A Non-delegatable Identity-based Designated Verifier Signature Scheme without Bilinear Pairings

Maryam Rajabzadeh Asaar[1], Mahmoud Salmasizadeh [2]
[1] Department of Electrical Engineering, [2] Electronics Research Institute, Sharif University of Technology, Tehran, Iran.
asaar@ee.sharif.ir,salmasi@sharif.edu

Abstract. Up to now, several non-delegatable identity-based (strong) designated verifier signature schemes using bilinear pairings are proposed. In these identity-based (strong) designated verifier signature schemes, bilinear pairings are employed either in signing and verifying steps or only in the verifying step. However, the computation cost of pairings at a security level equivalent to a 128-bit symmetric key of AES is approximately 20 times higher than that of exponentiation over an elliptic curve group. Hence, presenting a (strong) designated verifier signature scheme which is identity-based without pairings and supports non-delegatability as well is vital. In this study, a non-delegatable identity-based designated verifier signature scheme without bilinear pairings using two concatenated Schnorr signatures is proposed. Our construction not only is approximately 40 times more efficient compared to the existing non-delegatable identity-based (strong) designated verifier signature schemes due to the avoiding bilinear pairings but also it is provable secure in the random oracle.

Keywords : designated verifier signature, bilinear pairings, identity-based, random oracle model.

1 Introduction

Jakobsson et al. [21] introduced the notion of designated verifier proofs (DVP) in 1996. These proofs allow a signer (Alice) to designate a verifier (Bob) and prove the validity of a statement only to Bob; while Bob cannot use this transcript to convince anyone else. This motivates non-transferability and is generally achieved by proving either the validity of the statement or the knowledge of Bob's secret key. Consequently, Bob can always generate the same transcript. A designated verifier signature (DVS) is the non-interactive version of the DVP. A DVS is publicly verifiable and a valid DVS is generated by Alice or Bob. The DVS is applied in various cryptographic schemes such as voting [21], undeniable signature [10, 14, 16], deniable authentication [36] where it is required that only designated entities can be convinced of several statements. It is desirable that a third party except Alice and Bob cannot tell whose signature is sent to Bob. A DVS with this property is called a strong designated verifier signature (SDVS)[21]. The strength of a SDVS as privacy of a signer's identity (PSI) is formalized by Laguillamie and Vergnand in 2004 [25]. A valid designated verifier signature for Bob on behalf of Alice is generated if and only if the secret key of either Alice or Bob is known. This property means non-delegatability for signing and is introduced by Lipmaa et al. [27] in 2005.

1.1 Related Work

Several variants for DVS such as ring signatures [28, 30], universal designated verifier signatures (UDVS) [15, 16, 23, 32, 35, 37], multi-designated verifier signatures (MDVS) [21, 24], and SDVS [3, 17] are proposed. Besides the aforementioned designated verifier signature schemes and its variants in the conventional public key infrastructure (PKI) setting, another useful variant which is combination of DVS and identity-based encryption [31] is identity-based designated verifier signatures (IBDVS)[3, 8, 22, 17–19, 34, 38]. On the other hand, several DVS schemes [32, 8, 22, 25, 26, 16, 33] are shown to be delegatable since the notion of non-delegatability [27] is introduced, while there are a few DVS schemes [18–20, 27] which are non-delegatable. Since 2009, two identity-based non-delegatable (S)DVS [18, 19] are proposed which their performances are not satisfactory enough to be used wildly.

1.2 Contribution

Up to now, several non-delegatable identity-based (strong) designated verifier signature schemes using bilinear pairings are proposed. In these identity-based (strong) designated verifier signature schemes, bilinear pairings are employed either in signing and verifying steps or only in the verifying step. However, the computation cost of pairings at a security level equivalent to a 128-bit symmetric key of AES is approximately 20 times higher than that of exponentiation over an elliptic curve group [12]. Hence, presenting a (strong) designated verifier signature scheme which is identity-based without pairings and supports non-delegatability as well is essential. In this study, we focus on identity-based designated verifier signature schemes. As a result, the first non-delegatable identity-based designated verifier signature scheme without bilinear pairings is proposed. The idea is applying two concatenated Schnorr signatures [29] which has been used elsewhere [1, 4, 9, 13]. Hence, we can employ this idea along with the typical OR proofs of two three-round zero-knowledge protocols as used in [18, 19] to propose the first efficient identity-based non-delegatable designated verifier signature schemes without pairings. In our proposal, private key generation (PKG) generates a Schnorr signature on the user's identity to produce user's secret key using the master secret key. Then, the user produces a designated verifier signature on a message using her secret key. The verification of a signature on the message under the identities of the designated verifier and the signer is performed by checking the two concatenated Schnorr signature.

Our construction not only is approximately 40 times more efficient compared to the existing non-delegatable identity-based (strong) designated verifier signature schemes due to the avoiding bilinear pairings, but also it is provable secure in the random oracle [2] .

1.3 Outline of the paper

The rest of this manuscript is organized as follows. Section 2 presents definition of bilinear pairings and complexity assumptions. The model of IBDVS including outline of the IBDVS scheme and its security properties are described in section 3. The proposed scheme and its formal security proofs are presented in section 4. Section 5 and 6 present comparison and conclusion, respectively.

2 Preliminaries

In this section, we review definition of bilinear pairings and complexity assumptions.

2.1 Definitions and complexity assumptions

Definition 1 (Bilinear pairings). Let G and G_T be two cyclic multiplicative groups of prime order p; furthermore, let g be a generator of G. The map $e : G \times G \longrightarrow G_T$ is a bilinear pairing. We refer readers to [6] for more details on the construction of bilinear pairings.

Assumption 1 (Discrete Logarithm (DL) assumption). The discrete logarithm (DL) assumption (t', ϵ')-holds in Z_p if there is no probabilistic polynomial time (PPT) algorithm A which runs in time at most t' to output a on a given input $(g, g^a \bmod p, p, q)$ with non-negligible probability ϵ', where p and q are two large primes such that $q|p-1$, Z_p is a finite field, and g is its generator with order q.

3 Model of identity-based designated verifier signature schemes

In this section, we review the outline and security properties of identity-based designated verifier signature schemes.

3.1 Outline of identity-based designated verifier signature scheme

There are two participants in an identity-based designated verifier signature (IBDVS) scheme, the signer with identity id_s and the designated verifier with identity id_v. An IBDVS scheme consists of five algorithms as follows [18].

- Setup: Given a security parameter k, this algorithm outputs a master key pair for the private key generator (PKG), i.e. $(mpk, msk) \leftarrow Setup(k)$, where mpk is the master public key of the PKG and msk is the master secret key of the PKG.

- Extract: It (Extract oracle O_E) takes the master secret key msk and an identity id, a string with an arbitrary length, as inputs, then, it outputs the secret key sk_{id} corresponding to id, i.e. $sk_{id} \leftarrow Extract(msk, id)$.

- Sign: This algorithm (Signing oracle O_S) takes the signer's secret key sk_{id_s}, the signer's identity id_s, the designated verifier's identity id_v, the master public key mpk, and a message $M \in \{0,1\}^*$ as its inputs to generate a signature θ, i.e. $\theta \leftarrow Sign(sk_{id_s}, id_s, id_v, mpk, M)$.

- Verify: This algorithm (Verification oracle O_V) takes the designated verifier's identity id_v, the signer's identity id_s, the master public key mpk, the message M, and the signature θ as its inputs and returns a bit b, which is 1 if the signature is valid, otherwise is 0, i.e. $b \leftarrow Verify(id_s, id_v, mpk, \theta, M)$.

– Simulate: This algorithm (Simulation oracle O_{Sim}) takes the designated verifier's secret key sk_{id_v}, the signer's identity id_s, the designated verifier's identity id_v, the master public key mpk, and a message M as its inputs to output an identically distributed transcript θ' which is indistinguishable from the one generated by the signer, i.e. $\theta' \leftarrow Simulate(sk_{id_v}, id_s, id_v, mpk, M)$.

3.2 Security properties of identity-based designated verifier signature schemes

An IBDVS scheme ought to be unforgeable, non-transferable, and non-delegatable. Formal definitions of these properties are expressed as follows [18].

1. Completeness: A properly formed IBDVS must be accepted by the Verify algorithm. Formally, the completeness of the IBDVS requires that for any $(mpk, msk) \leftarrow Setup(k)$, $id_s, id_v \in \{0,1\}^*$, $sk_{id_s} \leftarrow Extract(msk, id_s)$, $sk_{id_v} \leftarrow Extract(msk, id_v)$ and any message $M \in \{0,1\}^*$, we have $pr[Verify(id_s, id_v, mpk, \theta = Sign(sk_{id_s}, id_s, id_v, mpk, M), M) = 1] = 1$ and $pr[Verify(id_s, id_v, mpk, \theta' = Simulate(sk_{id_v}, id_s, id_v, mpk, M), M) = 1] = 1$.

2. Unforgeability: It requires that no one other than the signer with identity id_s and the designated verifier with identity id_v can produce a valid designated verifier signature. The formal definition of unforgeability [21] is expressed in Definition 2. To have a formal definition for unforgeability, the following game between the simulator B and a probabilistic polynomial time (PPT) adversary A is considered to be played.

   (a) B runs the Setup algorithm to generate a master key pair (mpk, msk), and gives mpk to A.
   (b) A issues queries to the following oracles:
      – O_E: This oracle generates the user's secret key $sk_{id} \leftarrow Extract(msk, id)$ on a given id, then returns it to A.
      – O_S: Given a query of the form of (id_s, id_v, M), this oracle first generates the secret key of id_s as $sk_{id_s} \leftarrow Extract(msk, id_s)$, and signs a message M as $\theta \leftarrow Sign(sk_{id_s}, id_s, id_v, mpk, M)$, then returns it to A.
      – O_{Sim}: Given a query of the form of (id_s, id_v, M), this oracle first generates the secret key of id_v as $sk_{id_v} \leftarrow Extract(msk, id_v)$, and signs a message M as $\theta \leftarrow Simulate(sk_{id_v}, id_s, id_v, mpk, M)$, then returns it to A.
   (c) A outputs a forgery $(id_s^*, id_v^*, M^*, \theta^*)$ and wins the game if the three following conditions hold
      – $Verify(id_s^*, id_v^*, M^*, \theta^*) = 1$
      – A did not query O_E on input id_s^* and id_v^*, and
      – A did not query O_S and O_{Sim} on input (id_s^*, id_v^*, M^*).

   The formal definition of unforgeability [21] is expressed in Definition 2.

   Definition 2 (Unforgeability). An IBDVS scheme is $(t, q_E, q_S, q_{Sim}, \epsilon)$-unforgeable if no adversary A which runs in time at most t; issues at most q_E queries to O_E; issues at most q_S queries to O_S; and issues at most q_{Sim} queries to O_{Sim} can win the above game with probability at least ϵ.

3. Non-transferability: This property means that it should be infeasible for any PPT distinguisher to tell whether θ on a message M was generated by the signer with identity id_s or simulated by the designated verifier with identity id_v. The formal definition of non-transferability [21] is expressed in Definition 3.

   Definition 3 (Non-transferability). An IBDVS is non-transferable if there exists a PPT simulation algorithm Sim on sk_{id_v}, id_s, id_v, and a message M outputs a simulated signature which is indistinguishable from the real signatures generated by the signer on the same message. For any PPT distinguisher A, any (id_s, sk_{id_s}), (id_v, sk_{id_v}), and any message $M \in \{0,1\}^*$, Eq. (1) holds.

$$\left| pr \left[\begin{array}{l} \theta_0 \longleftarrow Sign(sk_{id_s}, id_s, id_v, M), \\ \theta_1 \longleftarrow Sim(sk_{id_v}, id_s, id_v, M), \\ b \longleftarrow \{0,1\}, \\ b' \longleftarrow A(id_s, id_v, sk_{id_s}, sk_{id_v}, \theta_b) \\ : b' = b \end{array} \right] - \frac{1}{2} \right| < \epsilon(k) \tag{1}$$

   Where $\epsilon(k)$ is a negligible function in the security parameter k, and the probability is taken over the randomness used in $Sign$ and Sim, and the random coins consumed by A. If the probability is equal to $\frac{1}{2}$, the IBDVS scheme is perfectly non-transferable or source hiding [21].

4. Non-delegatability: It requires that if one generates a valid IBDVS on a message, it must "know" the secret key corresponding to either id_s or id_v. Therefore, a non-delegatable signature is a proof of knowledge of secret key corresponding to either id_s or id_v. The formal definition of non-delegatability [27] is presented in Definition 4.

   Definition 4 (Non-delegatability). It is assumed that $\kappa \in [0,1]$ be the knowledge error. An IBDVS scheme is (t,κ)-non-delegatable if there is a black box knowledge extractor which produces either the secret key of the signer or the secret key of the designated verifier with oracle access to the forger F. If the forger algorithm F generates a valid signature with probability ϵ on a message M for every $mpk \leftarrow Setup(k)$, every $id_s, id_v \in \{0,1\}^*$, every $sk_{id_v} \leftarrow Extract(msk, id_v)$, $sk_{id_s} \leftarrow Extract(msk, id_s)$ and $M \in \{0,1\}^*$, then, the extractor can extract either the secret key of the signer or the secret key of the designated verifier in expected time $t(\epsilon - \kappa)^{-1}$ with the help of the forger F, where $\epsilon > \kappa$ without considering the required time to make oracle queries. Note that, the probability of F is taken over the choice of its random coins and the choices of random oracles.

4 Our non-delegatable identity-based designated verifier signature scheme

To the best of our knowledge, all the IB(S)DVS use bilinear pairings. The computation cost of pairings at a security level equivalent to a 128-bit symmetric key of AES is approximately 20 times higher than that of exponentiation over elliptic curve groups [12]. In this study,

we focus on identity-based designated verifier signature schemes. In this section, we propose an IBDVS scheme without pairings which supports non-delegatability. Before explaining the scheme in details, we first briefly discuss the employed idea to remove pairings. In this study, the idea is applying two concatenated Schnorr signatures [29] which has been used elsewhere [1, 4, 9, 13]. Hence, we can employ this idea along with the typical OR proofs of two three-round zero-knowledge protocols as used in [18, 19] to propose the first efficient identity-based non-delegatable designated verifier signature schemes without pairings. In our proposal, PKG generates a Schnorr signature on the user's identity to produce user's secret key using the master secret key. Then, the user produces a designated verifier signature on a message using her secret key. The verification of a signature on the message under the identities of the designated verifier and the signer is performed by checking the two concatenated Schnorr signature.

4.1 Overview of the identity-based designated verifier signature scheme

There are two participants in the system the signer with identity id_s and the designated verifier with identity id_v. Our scheme consists of five algorithms as follows.

1. Setup: The system parameters are as follows. Let (p, q) be two large primes such that $q|p-1$; further, let Z_p be a finite field and g be its generator with order q. PKG selects $\alpha \in_R Z_q^*$ as its master secret key and sets $g_1 = g^\alpha \bmod p$ as its master public key. The public parameters are $(G, p, q, g, g_1, H_1, H_2)$, where H_1 and $H_2 : \{0,1\}^* \rightarrow Z_q^*$ are two collision-resistant hash functions.

2. Extract: Given an identity id, PKG selects $r \in_R Z_q^*$, and computes $w = g^r$, $h = H_1(id, w)$, and $y = r + \alpha h \bmod q$. The secret key of the user with identity id is $sk_{id} = (w, y)$, while w is also a public parameter.

3. Sign: Let M be a message to be signed by the signer with identity id_s for a designated verifier with identity id_v. First, the signer picks random values b_v and c_v, and $a_s \in_R Z_q^*$, then, the designated verifier signature $\theta = ((c_s, b_s), (c_v, b_v))$ on M is constructed as expressed in Eq.(2).

$$\begin{aligned} R_s &= g^{a_s} \bmod p \\ R_v &= g^{b_v}(w_v g_1^{h_v})^{-c_v} \bmod p \\ c &= H_2(id_s, id_v, R_s, R_v, M) \\ c_s &= c - c_v \\ b_s &= a_s + y_s c_s \bmod q \end{aligned} \tag{2}$$

4. Verify: To check whether θ is a valid designated verifier signature on the message M w.r.t. id_s and id_v, the designated verifier checks whether Eq.(3) holds.

$$\begin{aligned} h_s &= H_1(id_s, w_s) \\ h_v &= H_1(id_v, w_v) \\ R_s &= g^{b_s}(w_s g_1^{h_s})^{-c_s} \bmod p \\ R_v &= g^{b_v}(w_v g_1^{h_v})^{-c_v} \bmod p \\ c_s + c_v &= H_2(id_s, id_v, R_s, R_v, M) \end{aligned} \tag{3}$$

If the equality holds, the designated verifier accepts the signature θ; otherwise, the designated verifier rejects it.

5. Simulate: The designated verifier picks random values b_s, c_s, and $a_v \in_R Z_q^*$, then, the designated verifier signature $\theta = ((c_s, b_s), (c_v, b_v))$ on M is constructed as expressed in Eq. (4).

$$\begin{aligned} &R_v = g^{a_v} \bmod p \\ &R_s = g^{b_s}(w_s g_1^{h_s})^{-c_s} \bmod p \\ &c = H_2(id_s, id_v, R_s, R_v, M) \\ &c_v = c - c_s \\ &b_v = a_v + y_v c_v \bmod q \end{aligned} \tag{4}$$

4.2 Analysis of the scheme

In this section, we will primarily show the completeness of the proposed scheme. Subsequently, we prove that the proposal is secure in the random oracle model.

Completeness. The completeness of the scheme is clear by inspection.

In case of having a real signature, we have

$$\begin{aligned} &h_s = H_1(id_s, w_s) \\ &h_v = H_1(id_v, w_v) \\ &g^{b_s}(w_s g_1^{h_s})^{-c_s} = g^{a_s + y_s c_s}(w_s g_1^{h_s})^{-c_s} \\ &= g^{a_s} g^{y_s c_s}(w_s g_1^{h_s})^{-c_s} \\ &= g^{a_s} g^{(r_s + \alpha h_s)c_s}(w_s g_1^{h_s})^{-c_s} \\ &= g^{a_s}(g^{r_s} g^{\alpha h_s})^{c_s}(w_s g_1^{h_s})^{-c_s} \\ &= R_s(w_s g_1^{h_s})^{c_s}(w_s g_1^{h_s})^{-c_s} \\ &g^{b_v}(w_v g_1^{h_v})^{-c_v} = R_v \\ &c_s + c_v = c - c_v + c_v = c = H_2(id_s, id_v, R_s, R_v, M) \end{aligned} \tag{5}$$

In case of having a simulated signature, we have

$$\begin{aligned} &h_s = H_1(id_s, w_s) \\ &h_v = H_1(id_v, w_v) \\ &g^{b_v}(w_v g_1^{h_v})^{-c_v} = g^{a_v + y_v c_v}(w_v g_1^{h_v})^{-c_v} \\ &= g^{a_v} g^{y_v c_v}(w_v g_1^{h_v})^{-c_v} \\ &= g^{a_v} g^{(r_v + \alpha h_v)c_v}(w_v g_1^{h_v})^{-c_v} \\ &= g^{a_v}(g^{r_v} g^{\alpha h_v})^{c_v}(w_v g_1^{h_v})^{-c_v} \\ &= R_v(w_v g_1^{h_v})^{c_v}(w_v g_1^{h_v})^{-c_v} \\ &g^{b_v}(w_v g_1^{h_v})^{-c_v} = R_v \\ &c_s + c_v = c - c_v + c_v = c = H_2(id_s, id_v, R_s, R_v, M) \end{aligned} \tag{6}$$

As we shall see later (Theorem 2), the scheme is perfectly non-transferable. Hence, making query to Simulation oracle, O_{Sim}, is equivalent to making query to the signing oracle, O_S, in the proof of unforgeability.

Theorem 1. If DL assumption (t', ϵ') holds, then, IBDVS scheme is $(t, q_E, q_S, q_{Sim}, \epsilon)$-existentially unforgeable against adaptively chosen message and identity attack in the random oracle model.

$$\begin{aligned} &\epsilon' \geq \epsilon(\tfrac{\epsilon^3}{(q_{H_1} q_{H_2})^6} - \tfrac{3}{q}) \\ &t' \simeq t + O(q_E + q_S + q_{Sim})t_e \end{aligned} \tag{7}$$

Where t_e is required dominant time for exponentiation in Extract, Sign, and Sim queries. Furthermore, q_{H_1} and q_{H_2} denote the number of queries to the random oracles H_1 and H_2, respectively.

Proof. It is supposed that there is an adversary A against the unforgeability of the scheme with success probability ϵ. We construct another algorithm B to solve DL problem with success probability ϵ'. Given a random instance of DL problem $(g_1 = g^\alpha \bmod p, g, p, q)$, B plans to find α.

Setup. B chooses two collision-resistant hash functions H_1 and H_2 and invokes adversary A on input $(g_1 = g^\alpha \bmod p, g, p, q, H_1, H_2)$. Note that α is the secret key of PKG which is unknown to B. Furthermore, A simulates hash functions H_1 and H_2 as random oracles by keeping two lists l_{H_1} and l_{H_2} including the queried values along with the answers given to A.

- Extract query: Given an identity id, B chooses $(h, y) \in Z_q^* \times Z_q^*$ and sets $w = g_1^{-h} g^y$, next, B adds (w, id, h) to the list l_{H_1}. Finally, B returns (y, w) to A.

- Sign (Sim) query: Given a query (M, id_s, id_v), if there is an entry including id_s on the list l_{H_1}, B retrieves the corresponding secret key (y_s, w_s) to generate (simulate) the designated verifier signature as expressed in Eq.(2)(Eq. (4)). Otherwise, B computes the secret key of id_s as described in Extract query. Then, B computes Eq.(2) to generate the designated verifier signature on M, and adds $((id_s, id_v, M, R_s, R_v), c)$ to the list l_{H_2} and $((y_s, w_s), id_s)$ into the list l_{H_1} and gives the signature to the adversary A.

Finally, A outputs its forgery $(id_s^*, id_v^*, M^*, \theta^*)$, where $\theta^* = ((c_s^*, b_s^*), (c_v^*, b_v^*))$.
Next, B runs the Multiple-Forking algorithm [5] three times to obtain four valid forgeries with the same identities (id_s^*, id_v^*) as presented in Eq.(8).

$$\begin{aligned} c_{s_1}^* + c_{v_1}^* = c_1^* &= H_2(id_s^*, id_v^*, g^{b_{s_1}^*}(w_s^* g_1^{h_{1s}^*})^{-c_{s_1}^*}, g^{b_{v_1}^*}(w_v^* g_1^{h_{1v}^*})^{-c_{v_1}^*}, M) \\ c_{s_2}^* + c_{v_2}^* = c_2^* &= H_2(id_s^*, id_v^*, g^{b_{s_2}^*}(w_s^* g_1^{h_{1s}^*})^{-c_{s_2}^*}, g^{b_{v_2}^*}(w_v^* g_1^{h_{1v}^*})^{-c_{v_2}^*}, M) \\ c_{s_3}^* + c_{v_3}^* = c_3^* &= H_2(id_s^*, id_v^*, g^{b_{s_3}^*}(w_s^* g_1^{h'^*_{1s}})^{-c_{s_3}^*}, g^{b_{v_3}^*}(w_v^* g_1^{h'^*_{1v}})^{-c_{v_3}^*}, M) \\ c_{s_4}^* + c_{v_4}^* = c_4^* &= H_2(id_s^*, id_v^*, g^{b_{s_4}^*}(w_s^* g_1^{h'^*_{1s}})^{-c_{s_4}^*}, g^{b_{v_4}^*}(w_v^* g_1^{h'^*_{1v}})^{-c_{v_4}^*}, M) \end{aligned} \tag{8}$$

If $h^*_{1s} \neq h'^*_{1s}$, $c^*_{s_1} \neq c^*_{s_2}$ and $c^*_{s_3} \neq c^*_{s_4}$, the value of α from Eq.(9) can be computed contradicting assumption 1.

$$\begin{array}{l} b^*_{s_1} = a^*_s + y^*_s c^*_{s_1} = a^*_s + (r^*_s + \alpha h^*_{1s}) c^*_{s_1} \\ b^*_{s_2} = a^*_s + y^*_s c^*_{s_2} = a^*_s + (r^*_s + \alpha h^*_{1s}) c^*_{s_2} \\ b^*_{s_3} = a^*_s + y'^*_s c^*_{s_3} = a^*_s + (r^*_s + \alpha h'^*_{1s}) c^*_{s_3} \\ b^*_{s_4} = a^*_s + y'^*_s c^*_{s_4} = a^*_s + (r^*_s + \alpha h'^*_{1s}) c^*_{s_4} \end{array} \tag{9}$$

In this case, $\alpha = \frac{(b^*_{s_3} + b^*_{s_2} - b^*_{s_1} - b^*_{s_4})}{(h'^*_{1s}(c^*_{s_3} - c^*_{s_4}) - h^*_{1s}(c^*_{s_1} - c^*_{s_2}))}$.

If $h^*_{1v} \neq h'^*_{1v}$, $c^*_{v_1} \neq c^*_{v_2}$ and $c^*_{v_3} \neq c^*_{v_4}$, the value of α from Eq.(10) can be computed contradicting assumption 1.

$$\begin{array}{l} b^*_{v_1} = a^*_v + y^*_v c^*_{v_1} = a^*_v + (r^*_v + \alpha h^*_{1v}) c^*_{v_1} \\ b^*_{v_2} = a^*_v + y^*_v c^*_{v_2} = a^*_v + (r^*_v + \alpha h^*_{1v}) c^*_{v_2} \\ b^*_{v_3} = a^*_v + y'^*_v c^*_{v_3} = a^*_v + (r^*_v + \alpha h'^*_{1v}) c^*_{v_3} \\ b^*_{v_4} = a^*_v + y'^*_v c^*_{v_4} = a^*_v + (r^*_v + \alpha h'^*_{1v}) c^*_{v_4} \end{array} \tag{10}$$

In this case, $\alpha = \frac{(b^*_{v_3} + b^*_{v_2} - b^*_{v_1} - b^*_{v_4})}{(h'^*_{1v}(c^*_{v_3} - c^*_{v_4}) - h^*_{1v}(c^*_{v_1} - c^*_{v_2}))}$.

As a result, success probability of B according to the Multiple-Forking Lemma of Boldyreva et al. [5] is bounded by $\epsilon' \geq frk \geq \epsilon(\frac{\epsilon^3}{(q_{H_1} q_{H_2})^6} - \frac{3}{q})$.
Note that, we consider t' as the required time for the simulation and generating the first forgery by A.

Theorem 2. The proposal is perfectly non-transferable.
Proof. To prove non-transferability of the scheme, we show that the signature simulated by the designated verifier is indistinguishable from that generated by the signer. As a result, we have to show that the two following distributions are identical.

$$\theta = \begin{cases} a_s \in_R Z^*_q \\ b_v \in_R Z^*_q \\ c_v \in_R Z^*_q \\ R_s = g^{a_s} \bmod p \\ R_v = g^{b_v}(w_v g_1^{h_v})^{-c_v} \bmod p \\ c = H_2(id_s, id_v, R_s, R_v, M) \bmod q \\ c_s = c - c_v \bmod q \\ b_s = a_s + y_s c_s \bmod q \end{cases} \tag{11}$$

$$\theta' = \begin{cases} a_v \in_R Z^*_q \\ b_s \in_R Z^*_q \\ c_s \in_R Z^*_q \\ R_v = g^{a_v} \bmod p \\ R_s = g^{b_s}(w_s g_1^{h_s})^{-c_s} \bmod p \\ c = H_2(id_s, id_v, R_s, R_v, M) \bmod q \\ c_v = c - c_s \bmod q \\ b_v = a_v + y_v c_v \bmod q \end{cases} \tag{12}$$

Let $\overline{\theta}$ be a valid signature which is randomly chosen from the set of all valid signer's signatures intended to the verifier. Subsequently, we have distributions of probabilities as follows:

$$Pr_{\theta} = Pr[\theta = \overline{\theta}] = \frac{1}{(q-1)^3}, \tag{13}$$

$$Pr_{\theta'} = Pr[\theta' = \overline{\theta}] = \frac{1}{(q-1)^3} \tag{14}$$

The analysis means both distributions of probability are the same. Hence, our proposal satisfies perfect non-transferability.

Theorem 3. If there is an algorithm F which can generate a valid signature in time t with probability ϵ for some identities id_s and $id_v \in \{0,1\}^*$ and some message $M \in \{0,1\}^*$, then, the proposal is $(\frac{56t}{\epsilon}, \frac{1}{q})$-non-delegatable in the random oracle model.

Proof. It is assumed that $\epsilon > \frac{1}{q}$, where $\frac{1}{q}$ is the probability that F guesses correctly the value of $H_2(id_s, id_v, R_s, R_v, M)$ without asking the random oracle H_2. Therefore, there is an extractor which can extract the secret key either the signer or the designated verifier on input θ and black-box oracle access to F.

It is supposed that F be a forger on input (id_s, id_v, M). Next, the knowledge extractor runs F two times on the same random input (R_s, R_v), while the knowledge extractor returns different random values (c versus c') as the answer to the hash query $H_2(id_s, id_v, R_s, R_v, M)$. It is supposed that both signatures $\theta = ((c_s, b_s), (c_v, b_v))$ and $\theta' = ((c'_s, b'_s), (c'_v, b'_v))$ are valid, then, one can call the extractor of the proof of knowledge to extract the secret key of either the signer, y_s, or the designated verifier, y_v. Hence, we have $c = c_s + c_v = H_2(id_s, id_v, R_s, R_v, M)$ and $c' = c'_s + c'_v = H_2(id_s, id_v, R'_s, R'_v, M)$ which implies either $b_s = a_s + y_s c_s$ and $b'_s = a_s + y_s c'_s$ or $b_v = a_v + y_v c_v$ and $b'_v = a_v + y_v c'_v$. $c \neq c'$ implies that $c_s \neq c'_s$ or $c_v \neq c'_v$. If $c_s \neq c'_s$, extractor can obtain $y_s = \frac{b_s - b'_s}{c_s - c'_s}$ from b_s and b'_s with probability 1. Similarly, if $c_v \neq c'_v$, extractor can obtain $y_v = \frac{b_v - b'_v}{c_v - c'_v}$ from b_v and b'_v with probability 1. The required time to compute two valid forgery using the rewind algorithm from [11] is $\frac{56}{\epsilon}$.

5 Comparison

This section compares the efficiency of our proposal with previous provably secure IB(S)DVS schemes in terms of $Sign - Cost$ and $Ver - Cost$, dominating computational cost in signature generation and verification, respectively; ND, if the scheme is non-delegatable based on definition of [27]; RO if the security of the scheme is in the random oracle model. We assume that our scheme should be built upon a group of points G of prime order p of a suitable elliptic curve. Hence, in mapping, an element in Z_p in our scheme is equivalent to an element in G, and an element in Z_q in our scheme corresponds to an element in Z_p. In table 1 and 2, P, E, E_T, and exp_G denote the pairing evaluation, exponentiation in group G, exponentiation in

group G_T, and the cost of computing an exponentiation in G, respectively.

Schemes	Type	Sign-Cost	Ver-Cost	Signature-Size	ND	RO
Ours	IBDVS	$4E$	$6E$	$4Z_p$	✓	✓
Huang et al. 2011 [19]	IBSDVS	$3P + 2E + 4E_T$	$5P + 1E + 4E_T$	$2G + 2G_T + 3Z_p$	✓	✓
Huang et al. 2009 [18]	IBDVS	$3P + 1E + 3E_T$	$4P + 4E_T$	$1G + 4Z_p$	✓	✓
Cao et al. 2009 [8]	IBDVS	$6E$	$5P$	$4G$	×	×
Huang et al. 2008 [17]	IBSDVS	$1P$	$1P$	$1Z_p$	×	✓
Kang et al. 2009 [22]	IBSDVS	$2P + 2E + 1E_T$	$1P + 1E_T$	$2G_T$	×	✓
Zhang et al. 2008 [38]	IBSDVS	$4E$	$3P$	$3G$	×	✓

Table 1. Comparison between our proposal and other existing schemes

As shown in Table 1, our scheme only needs 4 exponentiations over an elliptic curve group which $g_1^{h_v}$ can be computed in advance; hence, 3 exponentiations are needed in the signing Step which 3 exponentiations according to [7] has a cost of about 1.5 times that of a single exponentiation in G. For verification, 6 exponentiations need to be done which two of them $g_1^{h_s}$ and $g_1^{h_v}$ can be performed in advance; hence, verification includes 4 exponentiations in G which it requires a cost of slightly more than 1.5 times that of a single exponentiation in G according to [7]. Exponentiating in G_T needs a higher computational cost than computing an exponentiating in G. The exact cost depending on how the arithmetic on those groups is implemented. Hence, we leave them un-quantified. Furthermore, computing a paring can be as expensive as 21 exponentiations in G at a security level equivalent to a 128-bit symmetric key of AES [12].

We form Table 2 as an equivalent Table with substituting $1P$ with $21E$ which shows that our proposal in comparison with previous IB(S)DVS schemes are more efficient due to avoiding computing bilinear pairings and exponentiating in G_T. Note that, this inefficiency is considerable when (strong) designated verifier signature scheme is non-delegatable.

Schemes	Type	Sign-Cost	Ver-Cost	Signature-Size	ND	RO
Ours	IBDVS	$1.5E$	$1.5E \leq exp_G \leq 2E$	$4Z_p$	✓	✓
Huang et al. 2011 [19]	IBSDVS	$63E + 4E_T$	$106E + 4E_T$	$2G + 2G_T + 3Z_p$	✓	✓
Huang et al. 2009 [18]	IBDVS	$64E + 3E_T$	$84E + 4E_T$	$1G + 4Z_p$	✓	✓
Cao et al. 2009 [8]	IBDVS	$6E$	$105E$	$4G$	×	×
Huang et al. 2008 [17]	IBSDVS	$21E$	$21E$	$1Z_p$	×	✓
Kang et al. 2009 [22]	IBSDVS	$44E + 1E_T$	$21E + 1E_T$	$2G_T$	×	✓
Zhang et al. 2008 [38]	IBSDVS	$4E$	$63E$	$3G$	×	✓

Table 2. Equivalent comparison table between our proposal and other existing schemes

Note that, signature size of our proposal is $|G|$ smaller than the size of the non-delegatable IBDVS [18] and also it is approximately $(|Z_p| - 2|G| - 2|G_T|)$ larger than size of non-delegatable

IBSDVS [19], where |.| denotes the number of bits representing an element.

6 Conclusion and future work

We propose the first non-delegatable identity-based designated verifier signature scheme without bilinear pairings. Security of our construction is proved in the random oracle; our construction not only is identity based without pairings (it is in turn more efficient compared to the existed identity-based designated verifier signature schemes due to the avoiding bilinear pairings), but also it is non-delegatable. The performance enjoyed by our proposal make it specially suited for deployment in resource-constrained devices where savings in computation and are a premium, e.g. wireless sensor networks.
Proposing a provable secure IBSDVS without pairings which supports non-delegatability will be left as a future work.

References

1. Baek, J., Safavi-Naini, R., Susilo, W., *Certificateless public key encryption without pairing,* Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) Information Security Conference (ISC) 2005, vol. 3650 of Lecture Notes in Computer Science, pp. 134-148, Springer-Verlag (2005).
2. Bellare, M., Rogaway, P., *Random oracles are practical: a paradigm for designing efficient protocols,* ACM Conference on Computer and Communications Security, pp. 62-73, ACM (1993).
3. Bhaskar, R., Herranz, J., Laguillaumie, F. *Aggregate designated verifier signatures and application to secure routing,* International Journal of Security Network, vol. 2(3/4), pp.192-201, (2007).
4. Boldyreva, A., Fischlin, M., Palacio, A., Warinschi, B., *A closer look at PKI: Security and efficiency,* Okamoto, T., Wang, X. (eds.) Proceedings of Public Key Cryptography 2007, vol. 4450 of Lecture Notes in Computer Science, pp. 458-475, Springer-Verlag (2007).
5. Boldyreva, A., Palacio, A., Warinschi, B., *Secure proxy signature schemes for delegation of signing rights,* Cryptology ePrint Archive, Report 2003/096, 2003.
6. Boneh, D., Franklin, M., *Identity-based encryption from the Weil pairings,* Advances in Cryptology - Crypto 2001, vol. 3494 of Lecture Notes in Computer Science, pp. 213-229, Springer-Verlag (2001).
7. Brumley, B. B., *Efficient three-term simultaneous elliptic scalar multiplication with applications,* Fak, V. (ed.), Proceedings of 11th Noradic workshop on secure IT systems, NordSec 2006, pp. 105-116, 2006.
8. Cao, F., Cao, Z., *An identity based universal designated verifier signature scheme secure in the standard model,* International Journal of Systems and Software, vol. 82(4), pp. 643-649, 2009.
9. Castelluccia, C., Jarecki, S., Tsudik, G., *Secret handshakes from CA-oblivious encryption,* Lee, P.J. (ed.) ASIACRYPT 2004, vol. 3329 of Lecture Notes in Computer Science vol. 82(4), pp. 293-307, Springer-Verlag (2004).
10. Chaum, D., van Antwerpen, H., *Undeniable signatures,* Proceedings of Advances in Cryptology-CRYPTO 1989, vol. 435 of Lecture Notes in Computer Science, pp. 212-216. Springer (1989).
11. Damgard, I. and Fujisaki, E., *An integer commitment scheme based on groups with hidden order,* ASIACRYPT 2002, vol. 2501 of Lecture Notes in Computer Science, pp. 125-142. Springer (2002).
12. ECRYPT, *Ecrypt yearly report on algorithms and key length,* http://www.ecrypt.eu.org/documents/D.SPA.21-1.1.pdf revision 1.1 (2006).

13. Galindo, D., Garcia, F. D., *A Schnorr-like lightweight identity-based signature scheme,*Proceedings of 2nd African International Conference on Cryptology, AfricaCrypt 2009, Lecture Notes in Computer Science, vol. 5580, pp. 135-148, 2009.
14. Huang, X., Mu, Y., Susilo, W., Wu, W., *Provably secure pairing based convertible undeniable signature with short signature length,* Proceedings of 1st International Conference on Pairing-Based Cryptography, Pairing 2007, vol. 4575 of Lecture Notes in Computer Science, pp. 367-391, Springer (2007).
15. Huang, X., Susilo, W., Mu, Y., Wu, W., *Universal designated verifier signature without delegatability,* Proceedings of 8th International Conference on Information and Communications Security, ICICS 2006, vol. 4307 of Lecture Notes in Computer Science, pp. 479-498, Springer (2006).
16. Huang, X., Susilo, W., Mu, Y., Wu, W., *Secure universal designated verifier signature without random oracles,* International Journal of Information Security, vol. 7(3), pp. 171-183, (2007).
17. Huang, X., Susilo, W., Mu, Y., Zhang, F., *Short designated verifier signature scheme and its identity-based variant,* International Journal of Network Security, vol. 6(1), pp.82-93, (2008).
18. Huang, Q., Susilo, W., Wong, D. S., *Non-delegatable identity- based designated verifier signature,* Cryptology ePrint Archive, Report 2009/367, 2009.
19. Huang, Q., Yang, G., Wong, D. S., Susilo, W., *Identity-based strong designated verifier signature revisited,* International Journal of Systems and Software, vol.84(1), pp.120-129, 2011.
20. Huang, Q., Yang, G., Wong, D. S., Susilo, W., *Efficient strong designated verifier signature schemes without Random Oracle or with non-delegatability,* International Journal of Information Security, Springer, pp.373-385, 2011.
21. Jakobsson, M., Sako, K., Impagliazzo, R., *Designated verifier proofs and their applications,* Proceedings of Advances in Cryptology-EUROCRYPT 1996, vol. 1070 of Lecture Notes in Computer Science, pp. 143-154, Springer (1996).
22. Kang, B., Boyd, C., Dawson, E., *A novel identity based strong designated verifier signature scheme,* International Journal of Systems and Software, vol. 82(2), pp. 270-273,2009.
23. Laguillaumie, F., Libert, B., Quisquater, J.-J., *Universal designated verifier signatures without random oracles or non-black box assumptions,* Proceedings of 5th International Conference on Security and Cryptography for Networks, SCN 2006, vol. 4116 of Lecture Notes in Computer Science, pp. 63-77, Springer (2006).
24. Laguillaumie, F., Vergnaud, D., *Multi-designated verifiers signatures,* Proceedings of 6th International Conference on Information and Communications Security, ICICS 2004, vol. 3269 of LectureNotes in Computer Science, pp. 495-507, Springer (2004b).
25. Laguillaumie,F., Vergnaud,D., *Designated verifier signature: anonymity and efficient construction from any bilinear map*, Proceedings of 3th International Conference on Security and Cryptography for Networks, SCN 2004, Lecture Notes in Computer Science, pp. 105-119, Springer (2004).
26. Li, Y., Lipmaa, H., Pei, D., *On delegatability of four designated verifier signatures,* Proceedings of 7th International Conference on Information and Communications Security, ICICS 2005, vol.e 3783 of Lecture Notes in Computer Science, pp. 61-71, Springer (2005).
27. Lipmaa, H., Wang, G., Bao, F., *Designated verifier signature schemes: Attacks, new security notions and a new construction,* Proceedings of 32th International Colloquium on Automata, Languages and Programming, ICALP 2005, vol. 3580 of Lecture Notes in Computer Science, pp. 459-471, Springer (2005).
28. Rivest, R., Shamir, A., Tauman, Y., *How to leak a secret,* Boyd C. (ed.) Proceedings of Advances in Cryptology-ASIACRYPT 2001, vol. 2248 of Lecture Notes in Computer Science, pp. 552-565, Springer (2001).
29. Schnorr, C. P., *Efficient signature generation by smart cards,* International Journal of Cryptology 1991, vol. 4(3), pp. 161-174, 1991.

30. Shacham, H.,Waters, B., *Efficient ring signatures without random Oracles,* Okamoto, T., Wang, X. (eds.) Proceedings of Public Key Cryptography 2007, vol. 4450 of Lecture Notes in Computer Science, pp. 166-180, Springer (2007).
31. Shamir, A., *Identity-based cryptosystems and signature schemes,* Advances in Cryptology - Crypto 1984, Lecture Notes in Computer Science, pp. 47-53, Springer-Verlag(1984).
32. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J., *Universal designated verifier signatures,* Proceedings of Advances in Cryptology-ASIACRYPT 2003, vol. 2894 of Lecture Notes in Computer Science, pp. 523-542, Springer (2003).
33. Steinfeld, R.,Wang, H., Pieprzyk, J. *Efficient extension of standard Schnorr/RSA signatures into universal designated verifier signatures,* Proceedings of Public Key Cryptography 2004, vol. 2947 of LectureNotes in Computer Science, pp. 86-100. Springer-Verlag (2004).
34. Susilo, W., Zhang, F., Mu, Y., *Identity-based strong designated verifier signature schemes,* Proceedings of 9th Australasian Conference on Information Security and Privacy, ACISP 2004, vol. 3108 of LectureNotes in Computer Science, pp. 313-324, Springer-Verlag (2004).
35. Vergnaud, D., *New extensions of pairing-based signatures into universal designated verifier signatures,* Proceedings of 33th International Colloquium on Automata, Languages and Programming, ICALP 2006, vol. 4052 of Lecture Notes in Computer Science, pp. 58-69, Springer-Verlag (2006).
36. Wang, B., Song, Z., *A non-interactive deniable authentication scheme based on designated verifier proofs,* International Journal of Information Sciences, vol. 179(6), pp. 858- 865, 2009.
37. Zhang, R., Furukawa, J., Imai, H., *Short signature and universal designated verifier signature without random oracles,* Proceedings of 3rd International Conference on Applied Cryptography and Network Security, ACNS 2005, vol. 3531 of LectureNotes in Computer Science, pp. 483-498, Springer-Verlag (2005).
38. Zhang, J. and Mao, J., *A novel id-based designated verifier signature scheme,* International Journal of Information Sciences, vol.178(3), pp.766-773, 2008.