# A Pairing Based Strong Designated Verifier Signature Scheme without Random Oracles

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

**Abstract.** In this study, a novel strong designated verifier signature scheme based on bilinear pairings with provable security in the standard model is proposed, while the existing ones are secure in the random oracle model. In 2007 and 2011, two strong designated verifier signature schemes in the standard model are proposed by Huang et al. and Zhang et al., respectively; in the former, the property of privacy of the signer's identity is not proved and the security of the latter is based on the security of a pseudorandom function. Our proposal can deal with the aforementioned drawbacks of the previous schemes. Furthermore, it satisfies non-delegatability for signature verification.

**Keywords** : strong designated verifier signature, standard model, bilinear pairing, random oracle model.

## 1 Introduction

Jakobsson et al. [11] 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 [11], undeniable signature [4, 5, 7], deniable authentication [25] 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)[11]. The strongness of a SDVS as privacy of a signer's identity (PSI) is formalized in [14] by Laguillamie and Vergnand in 2004. 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 Limpaa et al. [16] in 2005. In 2011, Huang et al. [10] informally define non-delegatability for signature verification; it requires that if one verifies a valid designated verifier signature on a message, she must "know" the secret key of the designated verifier.

### 1.1 Related Work

Several variants for DVS such as ring signatures [18, 20], universal designated verifier signatures (UDVS) [6, 7, 12, 21, 24, 27], multi-designated verifier signatures [11, 13], and identity-based designated verifier signatures (IBDVS)[3, 8, 9, 23], and (SDVS)[3, 8] are proposed. Several DVS schemes [14, 15, 19, 21, 22] are shown to be delegatable since introducing the notion of non-delegatability [16], while there are a few non-delegatable DVS schemes [9, 16, 28] in the random oracle model [1]. Recently, two SDVS schemes in the standard model are proposed in [10] and [28], respectively.

### 1.2 Contribution

In this paper, a provable secure SDVS scheme based on bilinear pairings without random oracles is proposed. This scheme is based on Water's scheme proposed in [26]. The security of the proposal, i.e. unforgeability and privacy of the signer's identity are based on the standard complexity assumptions. On the top of non-transferability, this scheme is non-delegatable for signature verification which means Bob's secret key is required to verify a designated verifier signature, while it is delegatable for signing. Non-delegatability for signature verification of our proposal is based on BDH assumption which can be converted to the DL assumption in some conditions [17] which is equivalent with the definition of non-delegatability for signature verification. Compared to the SDVS scheme proposed in [10], our proposal does not use a pseudorandom function (fairly strong assumption); furthermore, in comparison to the SDVS scheme in [28], it has a security proof for the PSI property.

### 1.3 Outline of the paper

The rest of this manuscript is organized as follows. Section 2 presents a number of preliminaries, bilinear pairings and complexity assumptions, as the signature foundation. The model of SDVS including outline of the SDVS scheme and its security properties are described in section 3. The proposed scheme and its formal security proofs are presented in section 4. Sections 5 and 6 present the comparison for our scheme to other schemes; and the conclusion, respectively.

## 2 Preliminaries

In this section, we review several fundamental backgrounds employed in this research, including bilinear pairings and complexity assumptions.

### 2.1 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 said to be an admissible bilinear pairing if the following conditions hold true.

1. $e$ is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all $a$ and $b \in Z_p$
2. $e$ is non-degenerate, i.e. $e(g, g) \neq 1_{G_T}$
3. $e$ is efficiently computable.

We refer readers to [2] for more details on the construction of bilinear pairings.

### 2.2 Complexity assumptions

**Definition 1** (Bilinear Diffie-Hellman (BDH) problem in $(G, G_T)$). Given $(g, g^a, g^b, g^c \in G)$ for some unknown $a$, $b$, and $c \in Z_p$ compute $e(g, g)^{abc} \in G_T$.

**Definition 2** (Decisional Bilinear Diffie-Hellman (DBDH) problem in $(G, G_T)$). Given $(g, g^a, g^b, g^c \in G)$ for some unknown $a$, $b$, and $c \in Z_p$ and $Z \in G_T$, decide whether $Z = e(g, g)^{abc}$.

A DBDH oracle $O_{DBDH}$ which takes $(g, g^a, g^b, g^c \in G)$ and $Z \in G_T$ as inputs, outputs 1 if $Z = e(g, g)^{abc}$ and 0 otherwise.

**Definition 3** (Gap Bilinear Diffie-Hellman (GBDH) problem in $(G, G_T)$). Given $(g, g^a, g^b, g^c \in G)$ for some unknown $a$, $b$, and $c \in Z_p$ compute $e(g, g)^{abc} \in G_T$ with the help of the DBDH oracle $O_{DBDH}$.

The probability that a polynomial bounded algorithm $A$ can solve the GBDH problem is defined as $Succ_A^{GBDH} = pr[e(g, g)^{abc} \longleftarrow A(G, G_T, g, g^a, g^b, g^c, O_{DBDH})]$.

**Definition 4** (Gap Bilinear Diffie-Hellman (GBDH) assumption in $(G, G_T)$). Given $(g, g^a, g^b, g^c \in G)$ for some unknown $a$, $b$, and $c \in Z_p$, $Succ_A^{GBDH}$ is negligible.

## 3 Model of strong designated verifier signature schemes

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

### 3.1 Outline of designated verifier signature schemes

There are two participants in a designated verifier signature scheme, the signer $S$ and the designated verifier $V$. A designated verifier signature scheme consists of five algorithms as follows.

- Setup: Given a security parameter $k$, this algorithm outputs the system parameters.
- Key generation: It takes the security parameter $k$ as its input and outputs the secret-public key $(sk_i, pk_i)$ for $i \in \{S, V\}$.
- Sign: This algorithm takes the signer's secret key $sk_S$, the designated verifier's public key $pk_V$, and a message $M$ as its inputs to generate a signature $\sigma$.
- Verify: This algorithm takes the designated verifier's secret key $sk_V$, the signer's public key $pk_S$, the message $M$, and the signature $\sigma$ as its inputs and returns $\top$ if the signature is valid, otherwise returns $\perp$ indicating the signature is invalid.
- Transcript simulation: This algorithm takes the designated verifier's secret key $sk_V$, the signer's public key $pk_S$, and a message $M$ as its inputs to output an identically distributed transcript $\sigma'$ which is indistinguishable from the one generated by the signer.

### 3.2 Security properties of designated verifier signature schemes

A SDVS scheme ought to be unforgeable, non-transferable, and satisfy the privacy of the signer's identity. An SDVS is said to be non-delegatable if it satisfies non-delegatability. Formal definitions of these properties are expressed as follows.

1. Correctness: A properly formed SDVS must be accepted by the verifying algorithm. Formally, the correctness of the SDVS requires that for any $(pk_S, sk_S)$, $(pk_V, sk_V)$ and any message $M \in \{0,1\}^*$, we have $pr[ver(sk_V, pk_S, pk_V, M, \sigma = sign(sk_S, pk_S, pk_V, M)) = 1] = 1$.

2. Unforgeability: It requires that no one other than the signer $S$ and the designated verifier $V$ can produce a valid designated verifier signature. 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) $B$ prepares the key pairs $(pk_S, sk_S)$ for $S$ and $(pk_V, sk_V)$ for $V$, and gives $(pk_S, pk_V)$ to $A$.
   (b) $A$ issues queries to the following oracles.
      - $O_s$: This oracle generates a signature $\sigma$ on a given message $M$ using $sk_S$ such that this signature is valid w.r.t. $pk_S$ and $pk_V$, then returns it to $A$.
      - $O_{sim}$: This oracle generates a simulated signature $\sigma'$ on a given message $M$ using $sk_V$ such that this simulated signature is valid w.r.t. $pk_S$ and $pk_V$, then returns it to $A$.
      - $O_v$: This oracle takes a query of the form $(M, \sigma)$ as an input and returns a bit $b$ which is 1 if $\sigma$ is a valid signature on $M$ w.r.t. $pk_S$ and $pk_V$; otherwise, returns 0.
   (c) $A$ outputs a forgery $(M^*, \sigma^*)$ and wins the game if the two following conditions hold
      - $Ver(sk_V, pk_S, pk_V, M^*, \sigma^*) = 1$
      - It did not query $O_s$ and $O_{sim}$ on input $M^*$.

   The formal definition of unforgeability [11] is expressed in Definition 5.

   **Definition 5** (Unforgeability). An SDVS scheme is $(t, q_s, q_{sim}, q_v, \epsilon)$-unforgeable if no adversary $A$ which runs in time at most $t$; issues at most $q_s$ queries to $O_s$; issues at most $q_{sim}$ queries to $O_{sim}$; and issues at most $q_v$ queries to $O_v$ can win the above game with probability at least $\epsilon$.

3. Non-transferability: This property means that it should be infeasible for any PPT distinguisher to tell whether $\sigma$ on a message $M$ was generated by the signer $S$ or simulated by the designated verifier $V$. Formally, the definition 6 is considered [11].

   **Definition 6** (Non-transferability). An SDVS is non-transferable if there exists a PPT simulation algorithm $Sim$ on $sk_V$, $pk_S$, $pk_V$, 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 $(pk_S, sk_S)$, $(pk_V, sk_V)$, and any message $M \in \{0,1\}^*$, Eq. (1) holds.

$$\left| pr \begin{bmatrix} \sigma_0 \longleftarrow Sign(sk_S, pk_S, pk_V, m), \\ \sigma_1 \longleftarrow Sim(sk_V, pk_S, pk_V, m), \\ b \longleftarrow \{0,1\}, \\ b' \longleftarrow A(pk_S, sk_S, pk_V, sk_V, \sigma_b) \\ : b' = b \end{bmatrix} - \frac{1}{2} \right| < \epsilon(k) \tag{1}$$

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 SDVS scheme is perfectly non-transferable or source hiding.

4. Privacy of the Signer's Identity (PSI): A SDVS has the property of PSI if no one can tell signatures generated by the signer $S_0$ for a $V$ is different from signatures generated by the signer $S_1$ for the $V$ in case of not knowing the secret key of the $V$. To have a formal definition for PSI, the following game between the simulator $B$ and the distinguisher $A$ is considered.

   (a) $B$ generates key pairs $(pk_{S_0}, sk_{S_0})$ for signer $S_0$, $(pk_{S_1}, sk_{S_1})$ for signer $S_1$, and $(pk_V, sk_V)$ for designated verifier $V$, and invokes $A$ on input $pk_{S_0}$, $pk_{S_1}$, and $pk_V$.

   (b) $B$ issues queries $(M, d)$ to the $O_s$ and $O_v$ which $d \in \{0,1\}$ indicating which signer responds to that query.

   (c) $B$ tosses a coin $d \in \{0,1\}$ for the message $M^*$ submitted by $A$, then computes the challenge signature $\sigma^* \longleftarrow Sign(sk_{S_d}, pk_{S_d}, pk_V, M^*)$ and returns $\sigma^*$ to $A$

   (d) $A$ outputs a bit $d'$ and wins the game if the two following conditions hold.

      - $d' = d$
      - It did not query $O_v$ on input $(d, M^*, \sigma^*)$ for any $d \in \{0,1\}$

   The formal definition of this property [14] is given in Definition 7.

**Definition 7** (Privacy of the Signer's Identity). An SDVS scheme is $(t, q_s, q_v, \epsilon)$-PSI-secure if no adversary $A$ which runs in time at most $t$; issues at most $q_s$; and $q_v$ queries to $O_v$ can win the aforementioned game with probability that deviated from $\frac{1}{2}$ by more than $\epsilon$.

5. Non-delegatability for signing: It requires that if one generates a valid designated verifier signature on a message, it must "know" the secret key of either $S$ or $V$. So, a signature is a proof of knowledge of secret key of either $S$ or $V$. The formal definition of non-delegatability is presented in [16].

6. Non-delegatability for signature verification: It requires that if one verifies a valid designated verifier signature on a message, she must "know" the secret key of $V$ as aforementioned in [10].

   We consider Definition 8 to have a formal definition of non-delegatability for signature verification.

**Definition 8** (Non-delegatability for signature verification). It is assumed that $A$ is a verifier algorithm for a SDVS scheme. The SDVS scheme is non-delegatable for signature verification if there is a black-box knowledge extractor $B$ for every algorithm $A$ and every valid signature $\sigma$ on the message $M$ satisfies the following conditions: if $A$ outputs 1, the signature is valid, with probability $\epsilon$ in time $t$, then $B$ produces $sk_V$ with probability $\epsilon' = f(k, \epsilon)$ in expected polynomial time for every $pk_S$, $pk_V$, and the message $M$, where $f()$ is a polynomial.

## 4 Our designated verifier signature scheme

In this section, we describe our designated verifier signature scheme. There are two participants in the system the signer $S$ and the designated verifier $V$. In the following, all the messages to be signed will be represented as bit strings of length $n$. To construct a more flexible scheme which allows messages of arbitrary length, a collision resistant Hash function $H$ should be employed. Our scheme consists of five algorithms as follows.

1. Setup: The system parameters are as follows. Let $(G, G_T)$ be bilinear groups where $|G| = |G_T| = p$ for some prime $p$; further, let $g$ be the generator of $G$. $e$ denotes an admissible pairing $e : G \times G \longrightarrow G_T$. Pick $m' \in G$, and a vector $\boldsymbol{m} = (m_i)$ of length $n$, whose entries are random elements from $G$. The public parameters are $(G, G_T, e, m', \boldsymbol{m})$.

2. Key generation: The signer $S$ picks randomly $x_S$ and $y_S \in Z_p^*$ and sets her secret key $sk_S = (x_S, y_S)$. Then, the signer $S$ computes her public key $pk_S = (pk_{1S}, pk_{2S}) = (g^{x_S}, g^{y_S})$. Similarly, the designated verifier's secret key is $sk_V = x_V \in Z_p^*$ and his public key is $pk_V = g^{x_V}$.

3. Signing. Let $M$ be an $n$-bit message to be signed by the signer $S$ and $M_i$ denotes the $i$-bit of $M$, and $\widetilde{M} \subseteq \{1, 2, ..., n\}$ be the set all $i$ for which $M_i = 1$, the designated verifier signature is generated as follows. First, the signer $S$ picks a random value $r \in_R Z_p^*$ and computes $\sigma_1 = g^r$. The designated verifier signature $\sigma = (\sigma_1, \sigma_2)$ on $M$ is constructed as expressed in Eq.(2).
$$\sigma_2 = e(g^{x_S y_S}(m' \textstyle\prod_{i \in \widetilde{M}} m_i)^r, g^{x_V}) \tag{2}$$

4. Verifying. To check whether $\sigma = (\sigma_1, \sigma_2)$ is a valid designated verifier signature on the message $M$, the designated verifier $V$ uses his secret key to verify whether the Eq. (3) holds.
$$\sigma_2 = e(g^{x_S}, g^{y_S})^{x_V} e(m' \textstyle\prod_{i \in \widetilde{M}} m_i, \sigma_1)^{x_V} \tag{3}$$
If the equality holds, the designated verifier $V$ accepts the signature $\sigma = (\sigma_1, \sigma_2)$; otherwise, the designated verifier $V$ rejects it.

5. Simulation of a transcript. The designated verifier $V$ can use his secret key to compute a signature on an arbitrary message $M'$. He picks a random value $r' \in_R Z_p$ and computes $\sigma'_1 = g^{r'}$ and computes the Eq. (4).
$$\sigma'_2 = e(g^{x_S}, g^{y_S})^{x_V} e(m' \textstyle\prod_{i \in \widetilde{M}} m_i, \sigma'_1)^{x_V} \tag{4}$$

### 4.1 Analysis of the scheme

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

**Correctness.** The correctness of the scheme can be verified by the equation (5).

$$\begin{aligned}\sigma_2 &= e(g^{x_S y_S}(m' \textstyle\prod_{i\in\widetilde{M}} m_i)^r, g^{x_V})\\ &= e(g^{x_S y_S}, g^{x_V}) e((m' \textstyle\prod_{i\in\widetilde{M}} m_i)^r, g^{x_V})\\ &= e(g^{x_S}, g^{y_S})^{x_V} e((m' \textstyle\prod_{i\in\widetilde{M}} m_i), \sigma_1)^{x_V}\end{aligned} \tag{5}$$

**Theorem 1.** If there exists an adversary $A$ who can $(t, q_s + q_{sim}, q_v, \epsilon)$ forge the designated verifier signature scheme, then there exists another algorithm $B$ who can use $A$ to solve an instance of the GBDH problem in $(G, G_T)$ with probability $\epsilon'$ in time $t'$, such that

$$\begin{aligned}&\epsilon' \geq \tfrac{\epsilon}{8(n+1)(q_s+q_{sim}+q_v)}\\ &t' \leq (2n + 2 + 4(q_s + q_{sim}))T_1 + q_v T_2 + (q_s + q_{sim} + q_v)t_e + ((n + 2)(q_s + qsim) + 2 + nq_v)\\ &t_1 + 2q_v t_2 + t\end{aligned} \tag{6}$$

where $t_1$ and $t_2$ are the time for a multiplication in $G$ and $G_T$ respectively; $T_1$ and $T_2$ are the time for an exponentiation in $G$ and $G_T$ respectively; moreover, $t_e$ is the time for a pairing computation in $(G, G_T)$.

Proof. Let $A$ be a forger for the designated verifier signature. We use $A$ to construct another algorithm $B$ to break GBDH assumption with probability $\epsilon'$ in time $t'$. Given a random instance of GBDH problem $(g, g^a, g^b, g^c)$ of a bilinear group $(G, G_T)$, its goal is to output $e(g, g)^{abc}$ with the help of the DBDH oracle $O_{DBDH}$. $B$ will run $A$ as a subroutine and act as $A$' challenger to solve a random instance of GBDH problem. Hence, $B$ will response $A$'s queries in the following approach.

**Setup.** $B$ sets an integer $l = 4(q_s + q_{sim} + q_v)$ and chooses an integer $k$, uniformly at random between 0 and $n$. $B$ then chooses a value $x'$ and a random $n$-vector, $\boldsymbol{x} = (x_i)$ where $x', x_i \in Z_l$. Additionally, $B$ picks randomly a value $y'$ and a random $n$-vector, $\boldsymbol{y} = (y_i)$ where $y', y_i \in Z_p$. These values are kept internal to $B$.

For a message $M$, we let $\widetilde{M} \subseteq \{1, 2, ..., n\}$ be the set of all $i$ for which $M_i = 1$. To simplify the analysis as aforementioned in Water's scheme [26], we consider three functions $F(M) = (p - lk) + x' + \sum_{i\in\widetilde{M}} x_i$, $J(M) = y' + \sum_{i\in\widetilde{M}} y_i$ and $K(M)$ which takes the value 0 if $x' + \sum_{i\in\widetilde{M}} x_i = 0(\text{mod} l)$, takes 1, otherwise.

Afterwards, $B$ sets the public keys of users and common parameters as follows:

- $B$ assigns the public key of the signer $pk_S = (pk_{1S}, pk_{2S}) = (g^a, g^b)$ and the public key of the designated verifier $pk_V = g^c$ where $g^a$, $g^b$, and $g^c$ are the inputs of the GBDH problem.
- $B$ assigns $m' = pk_{2S}^{p-kl+x'} g^{y'}$ and $m_i = pk_{2S}^{x_i} g^{y_i}$and sets $\overrightarrow{m} = \{m_1, m_2, ..., m_n\}$

Hence, we have $(m' \prod_{i\in\widetilde{M}} m_i) = pk_{2S}^{F(M)} g^{J(M)}$. $B$ returns $(G, G_T, e, p, g, m', \overrightarrow{m})$ and $(pk_{1S}, pk_{2S}, pk_V)$ to $A$.

**Answering signature and simulation queries.** It is supposed that the adversary $A$ asks for a designated verifier signature on a $n$-bit message $M$. Thus, $B$ has to create a valid signature tuple without knowing the private key of $S$ or $V$. The simulator $B$ proceeds in the following approach.

- If $K(M) = 0$, $B$ terminates the simulation and reports failure.
- If $K(M) \neq 0$ which indicates that $F(M) \neq 0(\bmod p)$, since it is assumed $p > nl$ for any reasonable values of $p$, $n$, $l$ as mentioned in [26]. $B$ can construct a valid designated verifier signature by picking $r \in Z_p$ randomly and computes $\sigma = (\sigma_1, \sigma_2)$ where

$$\begin{aligned} \sigma_1 &= pk_{1S}^{\frac{-1}{F(M)}} g^r \\ \sigma_2 &= e(pk_{1S}^{\frac{-J(M)}{F(M)}} (m' \textstyle\prod_{i \in \widetilde{M}} m_i)^r, pk_V) \end{aligned} \tag{7}$$

**Correctness**

$$\begin{aligned} \sigma_2 &= e(pk_{1S}^{\frac{-J(M)}{F(M)}} (m' \textstyle\prod_{i \in \widetilde{M}} m_i)^r, pk_V) \\ &= e(pk_{1S}^{\frac{-J(M)}{F(M)}} (pk_{2S}^{F(M)} g^{J(M)})^r, g^{x_V}) \\ &= e(pk_{2S}^a ((pk_{2S})^{F(M)} g^{J(M)})^{\frac{-a}{F(M)}} ((pk_{2S})^{F(M)} g^{J(M)})^r, g^{x_V}) \\ &= e(pk_{2S}^a ((pk_{2S})^{F(M)} g^{J(M)})^{r - \frac{a}{F(M)}}, g^{x_V}) \\ &= e(g^{ab} ((pk_{2S})^{F(M)} g^{J(M)})^{\hat{r}}, g^{x_V}) \\ &= e(g^{ab} (m' \textstyle\prod_{i \in \widetilde{M}} m_i)^{\hat{r}}, g^{x_V}) \end{aligned} \tag{8}$$

Here, we have $\sigma_1 = pk_{1S}^{\frac{-1}{F(M)}} g^r = g^{r - \frac{a}{F(M)}}$.

**Answering verify queries.** Suppose $A$ issues a verify query for the message-signature pair $(M, \sigma = (\sigma_1, \sigma_2))$.

- If $F(M) = 0$, $B$ submits $(g, g^a, g^b, g^c, \frac{\sigma_2}{(e(g^c, \sigma_1))^{J(M)}})$ to the DBDH oracle $O_{DBDH}$. $B$ outputs "valid" if $O_{DBDH}$ returns 1; otherwise, $B$ returns "invalid".

**Correctness**

$$\begin{aligned} \sigma_2 &= e(pk_{1S}, pk_{2S})^{x_V} e((m' \textstyle\prod_{i \in \widetilde{M}} m_i), \sigma_1)^{x_V} \\ &= e(g^a, g^b)^c e(g^{J(M)}, \sigma_1)^c \\ &= e(g, g)^{abc} e(g^c, \sigma_1)^{J(M)} \end{aligned} \tag{9}$$

Which indicates $(g, g^a, g^b, g^c, \frac{\sigma_2}{(e(g^c, \sigma_1))^{J(M)}})$ is a valid BDH tuple.

- If $F(M) \neq 0$, $B$ can compute a valid signature on this message $M$ just as he responses to the designated verifier signature and simulation queries. Let $(M, \tilde{\sigma_1}, \tilde{\sigma_2})$ be the signature computed by $B$. Then $B$ submits $(g, (m' \prod_{i \in \widetilde{M}} m_i), \frac{\sigma_1}{\tilde{\sigma_1}}, g^c, \frac{\sigma_2}{\tilde{\sigma_2}})$ to the DBDH oracle. $B$ outputs "valid" if $O_{DBDH}$ returns 1. Otherwise, $B$ outputs "invalid".

**Correctness** If $(M, \sigma = (\sigma_1, \sigma_2))$ is a valid designated verifier signature, then we have

$$\sigma_2 = e(pk_{1S}, pk_{2S})^c e(m' \prod_{i \in \widetilde{M}} m_i, \sigma_1)^c \tag{10}$$

Similarly, since $(M, \tilde{\sigma_1} = (\tilde{\sigma_1}, \tilde{\sigma_2}))$ is another designated verifier signature computed by $B$, then we have

$$\tilde{\sigma_2} = e(pk_{1S}, pk_{2S})^c e(m' \prod_{i \in \widetilde{M}} m_i, \tilde{\sigma_1})^c \tag{11}$$

We can obtain

$$\begin{aligned} &(\tfrac{\sigma_2}{\tilde{\sigma_2}}) = (\tfrac{e(m' \prod_{i \in \widetilde{M}} m_i, \sigma_1)}{e(m' \prod_{i \in \widetilde{M}} m_i, \tilde{\sigma_1})})^{x_V} \\ &= e((m' \prod_{i \in \widetilde{M}} m_i), \tfrac{\sigma_1}{\tilde{\sigma_1}})^c. \end{aligned} \tag{12}$$

Therefore, $\frac{\sigma_2}{\tilde{\sigma_2}} = e(m' \prod_{i \in \widetilde{M}} m_i, \frac{\sigma_1}{\tilde{\sigma_1}})^c$ which indicates that $(g, (m' \prod_{i \in \widetilde{M}} m_i), (\frac{\sigma_1}{\tilde{\sigma_1}}), g^c, \frac{\sigma_2}{\tilde{\sigma_2}})$ is a valid BDH tuple.

If $B$ does not abort during the simulation, $A$ will output a valid designated verifier signature $\sigma^*$ on the message $M^*$ with success probability $\epsilon$.

**Probability analysis.** In order to compute the success probability of $B$, we consider events that $B$ will not abort. $B$ will not abort if both the two conditions hold as mentioned in [26].

- $\beta$: $B$ does not abort during the designated verifier signature, simulation, and verify queries.
- $\gamma$: $F(M^*) = 0(\mathrm{mod} p)$

The success probability of $B$ is $Suss_B^{GBDH} = Pr[\beta \wedge \gamma]\epsilon$

$$\begin{aligned} &Pr[\beta \wedge \gamma] = pr[\bigcap_{i=1}^{(q_s+q_{sim}+q_v)} K(M_i) \neq 0] pr[x + \sum_{i \in_{M^*}} x_i = lk|\beta] \\ &= (1 - pr[\bigcup_{i=1}^{(q_s+q_{sim}+q_v)} K(M_i)]) pr[x + \sum_{i \in_{M^*}} x_i = lk|\beta] \\ &\geq (1 - \tfrac{(q_s+q_{sim}+q_v)}{l}) pr[x + \sum_{i \in_{M^*}} x_i = lk|\beta] \\ &= \tfrac{1}{n+1}(1 - \tfrac{(q_s+q_{sim}+q_v)}{l}) \tfrac{pr[K(M^*)=0]}{pr(\beta)} pr[\beta|K[M^*] = 0] \\ &\geq \tfrac{1}{(n+1)l}(1 - \tfrac{(q_s+q_{sim}+q_v)}{l}) pr[\beta|K[M^*] = 0] \\ &\geq \tfrac{1}{(n+1)}(1 - \tfrac{(q_s+q_{sim}+q_v)}{l})(1 - pr[\bigcup_{i=1}^{(q_s+q_{sim}+q_v)} K(M_i) = 0|K[M^*] = 0]) \\ &= \tfrac{1}{(n+1)}(1 - \tfrac{(q_s+q_{sim}+q_v)}{l})^2 \\ &\geq \tfrac{1}{(n+1)l}(1 - \tfrac{2(q_s+q_{sim}+q_v)}{l}) \end{aligned} \tag{13}$$

Hence, $Succ_B^{GBDH} \geq \frac{1}{(n+1)l}(1 - \frac{2(q_s+q_{sim}+q_v)}{l})$ which is optimized by $l = 4(q_s + q_{sim} + q_v)$. Therefore, we have $Succ_B^{GBDH} \geq \frac{\epsilon}{8(n+1)(q_s+q_{sim}+q_v)}$

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

$$\sigma = (\sigma_1, \sigma_2) : \begin{cases} r \in_R Z_p^* \\ \sigma_1 = g^r (\mathrm{mod} p) \\ \sigma_2 = e(g^{x_S y_S} (m' \prod_{i \in \widetilde{M}} m_i)^r, g^{x_V}) \end{cases} \tag{14}$$

$$\sigma' = (\sigma'_1, \sigma'_2) : \begin{cases} r' \in_R Z_P^* \\ \sigma'_1 = g^{r'} (\mathrm{mod} p) \\ \sigma'_2 = e(g^{x_S}, g^{y_S})^{x_V} e((m' \prod_{i \in \widetilde{M}} m_i), \sigma'_1)^{x_V} \end{cases} \tag{15}$$

Let $\overline{\sigma} = (\overline{\sigma_1}, \overline{\sigma_2})$ be a valid signature which is randomly chosen from the set of all valid signer's signatures intended to the verifier $V$. Subsequently, we have distributions of probabilities as follows:

$$Pr_\sigma = Pr[(\sigma_1, \sigma_2) = (\overline{\sigma_1}, \overline{\sigma_2})] = Pr_{r \in_R Z_p^*} \begin{bmatrix} \sigma_1 = \overline{\sigma_1} \\ \sigma_2 = \overline{\sigma_2} \end{bmatrix} = \frac{1}{p-1}, \tag{16}$$

and

$$Pr_{\sigma'} = Pr[(\sigma'_1, \sigma'_2) = (\overline{\sigma_1}, \overline{\sigma_2})] = Pr_{r' \in_R Z_p^*} \begin{bmatrix} \sigma'_1 = \overline{\sigma_1} \\ \sigma'_2 = \overline{\sigma_2} \end{bmatrix} = \frac{1}{p-1} \tag{17}$$

The analysis means both distribution of probability are the same. Hence, our proposal satisfies the non-transferable property.

**Theorem 3.** If there exists an adversary $A$ who can $(t, q_s, q_v, \epsilon)$ break the PSI of the scheme, then there exists another algorithm $B$ who can use $A$ to solve an instance of the DBDH problem in $(G, G_T)$ with probability $\epsilon'$ in time $t'$, where

$$\begin{array}{l} \epsilon' \leq \epsilon_{DBDH} \\ t' \geq (n + 3 + q_s)T_1 + (q_v + q_s)T_2 + (q_s + q_v)t_e + n(q_s)t_1 + (q_v + q_s)t_2 + t \end{array} \tag{18}$$

where $t_1$ and $t_2$ are the time for a multiplication in $G$ and $G_T$ respectively; $T_1$ and $T_2$ are the time for an exponentiation in $G$ and $G_T$ respectively; moreover, $t_e$ is the time for a pairing computation in $(G, G_T)$.

Proof. Let $A$ be the distinguisher against privacy of a signer's identity. We use $A$ to construct another algorithm $B$ to break DBDH assumption with probability $\epsilon'$ in time $t'$. Given a random instance of DBDH problem of a bilinear group $(G, G_T)$, i.e. $(g, g^{a_0}, g^{a_1}, g^c, Z)$ where $a_0$, $a_1$, and $c$ are random elements of $Z_p$ unknown to it, $B$'s goal is to output whether $e(g, g)^{a_0 a_1 c} = Z$.

**Setup.** $B$ chooses a value $y'$ and a random $n$-vector, $\boldsymbol{y} = (y_i)$ where $y', y_i \in Z_p$. These values are kept internal to $B$.
For a message $M$, we let $\widetilde{M} \subseteq \{1, 2, ..., n\}$ be the set of all $i$ for which $M_i = 1$. We define a function $J'(M) = y' + \sum_{i \in \widetilde{M}} y_i$. Then $B$ sets the public keys of users and common parameters as follows:

- $B$ randomly chooses $b_0$ and $b_1 \in Z_p$ and sets the public keys of the two signers $pk_{S0} = (g^{a_0}, g^{a_1})$ and $pk_{S1} = (g^{b_0}, g^{b_1})$. $B$ sets the common secret key between $S_0$ and $V$, $k_{S_0 V} = Z$, the common secret key between $S_1$ and $V$, $k_{S_1 V} = e(g^{b_0}, g^c)^{b_1}$, and the public key of the designated verifier $pk_V = g^c$ where $g^{a_0}$, $g^{a_1}$, $g^c$, and $Z$ are the inputs to the DBDH problem.

- $B$ assigns $m' = g^{y'}$ and $m_i = g^{y_i}$ and sets $\overrightarrow{m} = (m_1, m_2, ..., m_n)$

Hence, we have $(m' \prod_{i \in \widetilde{M}} m_i) = g^{J'(M)}$. $B$ returns $(G, G_T, e, p, g, m', \boldsymbol{m})$ and $(pk_{S_0}, pk_{S_1}, pk_V)$ to $A$.

**Answering signature queries.** Suppose the adversary $A$ asks for a designated verifier signature on a $n$-bit message $M$ from the signer $S_d$ where $d \in \{0, 1\}$. $B$ has to create a valid signature tuple. The simulator $B$ proceeds as follows:

- $B$ can construct a valid designated verifier signature w.r.t. common secret key between the signer $S_d$ and the designated verifier $V$, $k_{S_dV}$, by picking $r \in Z_p$ randomly and computing $\sigma = (\sigma_1, \sigma_2)$ where

$$\begin{array}{l} \sigma_1 = g^r \\ \sigma_2 = k_{S_dV} e((m' \prod_{i \in \widetilde{M}} m_i), g^c)^r \end{array} \tag{19}$$

**Answering verify queries.** Suppose $A$ issues a verify query for the message-signature pair $(M, \sigma, d)$, $B$ can verify the signature $(M, \sigma, d)$ using $\sigma_2 = k_{S_dV} e(\sigma_1, g^c)^{J'(M)}$ since it knows $k_{S_dV}$ and $J'(M)$. When $A$ submits its challenge message $M^*$, $B$ chooses a random bit $d \in_R \{0, 1\}$ and returns $\sigma^*$. The successive queries issued by $A$ are handled as mentioned above. Finally, $A$ outputs a bit $d'$. Then, $B$ outputs 1 if $d' = d$, indicating $Z = e(g, g)^{a_0 b_0 c}$ and 0 otherwise, indicating $Z$ is a random element of $G$. Let $b$ be the bit $B$ outputs. We have

$$\begin{array}{l} \epsilon' = |pr[b = 1 \wedge Z = e(g, g)^{a_0 a_1 c}] - pr[b = 1 \wedge Z \longleftarrow_R G_T]| \\ \quad = \frac{1}{2}(|pr[b = 1 | Z = e(g, g)^{a_0 a_1 c}] - pr[b = 1 | Z \longleftarrow_R G_T]|) \leq \frac{1}{2}\epsilon_{DBDH} \leq \epsilon_{DBDH} \end{array} \tag{20}$$

Our proposed scheme is non-delegatable for signature verification. Informally, it is assumed that the common secret key between $S$ and $V$, $k_{SV} = e(g^{x_S}, g^{y_S})^{x_V}$, and a SDVS signature $(M, \sigma = (\sigma_1, \sigma_2))$ are given to a third party. To verify the validity of the signature in the relation $\sigma_2 = k_{SV} e(m' \prod_{i \in \widetilde{M}} m_i, \sigma_1)^{x_V}$, she still needs to know the designated verifier's secret key. Non-delegatability for signature verification in theorem 4 is based on BDH assumption which can be converted to the BDL assumption in some conditions [17] which is equivalent with the definition of non-delegatability for signature verification. The proposal is delegatable for signing: a signer $S$ or a verifier $V$ can release $k_{SV}$; hence, any third party can sign a message $M$ on behalf of the signer for the verifier. To this purpose, the third party chooses $r \in_R Z_p^*$ and computes $\sigma_1 = g^r$ and $\sigma_2 = k_{SV} e((m' \prod_{i \in \widetilde{M}} m_i)^r, g^{x_V})$. To verify the validity of this signature $(M, \sigma = (\sigma_1, \sigma_2))$, the verifier $V$ acts as explained in Eq.(3).

**Theorem 4.** If there exists an adversary $A$ who can $(t, q_s + q_{sim}, \epsilon)$ violate the property of non-delegatability for signature verification, then there exists another algorithm $B$ who can solve an instance of the BDH problem in $(G, G_T)$ with probability $\epsilon'$ in time $t'$ such that

$$\begin{array}{l} \epsilon' \geq \frac{\epsilon}{8(n+1)(q_s + q_{sim})} \\ t' \leq (2n + 2 + 4(q_s + q_{sim}))T_1 + T_2 + (q_s + q_{sim} + 1)t_e + ((n + 2)(q_s + qsim) + 2 + n) \\ t_1 + 2t_2 + t \end{array} \tag{21}$$

where $t_1$ and $t_2$ are the time for a multiplication in $G$ and $G_T$ respectively; $T_1$ and $T_2$ are the time for an exponentiation in $G$ and $G_T$ respectively; moreover, $t_e$ is the time for a pairing computation in $(G, G_T)$.

Proof. Let $A$ be a verifier for the designated verifier signature. We use $A$ to construct another algorithm $B$ to break BDH assumption with probability $\epsilon'$ in time $t'$. Given a random instance of BDH problem $(g, g^a, g^b, g^c)$ of a bilinear group $(G, G_T)$, its goal is to output $e(g, g)^{abc}$ with the help of the verifier $A$. $B$ will run $A$ as a subroutine and act as $A$'s challenger to solve the instance of BDH problem. $B$ will response $A$'s signature and simulation queries as mentioned in the proof of theorem 1. If $B$ does not abort during the simulation and $A$ can output 1, which means that the signature is valid, $B$ can compute the value of $e(g, g)^{abc}$ in case of $F(M) = 0$ using Eq. (22).

$$e(g^a, g^b)^c = (\frac{\sigma_2}{e(g^c, \sigma_1)})^{J(M)} \tag{22}$$

If $B$ does not abort during the simulation, $A$ will output 1 with success probability $\epsilon$ which means the signature $\sigma^*$ on the message $M^*$ is valid.

**Probability analysis** In order to compute the success probability of $B$, we consider events that $B$ will not abort. $B$ will not abort if both the two conditions hold as aforementioned in theorem 1.

- $\beta$: $B$ does not abort during the designated verifier signature and simulation queries.
- $\gamma$: $F(M^*) = 0(\mathrm{mod} p)$

The success probability of $B$ is $Suss_B^{BDH} = Pr[\beta \wedge \gamma]\epsilon$ which is computed in theorem 1.

Note that, BDH problem polytime reduces to the BDL problem in some conditions [17] which is equivalent to the Definition 8.

## 5 Comparison

As a comparison, we consider the existing SDVS schemes in the standard model as shown in Table 1. Security of our scheme is only based on standard complexity assumptions, while the security of Huang et al. scheme [10] is based on security of PRF in addition to standard complexity assumptions. The proposed scheme has the security proof for PSI, while the scheme in [28] does not have security proof for PSI. Furthermore, our proposal is non-delegatable for signature verification, while two schemes do not have this property.

| **schemes** | Unforge. | Non-dele. | PSI | Non-trans. |
|---|---|---|---|---|
| **Huang et al. 2011** | ✓ | × | ✓ | ✓ |
| **Zhang et al. 2007** | ✓ | × | × | ✓ |
| **Our Scheme** | ✓ | ✓ | ✓ | ✓ |

**Table 1.** Comparison table based on properties(✓: satisfied, ×: unsatisfied)

Note that, non-delegatability in Table 1 means non-delegatability for signature verification.

## 6 conclusion

We propose a novel designated verifier scheme and prove that the scheme is secure without random oracles. To the best of our knowledge, this is the first designated verifier signature scheme that has non-delegatability for signature verification in the standard model. The security of our scheme relies on standard complexity assumptions not security of PRF or other primitives.

## References

1. 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).
2. 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).
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. 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).
5. 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).
6. 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).
7. 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).
8. 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).
9. 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.
10. 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.
11. 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).
12. 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).
13. 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).
14. 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).

15. 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).
16. 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).
17. Maurer, U., *Towards proving the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms,* Advances in Cryptology - Crypto '94, Lecture Notes in Computer Science, Vol. 839, pp. 271-281, 1994.
18. 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).
19. Saeednia, S., Kremer, S., Markowitch, O., *An efficient strong designated verifier signature scheme,* Proceedings of 6th International Conference on Information Security and Cryptology, ICISC 2003, vol. 2971 of Lecture Notes in Computer Science, pp. 40-54, Springer (2003).
20. 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).
21. 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).
22. 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 (2004).
23. 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 (2004).
24. 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 (2006).
25. Wang, B., Song, Z., *A non-interactive deniable authentication scheme based on designated verifier proofs,* Information Sciences, Inf. Sci. 2009, vol. 179(6), pp. 858- 865, 2009.
26. Waters, B., *Efficient identity based encryption without random oracles,* Eurocrypt 2005, vol. 3494 of LectureNotes in Computer Science, pp. 114-127, Springer (2005)
27. 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 (2005).
28. Zhang, J. and Ji, C., *An efficient designated verifier signature scheme without Random Oracles,* First International Symposium on Data, Privacy and E-Commerce, ISDPE 2007, pp.338-340, 2007.