th International Conference on Security in Communication Networks, SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 105–119. Springer, 2004.

[LV04b] Fabien Laguillaumie and Damien Vergnaud. Multi-designated verifiers signatures. In *Proceedings of 6th International Conference on Information and Communications Security, ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 495–507. Springer, 2004.

[LWB05] Helger Lipmaa, Guilin Wang, and Feng Bao. Designated verifier signature schemes: Attacks, new security notions and a new construction. In *Proceedings of 32th International Colloquium on Automata, Languages andProgramming, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 459–471. Springer, 2005.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[PS06] Kenneth G. Paterson and Jacob C.N. Schuldt. Efficient identity-based signature secure in the standard model. In *Proceedings of 11th Australasian Conference on Information Security and Privacy, ACISP 2006*, volume 4058 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2006.

[SBWP03] Ron Steinfeld, Laurence Bull, Huaxiong Wang, and Josef Pieprzyk. Universal designated-verifier signatures. In *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 523–542. Springer, 2003.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO 84*, pages 47–53, 1984.

[SZM04] Willy Susilo, Fangguo Zhang, and Yi Mu. Identity-based strong designated verifier signature schemes. In *Proceedings of 9th Australasian Conference on Information Security and*

Privacy, ACISP 2004, volume 3108 of *Lecture Notes in Computer Science*, pages 313–324. Springer, 2004.

[Ver06] Damien Vergnaud. New extensions of pairing-based signatures into universal designated verifier signatures. In *Proceedings of* 33*th International Colloquium on Automata, Languages andProgramming, ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 58–69. Springer, 2006.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.

[WS09] Bin Wang and Zhaoxia Song. A non-interactive deniable authentication scheme based on designated verifier proofs. *Information Sciences*, 179(6):858–865, 2009.

[YXZL09] Yong Yu, Chunxiang Xu, Xiaosong Zhang, and Yongjian Liao. Designated verifier proxy signature scheme without random oracles. *Computers and Mathematics with Applications*, 57(8):1352–1364, 2009.

[ZFI05] Rui Zhang, Jun Furukawa, and Hideki Imai. 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*, volume 3531 of *Lecture Notes in Computer Science*, pages 483–498. Springer, 2005.

[ZM08] Jianhong Zhang and Jane Mao. A novel ID-based designated verifier signature scheme. *Information Sciences*, 178(3):766–773, 2008.