Underlying Assumptions and Designated Verifier Signatures

Chifumi Sato[1], Takeshi Okamoto[2], and Eiji Okamoto[3]

[1] SBI Net Systems Co., Ltd., Sumitomo Ichigaya Bldg., 16th Floor, 1–1 Ichigaya-honmuracho Shinjuku-ku, Tokyo, 162–0845, Japan
c-sato@sbins.co.jp

[2] Department of Computer Science, Faculty of Health Sciences, Tsukuba University of Technology, 4–12–7 Kasuga Tsukuba-shi, Ibaraki, 305–0821, Japan
ken@cs.k.tsukuba-tech.ac.jp

[3] Graduate School of Systems and Information Engineering, University of Tsukuba, 1–1–1 Tennodai Tsukuba-shi, Ibaraki, 305–8573, Japan
okamoto@risk.tsukuba.ac.jp

Abstract. In this paper, we define an underlying computational problem and its decisional problem. As an application of their problems, we propose an efficient designated verifier signature (DVS) scheme without random oracles (related to symmetric pairings). We formally redefine the (Strong) Privacy of Signature's Identity, and prove our DVS scheme satisfying security based on the difficulty of the problems. Also we prove that the difficulty of the computational problem is tightly equivalent to the Strong Unforgeability of our proposed conventional signature scheme (without random oracles) related to asymmetric pairings. We believe that our underlying problems are profitable to propose many efficient cryptographic schemes.

Keywords: Designated verifier signatures, Digital signatures, Standard model

1 Introduction

Security of cryptographic protocols and schemes in public-key infrastructure is usually reduced problems which everyone believes difficult to solve. For example, the Unforgeabilty (or Existential Unforgeability under an Adaptive Chosen Message Attack) of the Waters conventional signature scheme [16] is based on the difficulty of the Computational Diffie–Hellman (CDH) problem.

In this paper, we define an underlying computational problem and its decisional problem, which are stronger than the CDH problem. As an application of our problems, we propose a designated verifier signature (DVS) scheme without random oracles related to symmetric pairings. DVSes, introduced by [6], are signatures that will be only convinced by a (specific) designated verifier whether valid or invalid. The verifier cannot transfer the signature to a third party.

Our scheme satisfies the following security: Correctness, Strong Unforgeability (or Strong Unforgeability under an Adaptive Chosen Message Attack), Non-Transferability, and (Strong) Privacy of Signature's Identity. The security is based on the difficulty of our proposed problems. Our scheme is delegatable, so the signer can delegate the third party a right of signer's signing. From the delegatability, for example our DVS scheme is suitable for the following two cases. The first case is that a signer does not want to delegate such as an authentication associated with a payment. The second case is that a designated verifier allows for the signer to delegate such as an e-ticket for some service. However, our scheme is not suitable for an e-election since the signer is able to sell his own suffrage to a third party.

Huang–Yang–Wong–Susilo [5] provided a highly efficient DVS scheme without random oracles, which is to be secure if pseudo-random functions have the pseudo-randomness. The (t, ε)-pseudo-randomness means that outputs of the functions can only be distinguished from those of the random oracles in polynomial times t within probabilities $(1/2 - \varepsilon, 1/2 + \varepsilon)$. However the value ε must be negligible if the scheme is secure. Our scheme is more efficient than DVS schemes [17, 10] whose security can be proven without the assumption of random oracles, and without that of the pseudo-randomness.

On the other hand, it is known that the difficulty of solving the RSA problem is tightly equivalent to be the difficulty that ciphertexts of the RSA cryptosystem are perfectively broken against chosen ciphertext attacks. Also the difficulty of solving the CDH problem is tightly equivalent to be the difficulty that agreement keys of the primitive Diffie–Hellman key agreement [3] are perfectively broken against passive adversary attacks.

In this paper, we prove that the difficulty of our computational problem is tightly equivalent to Strong Unforgeability of our proposed conventional signature scheme. Unfortunately, this scheme is not enough to be efficient since the scheme is constructed based on asymmetric pairings. However, we believe that the result is important to justify our proposed problem. More detailed description is in Section 3.1.

The paper is organized in the following way. In Section 2, we prepare for the construction of our proposals. In Section 3, we will provide three proposals: a computational problem, a decisional problem and a DVS scheme. We describe security and efficiency of our DVS scheme in Section 4. In Section 5, we propose a conventional signature scheme and describe its security. We provide conclusions in Section 6.

2 Preliminaries

In this section, we state the definition of a symmetric pairing (or bilinear map). This definition is due to [2].

We assume that

- $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p;

– $e: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is the cryptographic symmetric *pairing* satisfying the following properties:
Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$ for any $u, v \in \mathbb{G}$ and any $a, b \in \mathbb{Z}$.
Non-degenerate: $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$ for $\langle g_1 \rangle = \langle g_2 \rangle = \mathbb{G}$.
Computable: There is an efficient algorithm to compute $e(u, v)$ for any $u, v \in \mathbb{G}$.

3 Our Proposals

In this section, we define an underlying computational problem and its decisional problem, and propose a DVS scheme without random oracles.

3.1 Proposed Computational Problem

We provide Assumption 1 related to the Computational Diffie–Hellman (CDH) Assumption. Here we say that the (t, ε)-CDH Assumption in $\mathbb{G}$ holds if for any adversary $\mathcal{A}$ running in time t and an advantage ε, we have that

$$\Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab}\right] < \varepsilon$$

where the probability is over the choice of a random generator $g \in_{\mathrm{R}} \mathbb{G}$, random numbers $a, b \in_{\mathrm{R}} \mathbb{Z}_p^*$ and the random bits of $\mathcal{A}$.

The problem is defined as follows. Given

$$\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i} \,\middle|\, i = 1, \ldots, q\right) \tag{1}$$

as input for random generators $g_1, g_2 \in_{\mathrm{R}} \mathbb{G}$ and random numbers $x, r_1, \ldots, r_q \in_{\mathrm{R}} \mathbb{Z}_p^*$, compute $\left(g_2^{r_*}, g_2^{x+1/r_*}\right)$ for some $r_* \in \mathbb{Z}_p^*$ and $r_* \notin \{r_1, \ldots, r_q\}$. Note that the index $x + 1/r_i$ means $x + (1/r_i)$. We say that algorithm $\mathcal{A}$ has an advantage ε in solving the problem if

$$\Pr\left[\mathcal{A}\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i} \,\middle|\, i = 1, \ldots, q\right) = \left(g_2^{r_*}, g_2^{x+1/r_*}\right) \,\middle|\, \begin{array}{l} r_* \in \mathbb{Z}_p^* , \\ r_* \notin \{r_1, \ldots, r_q\} \end{array}\right] \geq \varepsilon \ ,$$

where the probability is over the choice $g_1, g_2 \in_{\mathrm{R}} \mathbb{G}$, $x, r_1, \ldots, r_q \in_{\mathrm{R}} \mathbb{Z}_p^*$ and the random bits of $\mathcal{A}$.

Assumption 1 A (q, t, ε)-Computational Assumption I holds in $\mathbb{G}$ if no t-time adversary has an advantage of at least ε in solving the problem in $\mathbb{G}$.

Our problem is similar to those of [13, 14]. However, security of all schemes in [13, 14] is based on the problems with asymmetric pairings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. In particular, concerning how to compare signature lengths of the ID-based signature scheme [14], it assume that the representation of elements in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{Z}_p$ takes the same length, which is actually not true for asymmetric pairings.

In general, $|\mathbb{Z}_p| \leq |\mathbb{G}_1|$ and $|\mathbb{G}_2| \geq 2|\mathbb{G}_1|$. Then, in the best case, $|\mathbb{Z}_p| = |\mathbb{G}_1|$ and $|\mathbb{G}_2| = 2|\mathbb{G}_1|$, so we can use as unit $|\mathbb{Z}_p|$. Then, a signature in the scheme consists of 4 elements in $\mathbb{G}_2$ and 1 element in $\mathbb{G}_1$, amounting then to 9 such units. Now, if we use the generic construction from [4], applied to the DLP-based strong one-time signature in Appendix A of [1] and the ID-based signature (IBS) scheme in [11], we obtain a strongly unforgeable IBS consisting of 6 elements in $\mathbb{G}_1$ and 2 elements in $\mathbb{Z}_p$, and a computational cost dominated by 3 pairings, while relying on better computational assumptions than the IBS scheme [14].

On the other hand, our proposed problem is based on symmetric pairings $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Then we will construct an efficient designated verifier signature scheme. It is not easy to compare the problems either on asymmetric pairings or on symmetric ones. Our definition is profitable to propose many efficient cryptographic schemes.

Difficulty of solving our problem based on symmetric pairings is equivalent to be strongly unforgeable for a proposed DVS scheme. This is part of security in the DVS scheme. If the problem is based on asymmetric pairings, the problem is equivalent to be strong unforgeable for a proposed scheme. This is a main security of the conventional signature scheme.

3.2 Proposed Decisional Problem

The following problem is a decisional one of the above computational problem.

Given

$$\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i}, g_2^{r_*}, R \;\middle|\; i = 1, \ldots, q\right) \tag{2}$$

as input for random generators $g_1, g_2 \in_{\mathrm{R}} \mathbb{G}$ and random numbers $x, r_1, \ldots, r_q \in_{\mathrm{R}} \mathbb{Z}_p^*$, $r_* \in \mathbb{Z}_p^*$, $r_* \notin \{r_1, \ldots, r_q\}$ and $R \in \mathbb{G}$, output 1 if $R = g_2^{x+1/r_*}$ and output 0 otherwise. We say that algorithm $\mathcal{A}$ has an advantage ε in solving the problem if

$$\begin{aligned} \Big| \; &\Pr\left[\mathcal{A}\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i}, g_2^{r_*}, R \;\middle|\; i = 1, \ldots, q\right) = 1\right] \\ &-\Pr\left[\mathcal{A}\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i}, g_2^{r_*}, g_2^{x+1/r_*} \;\middle|\; i = 1, \ldots, q\right) = 1\right] \Big| \geq \varepsilon \;, \end{aligned}$$

where the probability is over the choice $g_1, g_2, R \in_{\mathrm{R}} \mathbb{G}$, $x, r_1, \ldots, r_q \in_{\mathrm{R}} \mathbb{Z}_p^*$, $r_* \in \mathbb{Z}_p^*, r_* \notin \{r_1, \ldots, r_q\}$ and the random bits of $\mathcal{A}$.

Assumption 2 A (q, t, ε)-Decisional Assumption II holds in $\mathbb{G}$ if no t-time adversary has an advantage of at least ε in solving the problem in $\mathbb{G}$.

3.3 A Designated Verifier Signature Scheme $\mathcal{DVS}$

We give a designated verifier signature (DVS) scheme with four phases: *DVS.Setup, DVS.SKeyGen, DVS.VKeyGen, DVS.Sign*, and *DVS.Verify*. Definition of the DVS scheme is based on [9]. For the moment we shall assume that the signature message M is an element in $\mathbb{Z}_p$, but the domain can be extended to all of $\{0,1\}^*$ using a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$.

DVS.Setup: Choose multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of sufficiently large prime order p. Assume that $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a symmetric pairing. The public key is generated as follows. Choose a random generator, $g_1 \in \mathbb{G}$. Public parameters are

$$\texttt{params} := (\mathbb{G}, \mathbb{G}_T, p, e, g_1) \ .$$

DVS.SKeyGen: A signer S generates $(x, y) \in_{\mathrm{R}} (\mathbb{Z}_p^*)^2$, and calculate $(X, Y) := (g_1^x, g_1^y) \in \mathbb{G}^2$. The public and private keys of S are:

$$PK_S := (X, Y) \quad \text{and} \quad SK_S := (x, y) \ ,$$

respectively.

DVS.VKeyGen: A designated verifier D generates $g_2 := g_1^d \in \mathbb{G}$ from $d \in_{\mathrm{R}} \mathbb{Z}_p^*$, sets $z := e(g_1, g_2)$, and sends (g_2, z) to the signer. The public and private keys of D are:

$$PK_D := (g_2, z) \quad \text{and} \quad SK_D := d \ ,$$

respectively.

DVS.Sign: A designated signature of a message $M \in \mathbb{Z}_p$ is generated as follows. First, a random $r \in \mathbb{Z}_p^*$ is chosen. The designated signature is then constructed as:

$$\sigma := \left(g_2^r, g_2^{x+My+1/r} \right) \in \mathbb{G}^2 \ . \tag{3}$$

DVS.Verify: Suppose we wish to check if $\sigma = (\sigma_1, \sigma_2)$ is a designated signature for a message M. The designated verifier verifies

$$e \left(X^{-1} \cdot Y^{-M} \cdot \sigma_2^{1/d}, \sigma_1 \right) = z \ .$$

If the equality holds the result is `valid`; otherwise the result is `invalid`.

4 Security and Efficiency of Our DVS Scheme

In this section, we prove that our proposed DVS scheme in Section 3.3 satisfies security: the Correctness, Strong Unforgeability, Non-Transferability, (Strong) Privacy of Signer's Identity, and Delegability. Also, we describe efficiency of our DVS scheme.

4.1 Correctness

If a signer S with the public key PK_S constructs a signature σ on a message M as described in the *DVS.Sign* phase above, it is easy to see that σ will be accepted by a designated verifier D:

$$\begin{aligned} e\left(X^{-1} \cdot Y^{-M} \cdot \sigma_2^{1/d}, \sigma_1\right) &= e\left((g_1^x)^{-1} \cdot (g_1^y)^{-M} \cdot \left(g_2^{x+My+1/r}\right)^{1/d}, g_2^r\right) \\ &= e\left(g_1^{-x} \cdot g_1^{-My} \cdot g_1^{x+My+1/r}, g_2^r\right) \\ &= e\left(g_1^{1/r}, g_2^r\right) \\ &= e(g_1, g_2) \ . \end{aligned}$$

Thus the scheme is correct.

4.2 Strong Unforgeability

Definition of Strong Unforgeability for DVS is based on the Unforgeability in [10], by changing the third requirement in the *Output* phase to the signature is not output as a response to a *Query* phase.

Theorem 1. *Suppose that the $(q_0, t_0, \varepsilon_0)$-Computational Assumption I holds in $\mathbb{G}$. Then the proposed designated verifier signature scheme $\mathcal{DVS}$ is (q, t, ε)-strongly unforgeable, provided that $q \leq q_0$, $t \leq t_0 - O(qT)$ and $\varepsilon \geq 2\varepsilon_0$, where T is the maximum time for an exponentiation in $\mathbb{G}$.*

An outline of the proof is as follows. Suppose that there exists an adversary, $\mathcal{A}$, who breaks the Strong Unforgeability for $\mathcal{DVS}$, and a challenger, $\mathcal{B}$, takes the Assumption I challenge. After $\mathcal{A}$ and $\mathcal{B}$ execute the strongly unforgeable game, $\mathcal{A}$ outputs a valid tuple for a message and a signature. Then $\mathcal{B}$ replies the Assumption I response with non-negligible probability. This is a contradiction to hold the Assumption I.

Although we can prove that $\mathcal{DVS}$ is strongly unforgeable if and only if the Assumption I holds in $\mathbb{G}$. However the Strong Unforgeability is part of security in the DVS scheme, so we will omit to prove it in this paper.

Proof. Suppose that there exists an adversary, $\mathcal{A}$, who breaks the (q, t, ε)-Strong Unforgeability (SUF) of our DVS scheme $\mathcal{DVS}$:

$$\Pr\left[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-SUF of } \mathcal{DVS}\right] \geq \varepsilon \ . \tag{4}$$

We construct a simulator, $\mathcal{B}$, to play the Computational Assumption I game. The simulator $\mathcal{B}$ will take the Assumption I challenge (1) for $x, r_i \in_{\mathrm{R}} \mathbb{Z}_p^*$ $(i = 1, \ldots, q)$ and run $\mathcal{A}$ executing the following steps.

Simulator Description
Setup: The simulator $\mathcal{B}$ sends

$$\texttt{params} = (\mathbb{G}, \mathbb{G}_T, p, e, g_1)$$

to the adversary $\mathcal{A}$. $\mathcal{B}$ generates $g_1^y \in \mathbb{G}$ from $y \in_{\mathrm{R}} \mathbb{Z}_p^*$. He generates $b \in_{\mathrm{R}} \{0,1\}$, and sends to $\mathcal{A}$ either

$$\begin{cases} \textbf{(4.2.1-1)} & PK_S = (g_1^x, g_1^y) \text{ if } b = 0 \text{ , or} \\ \textbf{(4.2.1-2)} & PK_S = (g_1^y, g_1^x) \text{ if } b = 1 \text{ .} \end{cases}$$

The adversary $\mathcal{A}$ cannot know whether the received parameter is in the cases (4.2.1-1) or (4.2.1-2). Also $\mathcal{B}$ sends

$$PK_D = (g_2, z)$$

to the adversary $\mathcal{A}$, where $z = e(g_1, g_2)$.

Signature Queries: The adversary $\mathcal{A}$ issues signature queries $M_1, \ldots, M_q$. These queries may be asked adaptively so that each query M_i may depend on the replies to $M_1, \ldots, M_{i-1}$.

In the case (4.2.1-1), the simulator $\mathcal{B}$ generates the signature

$$(\sigma_{i,1}, \sigma_{i,2}) := \left(g_2^{r_i}, \left(g_2^{x+1/r_i} \right) \cdot (g_2^y)^{M_i} \right) \left(= \left(g_2^{r_i}, g_2^{x+M_i y+1/r_i} \right) \right) ,$$

and sends it to $\mathcal{A}$. In the case (4.2.1-2) and $M_i \neq 0$, $\mathcal{B}$ generates

$$(\sigma_{i,1}, \sigma_{i,2}) := \left((g_2^{r_i})^{1/M_i}, \left(g_2^{x+1/r_i} \right)^{M_i} \cdot g_2^y \right) \left(= \left(g_2^{r_i/M_i}, g_2^{y+M_i x+M_i/r_i} \right) \right) ,$$

and sends it to $\mathcal{A}$. In the case (4.2.1-2) and $M_i = 0$, $\mathcal{B}$ generates $(\sigma_{i,1}, \sigma_{i,2}) := \left(g_2^{r'_i}, g_2^{y+1/r'_i} \right)$ for $r'_i \in_{\mathrm{R}} \mathbb{Z}_p^*$, and sends it to $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs (M_*, σ_*) such that $\sigma_* = (\sigma_{*,1}, \sigma_{*,2}) \in \mathbb{G}^2$ is a valid signature of M_*, and $(M_*, \sigma_{*,1}, \sigma_{*,2}) \notin \{(M_1, \sigma_{1,1}, \sigma_{1,2}), \ldots, (M_q, \sigma_{q,1}, \sigma_{q,2})\}$.

Analysis
The signature σ_* is valid. In the case (4.2.1-1), assume that $(\sigma_{*,1}, \sigma_{*,2}) := \left(g_2^{r_*}, g_2^{x+M_* y+1/r_*} \right)$ for $r_* \in \mathbb{Z}_p^*$ and

$$(M_*, r_*) \neq (M_i, r_i) \tag{5}$$

for $i = 1, \ldots, q$ (i.e. $(M_*, r_*) \notin \{(M_1, r_1), \ldots, (M_q, r_q)\}$). Also, in the case (4.2.1-2), assume that $(\sigma_{*,1}, \sigma_{*,2}) := \left(g_2^{r_*}, g_2^{y+M_* x+1/r_*} \right)$ for $r_* \in \mathbb{Z}_p^*$ and

$$(M_*, r_*) \not\equiv \begin{cases} (M_i, r_i/M_i) \pmod p & \text{for } M_i \neq 0 \text{ ,} \\ (0, r'_i) \pmod p & \text{for } M_i = 0 \end{cases}$$

for $i = 1, \ldots, q$.

(4.2.2-1.1) If $r_* \notin \{r_1, \ldots, r_q\}$ in the case (4.2.1-1), the simulator $\mathcal{B}$ calculates $\left(g_2^{r_*}, g_2^{x+1/r_*}\right) = \left(\sigma_{*,1}, \sigma_{*,2} \cdot (g_2^y)^{-M_*}\right)$ which is a valid output for the Assumption I game.

(4.2.2-1.2) Otherwise in the case (4.2.1-1), assume that $r_* = r_j$ $(1 \leq j \leq q)$. Then we have $M_* \neq M_j$ from (9). Though the simulator $\mathcal{B}$ can calculate $g_2^y = (\sigma_{*,2}/\sigma_{j,2})^{1/(M_* - M_j)}$ from the only $(M_*, \sigma_{*,2})$ and $(M_j, \sigma_{j,2})$, he does not seem to propose a valid output for the Assumption I game. However, notice that $\mathcal{A}$ does not know whether $\mathcal{B}$ simulates (4.2.2-1.2) or the following (4.2.2-2.1).

(4.2.2-2.1) In the case (4.2.1-2), if there exists an index j $(1 \leq j \leq q)$ such that

$$r_* \equiv \begin{cases} r_j/M_j & \text{for } M_* \neq M_j \text{ and } M_j \neq 0 \ , \\ r'_j & \text{for } M_* \neq 0 \text{ and } M_j = 0 \ , \end{cases}$$

the simulator $\mathcal{B}$ can calculate g_2^x such as (4.2.2-1.2). He can generate $\left(g_2^{r'_*}, g_2^{x+1/r'_*}\right)$ for $r'_* \notin \{r_1, \ldots, r_q\}$, which is a valid output for the Assumption I game.

(4.2.2-2.2) Otherwise in the case (4.2.1-2), assume that $r_* \not\equiv r_i/M_i$ for $M_i \neq 0$, and assume that $r_* \neq r'_i$ for $M_i = 0$. Since $\mathcal{B}$ cannot obtain new information of solving the Assumption I game from $\mathcal{A}$ at the simulator description in Section 4.2.1, it seems that $\mathcal{B}$ cannot propose a valid output for the game.

Let $\Pr[\mathcal{B} \text{ breaks } (q_0, t_0, \varepsilon_0)\text{-Assumption I}]$ be the probability that $\mathcal{B}$ generates a valid output for the Assumption I game.

Lemma 1.

$$\frac{1}{2}\, \mathit{Pr}[\mathcal{A} \textit{ breaks } (q, t, \varepsilon)\textit{-SUF of } \mathcal{DVS}] \leq \mathit{Pr}[\mathcal{B} \textit{ breaks } (q_0, t_0, \varepsilon_0)\textit{-Assumption I}] \ .$$

Proof of this lemma is proposed in Appendix A. From the assumption in Theorem 1,

$$\Pr[\mathcal{B} \text{ breaks } (q_0, t_0, \varepsilon_0)\text{-Assumption I}] < \varepsilon_0 \ . \tag{6}$$

From (4), (10) and Lemma 1, we have

$$\begin{aligned} \frac{\varepsilon}{2} &\leq \frac{1}{2} \Pr[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-SUF of } \mathcal{DVS}] \\ &\leq \Pr[\mathcal{B} \text{ breaks } (q_0, t_0, \varepsilon_0)\text{-Assumption I}] < \varepsilon_0 \ , \end{aligned}$$

which is a contradiction to the assumption $\varepsilon/2 \geq \varepsilon_0$ in Theorem 1.

Therefore, we have $\Pr[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-SUF of } \mathcal{DVS}] < \varepsilon$. $\square$

4.3 Non-Transferability

Definition of Non-Transferability is based on [15]. We show that, if designated signatures can be simulated by the verifier himself then a designated signature adds no computational ability to the verifier.

From verifier's secret d, the verifier generates

$$\left((g_1^r)^d, \left(g_1^x \cdot (g_1^y)^M \cdot g_1^{1/r}\right)^d\right) ,$$

which is equal to (3). Therefore the proposed DVS scheme satisfies the Non-Transferability.

4.4 Privacy of Signature's Identity

The (Strong) Privacy of Signature's Identity (PSID) is given in [8] as follows. Given a message M and a designated verifier signature σ of this message, it is computationally infeasible, without the knowledge of the private key of the designated verifier or the one of the signer, to determine which pair of signing keys was used to generate σ.

We formally define (q, t, ε)-PSID as follows. This security is defined using the following game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$:

Setup: The challenger $\mathcal{B}$ takes a security parameter k and runs the *DVS.Setup* phase of the DVS scheme. It gives the adversary $\mathcal{A}$ the resulting system parameters `params`. $\mathcal{B}$ runs the *DVS.SKeyGen* phase, and sends $\mathcal{A}$ the resulting public keys of signers S_0 and S_1. Also $\mathcal{B}$ runs the *DVS.VKeyGen* phase, and sends $\mathcal{A}$ the resulting a public key of a designated verifier D. It keeps the private keys of S_0, S_1, D to itself.

Queries 1: Signature queries $(b_1, M_1), \ldots, (b_{q_1}, M_{q_1})$ are issued by $\mathcal{A}$. Here each $b_i \in \{0, 1\}$ $(i = 1, \ldots, q_1)$ means the index of either S_0 or S_1. To each query M_j the challenger $\mathcal{B}$ responds by running *DVS.Sign* to generate a signature σ_i of M_i respecting to the private key SK_i of S_i, and sending σ_i to $\mathcal{A}$. These queries may be asked adaptively such that each query M_i may depend on the replies to $M_1, \ldots, M_{i-1}$.

Challenge: The adversary $\mathcal{A}$ submits a plaintext $M_* \in \mathcal{M}$ for $M_* \notin \{M_1, \ldots, M_{q_1}\}$. The challenger selects random bit $b \in_{\mathrm{R}} \{0, 1\}$, sets $\sigma_* = \mathit{DVS.Sign}(\mathtt{params}, SK_b, M_b)$, and sends σ_* to the adversary as its challenge designated signature.

Queries 2: This is a identical to *Queries 1* for $i = q_1 + 1, \ldots, q$, except that $\mathcal{A}$ may not request the signature of M_*

Guess: The adversary submits a guess $b' \in \{0, 1\}$. The adversary wins if $b = b'$.

We define $\mathtt{AdvPSID}_{\mathcal{A}}$ to be the probability that $\mathcal{A}$ wins the above game, taken over the coin tosses made by $\mathcal{B}$ and $\mathcal{A}$:

$$\mathtt{AdvPSID}_{\mathcal{A}} = \mid \Pr[(b = b')] - 1/2 \mid$$

Table 1. Security of DVS schemes

	ZJ [17]	LJ [10]	Our Proposal
Correctness	Yes	Yes	Yes
UF / SUF	–	UF	SUF
Non-Transferability	Yes	Yes	Yes
PSID	No	No	Yes

Definition 1. An adversary $\mathcal{A}$ (q, t, ε)-breaks a designated verifier signature (DVS) scheme if $\mathcal{A}$ runs in a time of at most t, $\mathcal{A}$ makes at most q Queries, and $\texttt{AdvPSID}_{\mathcal{A}}$ is at least ε. A DVS scheme is the (q, t, ε)-(Strong) Privacy of Signature's Identity (PSID), if no adversary (q, t, ε)-breaks it.

Theorem 2. *Suppose that the $(q_0, t_0, \varepsilon_0)$-Decisional Assumption II holds in $\mathbb{G}$. Then the proposed designated verifier signature scheme $\mathcal{DVS}$ satisfies the (q, t, ε)-(Strong) Privacy of Signature's Identity, provided that $q \le q_0$, $t \le t_0 - O(qT)$ and $\varepsilon \ge \varepsilon_0$, where T is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof of this theorem is proposed in Appendix B. An outline of the proof is as follows. Suppose that there exists an adversary, $\mathcal{A}$, who breaks PSID for $\mathcal{DVS}$, and a challenger, $\mathcal{B}$, takes the Assumption II challenge. After $\mathcal{A}$ and $\mathcal{B}$ execute PSID game, $\mathcal{A}$ outputs a bit to indicate which the private key of the signers is used. Then $\mathcal{B}$ replies the Assumption II response with non-negligible probability. This is a contradiction to hold the Assumption II.

If $\sigma' \in \mathbb{G}^2$ is a valid designated verifier signature of S_1, then it is an invalid one of S_0. Therefore it is computationally infeasible, without the knowledge of the private key of the designated verifier or the one of the signers, to determine whether valid or invalid at signatures of signers.

4.5 Delegatability

The Delegatability is that the signer can delegate her signing ability – with respect to a fixed designated verifier – to a third party, without revealing her private key or making it possible for the third party to sign with respect to other designated verifiers. Our scheme is delegatable if the signer sends the third party $\mathcal{A}$ a pair (g_2^x, g_2^y) for $x, y \in \mathbb{Z}_p^*$. Then $\mathcal{A}$ can generate a valid designated signature:

$$\sigma = (\sigma_1, \sigma_2) := \left(g_2^r, g_2^{x+My+1/r}\right)$$

for $r \in_{\mathrm{R}} \mathbb{Z}_p^*$.

4.6 Security Comparison of DVS schemes

Table 1 shows the security of DVS schemes without random oracles (and without pseudo-randomness), by comparing the Correctness, Unforgeability (UF)/Strong

Table 2. Efficiency of DVS schemes

	ZJ [17]	LJ [10]	Our Proposal
Signature lengths	3	3	2
Pairings	2	2	1

Unforgeability (SUF), Non-Transferability, and (Strong) Privacy of Signature's Identity (PSID). Only our scheme satisfies the PSID in this table. Notice that security of the DVS scheme [10] relies on the strange looking *Knowledge-of-Exponent* assumption [7, 10], which is non-black box in that security reductions from this assumption entail some kind of access to the internal state of the adversary. Our scheme does not rely on this assumption.

4.7 Efficiency Comparison of DVS schemes

Table 2 shows the efficiency of DVS schemes without random oracles (and without pseudo-randomness), by comparing signature lengths as unit $|\mathbb{Z}_p| = |\mathbb{G}|$, and by the number of pairings during one iteration of verification. Our scheme is the most efficient within the scope in these schemes.

5 Our Computational problem and a Signature Scheme

The aim of this section is to justify our proposed computational problem. We propose a conventional signature scheme $\mathcal{S}$, which gives the result that the difficulty of the computational problem (the Assumption I) is equivalent to be strongly unforgeable for $\mathcal{S}$. Unfortunately, the scheme is based on the asymmetric pairings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, so a signature in the scheme consists of 2 elements in $\mathbb{G}_2$, amounting then to 4 units as unit $|\mathbb{Z}_p| = |\mathbb{G}_1| = |\mathbb{G}_2|/2$.

5.1 A Conventional Signature Scheme $\mathcal{S}$

We give a conventional signature scheme with three phases: *S.KeyGen, S.Sign*, and *S.Verify*. For the moment we shall assume that the signature message M is an element in $\mathbb{Z}_p$, but the domain can be extended to all of $\{0,1\}^*$ using a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$.

S.KeyGen: Choose multiplicative cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ of sufficiently large prime order p, random generators g_2 of $\mathbb{G}_2$, the one-way isomorphism[4] $f : \mathbb{G}_2 \to \mathbb{G}_1$ with a generator $g_1 := f(g_2) \in \mathbb{G}_1$, the asymmetric pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The public key is generated as follows. Generate a private

[4] Saito–Hoshino–Uchiyama–Kobayashi [12] proposed one-way isomorphisms $\{f\}$ on multiplicative cyclic groups constructed on non-supersingular elliptic curves.

key $SK := (g_2^x, g_2^y) \in \mathbb{G}_2^2$ from secrets $x, y \in_{\mathrm{R}} \mathbb{Z}_p^*$, and calculate $(X, Y) := (f(g_2^x), f(g_2^y))$ $(= (g_1^x, g_1^y) \in \mathbb{G}_1^2)$.

$$\begin{array}{ccccc} (\mathbb{Z}_p^*)^2 & \longrightarrow & \mathbb{G}_2^2 & \xrightarrow{f} & \mathbb{G}_1^2 \\ (x,y) & \longmapsto & SK := (g_2^x, g_2^y) & \longmapsto & (X,Y) := (f(g_2^x), f(g_2^y)) \ \ (= (g_1^x, g_1^y)) \end{array}$$

Assume that $z := \hat{e}(g_1, g_2) \in \mathbb{G}_T$. The public and private keys are:

$$PK := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e}, f, g_1, g_2, X, Y, z) \quad \text{and} \quad SK := (g_2^x, g_2^y) \ ,$$

respectively.

S.Sign: A signature of a message $M \in \mathbb{Z}_p$ is generated as follows. First, a random $r \in \mathbb{Z}_p^*$ is chosen. The signature is then constructed as:

$$\sigma := \left(g_2^r, g_2^{x+My+1/r} \right) \ .$$

S.Verify: Suppose we wish to check if $\sigma = (\sigma_1, \sigma_2)$ is a signature for a message M. Verify

$$\hat{e}\left(X^{-1} \cdot Y^{-M} \cdot f(\sigma_2), \sigma_1 \right) = z \ .$$

If the equality holds the result is `valid`; otherwise the result is `invalid`.

It is easy to see that σ will be accepted by a verifier. Thus the scheme is correct.

5.2 The Computational Problem and Strong Unforgeability

Theorem 3. *Assume that $g_1 = f(g_2) \in \mathbb{G}_1$ for the one-way isomorphism $f: \mathbb{G}_2 \to \mathbb{G}_1$ and the random generator $g_2 \in_{\mathrm{R}} \mathbb{G}_2$. That the conventional signature scheme $\mathcal{S}$ is strongly unforgeable is almost equivalent to that the Computational Assumption I holds in $(\mathbb{G}_1, \mathbb{G}_2)$. More correctly,*

(T3-1) *Suppose that the $(q_0, t_0, \varepsilon_0)$-Computational Assumption I holds in $(\mathbb{G}_1, \mathbb{G}_2)$ with $g_1 = f(g_2)$. Then the signature scheme $\mathcal{S}$ is (q, t, ε)-strongly unforgeable, provided that $q \le q_0$, $t \le t_0 - O(qT)$ and $\varepsilon \ge 2\varepsilon_0$, where T is the maximum time for an exponentiation in $\mathbb{G}_2$.*

(T3-2) *Assume that the signature scheme $\mathcal{S}$ is $(q_1, t_1, \varepsilon_1)$-strongly unforgeable. Then the (q, t, ε)-Computational Assumption I holds in $(\mathbb{G}_1, \mathbb{G}_2)$ with $g_1 = f(g_2)$, provided that $q \le q_1$, $t \le t_1$ and $\varepsilon \ge \varepsilon_1$.*

Proof of (T3-1) in Theorem 3 is proposed in Appendix C, such as that of Theorem 1.

Proof (of (T3-2) in Theorem 3). The challenger generates a public pair (PK, SK), and sends the private key SK to the adversary. The adversary generates messages M_i $(i = 1, \ldots, q)$ which are $M_1 = \cdots = M_q$, and sends $M_1, \ldots, M_q$ to the challenger. For each message M_i $(i = 1, \ldots, q)$, The challenger generates

$$\sigma_i = \left(g_2^{r_i}, g_2^{x+M_i y+1/r_i} \right) \ ,$$

and sends it to the adversary. Then the challenger performs the Assumption I game for input

$$\left(g_1, g_2, g_1^x \cdot (g_1^y)^{M_i}, g_2^{r_i}, g_2^{x+M_i y+1/r_i} \,\middle|\, i = 1, \ldots q \right) \ ,$$

and receives a valid output

$$\sigma_* = \left(g_2^{r_*}, g_2^{x+M_* y+1/r_*} \right)$$

for $r_* \notin \{r_1, \ldots, r_q\}$ and $M_* = M_1 = \cdots = M_q$. This output is valid with a probability greater than ε. The adversary outputs (M_*, σ_*), which is valid in the strongly unforgeable game with a probability greater than ε. This is a contradict to the assumption of the theorem. We have proven (T3-2) of Theorem 3.

6 Conclusions

In this paper, we defined an underlying computational problem and its decisional problem. As an application of their problems, we proposed an efficient DVS scheme without random oracles. We proved our DVS scheme satisfying security based on the difficulty of the problems. Also we proved that the difficulty of the computational problem was tightly reduced to the Strong Unforgeability of our proposed conventional signature scheme related to asymmetric pairings.

References

1. M. Abe, Y. Cui, H. Imai, and E. Kiltz. Efficient hybrid encryption from ID-based encryption. Cryptology ePrint Archive, Report 2007/023, 2007. `http://eprint.iacr.org/`.
2. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO* 2001, *LNCS* 2139, pages 213–229. Springer-Verlag, 2001.
3. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.
4. Q. Huang, D. S. Wong, and Y. Zhao. Generic transformation to strongly unforgeable signatures. In *ACNS* 2007, *LNCS* 4521, pages 1–17. Springer-Verlag, 2007.
5. Q. Huang, G. Yang, D. S. Wong, and W. Susilo. Efficient strong designated verifier signature schemes without random oracles or delegatability. Cryptology ePrint Archive, Report 2009/518, 2009. `http://eprint.iacr.org/`.
6. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT*, pages 143–154, 1996.

7. F. Laguillaumie, B. Libert, and J.-J. Quisquater. Universal designated verifier signatures without random oracles or non-black box assumptions. In *SCN* 2006*, LNCS* 4116, pages 63–77, 2006.
8. F. Laguillaumie and D. Vergnaud. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In *SCN* 2004*, LNCS* 3352, pages 105–119, 2004.
9. Y. Li, W. Susilo, Y. Mu, and D. Pei. Designated verifier signature: Definition, framework and new constructions. In *UIC*, pages 1191–1200, 2007.
10. Y. Liao and C. Jia. Designated verifier signature without random oracles. In *ICCCAS* 2008, pages 474–477. IEEE Computer Society Press, 2008.
11. K. G. Paterson and J. C. N. Schuldt. Efficient identity-based signatures secure in the standard model. In *ACISP* 2006*, LNCS* 4058, pages 207–222. Springer-Verlag, 2006.
12. T. Saito, F. Hoshino, S. Uchiyama, and T. Kobayashi. Candidate one-way functions on non-supersingular elliptic curves. *IEICE Transactions*, 89-A(1):144–150, 2006.
13. C. Sato, T. Okamoto, and E. Okamoto. Sender authenticated key agreements without random oracles. *IEICE Transactions*, 92-A(8):1787–1794, 2009.
14. C. Sato, T. Okamoto, and E. Okamoto. Strongly unforgeable ID-based signatures without random oracles. In *ISPEC* 2009*, LNCS* 5451, pages 35–46. Springer-Verlag, 2009.
15. S. F. Shahandashti and R. Safavi-Naini. Construction of universal designated-verifier signatures and identity-based signatures from standard signatures. In *Public Key Cryptography* 2008 *LNCS* 4939, pages 121–140, 2008.
16. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT* 2005*, LNCS* 3027, pages 114–127. Springer-Verlag, 2005.
17. J. Zhang and C. Ji. An efficient designated verifier signature scheme without random oracles. In *ISDPE '07: Proceedings of The First International Symposium on Data, Privacy, and E-Commerce*, pages 338–340, Washington, DC, USA, 2007. IEEE Computer Society.

A Security Proof of Lemma 1 (SUF for $\mathcal{DVS}$)

Proof (Proof of Lemma 1). Let $\varepsilon_{i,j}$ $(i,j=1,2)$ be the probability in (4.2.2-i,j) that $\mathcal{B}$ can generate a valid output for the Assumption I game after he receive the valid output of the simulator description from $\mathcal{A}$. Let $\varepsilon'_{1,2}$ be the probability that $\mathcal{B}$ can generate g_2^y from only $(M_*, \sigma_{*,2})$ and $(M_j, \sigma_{j,2})$ in (4.2.2-1.2).

Since $\varepsilon'_{1,2} = \varepsilon_{2,1}$ and $\varepsilon_{1,1} + \varepsilon'_{1,2} = 1$, we have

$$\begin{aligned}
&\Pr[\mathcal{B} \text{ breaks } (q_0, t_0, \varepsilon_0)\text{-Assumption I}] \\
&\quad = \frac{1}{2}(\varepsilon_{1,1} + \varepsilon_{1,2} + \varepsilon_{2,1} + \varepsilon_{2,2}) \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{DVS}] \\
&\quad \geq \frac{1}{2}(\varepsilon_{1,1} + \varepsilon'_{1,2}) \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{DVS}] \\
&\quad = \frac{1}{2} \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{DVS}] \ .
\end{aligned}$$

□

B Security Proof of Theorem 2 (PSID for $\mathcal{DVS}$)

Proof. Suppose that there exists an adversary, $\mathcal{A}$, who breaks the (q, t, ε)-(Strong) Privacy of Signature's Identity (PSID) of our DVS scheme $\mathcal{DVS}$:

$$\left| \Pr\left[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-SUF of } \mathcal{DVS}\right] - \frac{1}{2} \right| \geq \varepsilon \ . \tag{7}$$

We construct a simulator, $\mathcal{B}$, to play the Decisional Assumption II game. The simulator $\mathcal{B}$ will take the Decisional Assumption II challenge (2) for $x, r_* \in_{\mathrm{R}} \mathbb{Z}_p^*$ and run $\mathcal{A}$ executing the following steps.

B.1 Simulator Description

Setup: The simulator $\mathcal{B}$ sends

$$\mathtt{params} = (\mathbb{G}, \mathbb{G}_T, p, e, g_1)$$

to the adversary $\mathcal{A}$. $\mathcal{B}$ generates $g_1^y, (g_1^x)^s, g_1^t \in \mathbb{G}$ from $y, s, t \in_{\mathrm{R}} \mathbb{Z}_p^*$. He generates $b \in_{\mathrm{R}} \{0, 1\}$, and sends

$$PK_{S_0} = (g_1^x, g_1^y) \text{ and } PK_{S_1} = (g_1^{xs}, g_1^t)$$

to $\mathcal{A}$. Also $\mathcal{B}$ sends

$$PK_D = (g_2, z)$$

to the adversary $\mathcal{A}$, where $z = e(g_1, g_2)$.

Queries 1: The adversary $\mathcal{A}$ issues $(b_1, M_1), \ldots, (b_{q_1}, M_{q_1})$ where $b_i \in \{0, 1\}$ $(i = 1, \ldots, q_1)$ means the index of either S_0 or S_1. These queries may be asked adaptively such that each query M_i may depend on the replies to $M_1, \ldots, M_{i-1}$. In the case $b_i = 0$, the simulator $\mathcal{B}$ generates the signature for S_0:

$$\sigma_i := \left(g_2^{r_i}, g_2^{x+1/r_i} \cdot g_2^{M_i y}\right) \ ,$$

and sends it to $\mathcal{A}$. In the case $b_i = 1$, $\mathcal{B}$ generates the signature for S_1:

$$\sigma_i := \left((g_2^{r_i})^{1/s}, \left(g_2^{x+1/r_i}\right)^s \cdot g_2^{M_i t}\right) \ ,$$

and sends it to $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ submits a message $M_* \in \mathcal{M}$ for $M_* \notin \{M_1, \ldots, M_{q_1}\}$. The simulator $\mathcal{B}$ selects random bit $b \in_{\mathrm{R}} \{0, 1\}$. In the case $b = 0$, the simulator $\mathcal{B}$ generates the signature for S_0:

$$\sigma_* := \left(g_2^{r_*}, R \cdot g_2^{M_* y}\right) \left(= \left(g_2^{r_*}, g_2^{x+M_* y+1/r_*}\right) \text{ if } R = g_2^{x+1/r_*}\right) \ ,$$

and sends it to $\mathcal{A}$. In the case $b = 1$, $\mathcal{B}$ generates the signature for S_1:

$$\sigma_* := \left((g_2^{r_*})^{1/s}, R^s \cdot g_2^{M_* t} \right) \left(= \left(g_2^{r_*/s}, g_2^{xs+M_* t+s/r_*} \right) \text{ if } R = g_2^{x+1/r_*} \right),$$

and sends it to $\mathcal{A}$.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$ then $\mathcal{B}$ outputs a guess of $\mathcal{R} = 1$ (indicating that $R = g_2^{x+1/r_*}$); otherwise, it outputs $\mathcal{R} = 0$.

B.2 Analysis

If $R = g_2^{x+1/r_*}$, then the simulation is perfect, and $\mathcal{A}$ will guess the bit b correctly with probability $1/2 + \varepsilon'$. Otherwise, R is uniformly random, and thus σ_* is uniformly random and independent, and can not impart no information regarding the bit b. From (7), we have that

$$\begin{aligned} &\Big| \ \Pr\left[\mathcal{A}\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i}, g_2^{r_*}, R \ \middle|\ i = 1, \ldots, q\right) = 1\right] \\ &\quad -\Pr\left[\mathcal{A}\left(g_1, g_2, g_1^x, g_2^{r_i}, g_2^{x+1/r_i}, g_2^{r_*}, g_2^{x+1/r_*} \ \middle|\ i = 1, \ldots, q\right) = 1\right] \Big| \\ &\geq \left| \frac{1}{2} - \Pr\left[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-PSID of } \mathcal{DVS}\right] \right| \geq \varepsilon \end{aligned}$$

for $x, r_i \in_{\mathrm{R}} \mathbb{Z}_p^*, r_* \in \mathbb{Z}_p^*, r_* \notin \{r_1, \ldots, r_q\}$ and $R \in_{\mathrm{R}} \mathbb{G}$, which is a contradiction to the assumption $\varepsilon \geq \varepsilon_0$ in Theorem 2. We have proven the theorem. □

C Security Proof of (T3-1) in Theorem 3 (SUF for $\mathcal{S}$)

An outline of the proof is as follows. Suppose that there exists an adversary, $\mathcal{A}$, who breaks the Strong Unforgeability (SUF) of our signature scheme $\mathcal{S}$ in Section 5.1, and a challenger, $\mathcal{B}$, takes the Assumption I challenge (with $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$). After $\mathcal{A}$ and $\mathcal{B}$ execute the strongly unforgeable game, $\mathcal{A}$ outputs a valid tuple for a message and a signature. Then $\mathcal{B}$ replies the Assumption I response with non-negligible probability. This is a contradiction to hold the Assumption I.

Proof (of (T3-1) in Theorem 3). Suppose that there exists an adversary, $\mathcal{A}$, who breaks (q, t, ε)-SUF of our signature scheme $\mathcal{S}$:

$$\Pr\left[\mathcal{A} \text{ breaks } (q, t, \varepsilon)\text{-SUF of } \mathcal{S}\right] \geq \varepsilon \ . \tag{8}$$

We construct a simulator, $\mathcal{B}$, to play the Assumption I game. The simulator $\mathcal{B}$ will take the Assumption I challenge (1) for $x, r_i \in_{\mathrm{R}} \mathbb{Z}_p^*$ $(i = 1, \ldots, q)$ and run $\mathcal{A}$ executing the following steps.

C.1 Simulator Description

Setup: The simulator $\mathcal{B}$ generates $g_1^y \in \mathbb{G}_1$ from $y \in_{\mathrm{R}} \mathbb{Z}_p^*$. He generates $b \in_{\mathrm{R}} \{0,1\}$, and sends to the adversary $\mathcal{A}$ either

$$\begin{cases} \textbf{(C.1-1)} & (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e}, f, g_1, g_2, g_1^x, g_1^y) \text{ if } b = 0 \text{ , or} \\ \textbf{(C.1-2)} & (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \hat{e}, f, g_1, g_2, g_1^y, g_1^x) \text{ if } b = 1 \text{ .} \end{cases}$$

The adversary $\mathcal{A}$ cannot know whether the received parameter is in the cases (C.1-1) or (C.1-2).

Signature Queries: The adversary $\mathcal{A}$ issues signature queries $M_1, \ldots, M_q$. These queries may be asked adaptively so that each query M_i may depend on the replies to $M_1, \ldots, M_{i-1}$.

In the case (C.1-1), the simulator $\mathcal{B}$ generates the signature

$$(\sigma_{i,1}, \sigma_{i,2}) := \left(g_2^{r_i}, \left(g_2^{x+1/r_i} \right) \cdot (g_2^y)^{M_i} \right) \left(= \left(g_2^{r_i}, g_2^{x+M_i y+1/r_i} \right) \right) ,$$

and sends it to $\mathcal{A}$. In the case (C.1-2) and $M_i \neq 0$, $\mathcal{B}$ generates

$$(\sigma_{i,1}, \sigma_{i,2}) := \left((g_2^{r_i})^{1/M_i}, \left(g_2^{x+1/r_i} \right)^{M_i} \cdot g_2^y \right) \left(= \left(g_2^{r_i/M_i}, g_2^{y+M_i x+M_i/r_i} \right) \right) ,$$

and sends it to $\mathcal{A}$. In the case (C.1-2) and $M_i = 0$, $\mathcal{B}$ generates $(\sigma_{i,1}, \sigma_{i,2}) := \left(g_2^{r'_i}, g_2^{y+1/r'_i} \right)$ for $r'_i \in_{\mathrm{R}} \mathbb{Z}_p^*$, and sends it to $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs (M_*, σ_*) such that $\sigma_* = (\sigma_{*,1}, \sigma_{*,2}) \in \mathbb{G}_2^2$ is a valid signature of M_*, and $(M_*, \sigma_{*,1}, \sigma_{*,2}) \notin \{(M_1, \sigma_{1,1}, \sigma_{1,2}), \ldots, (M_q, \sigma_{q,1}, \sigma_{q,2})\}$.

C.2 Analysis

The signature σ_* is valid. In the case (C.1-1), assume that $(\sigma_{*,1}, \sigma_{*,2}) := \left(g_2^{r_*}, g_2^{x+M_* y+1/r_*} \right)$ for $r_* \in \mathbb{Z}_p^*$ and

$$(M_*, r_*) \neq (M_i, r_i) \tag{9}$$

for $i = 1, \ldots, q$ (i.e. $(M_*, r_*) \notin \{(M_1, r_1), \ldots, (M_q, r_q)\}$). Also, in the case (C.1-2), assume that $(\sigma_{*,1}, \sigma_{*,2}) := \left(g_2^{r_*}, g_2^{y+M_* x+1/r_*} \right)$ for $r_* \in \mathbb{Z}_p^*$ and

$$(M_*, r_*) \not\equiv \begin{cases} (M_i, r_i/M_i) \pmod{p} & \text{for } M_i \neq 0 \text{ ,} \\ (0, r'_i) \pmod{p} & \text{for } M_i = 0 \end{cases}$$

for $i = 1, \ldots, q$.

(C.2-1.1) If $r_* \notin \{r_1, \ldots, r_q\}$ in the case (C.1-1), the simulator $\mathcal{B}$ calculates $\left(g_2^{r_*}, g_2^{x+1/r_*} \right) = \left(\sigma_{*,1}, \sigma_{*,2} \cdot (g_2^y)^{-M_*} \right)$ which is a valid output for the Assumption I game.

(C.2-1.2) Otherwise in the case (C.1-1), assume that $r_* = r_j$ $(1 \leq j \leq q)$. Then we have $M_* \neq M_j$ from (9). Though the simulator $\mathcal{B}$ can calculate $g_2^y =$

$(\sigma_{*,2}/\sigma_{j,2})^{1/(M_*-M_j)}$ from the only $(M_*, \sigma_{*,2})$ and $(M_j, \sigma_{j,2})$, he does not seem to propose a valid output for the Assumption I game. However, notice that $\mathcal{A}$ does not know whether $\mathcal{B}$ simulates (C.2-1.2) or the following (C.2-2.1).

(C.2-2.1) In the case (C.1-2), if there exists an index j $(1 \le j \le q)$ such that

$$r_* \equiv \begin{cases} r_j/M_j & \text{for } M_* \neq M_j \text{ and } M_j \neq 0 \ , \\ r'_j & \text{for } M_* \neq 0 \text{ and } M_j = 0 \ , \end{cases}$$

the simulator $\mathcal{B}$ can calculate g_2^x such as (C.2-1.2). He can generate $\left(g_2^{r'_*}, g_2^{x+1/r'_*}\right)$ for $r'_* \notin \{r_1, \ldots, r_q\}$, which is a valid output for the Assumption I game.

(C.2-2.2) Otherwise in the case (C.1-2), assume that $r_* \not\equiv r_i/M_i$ for $M_i \neq 0$, and assume that $r_* \neq r'_i$ for $M_i = 0$. Since $\mathcal{B}$ cannot obtain new information of solving the Assumption I game from $\mathcal{A}$ at the simulator description in Section C.1, it seems that $\mathcal{B}$ cannot propose a valid output for the game.

Let $\Pr[\mathcal{B}$ breaks $(q_0, t_0, \varepsilon_0)$-Assumption I $]$ be the probability that $\mathcal{B}$ generates a valid output for the Assumption I game.

Lemma 2.
$$\frac{1}{2}\ \mathit{Pr}[\mathcal{A} \textit{ breaks } (q,t,\varepsilon)\textit{-SUF of } \mathcal{S}] \le \mathit{Pr}[\mathcal{B} \textit{ breaks } (q_0,t_0,\varepsilon_0)\textit{-Assumption I}\ ] \ .$$

Proof (Proof of Lemma 2). Let $\varepsilon_{i,j}$ $(i,j = 1,2)$ be the probability in (C.2-i,j) that $\mathcal{B}$ can generate a valid output for the Assumption I game after he receive the valid output of the simulator description from $\mathcal{A}$. Let $\varepsilon'_{1,2}$ be the probability that $\mathcal{B}$ can generate g_2^y from only $(M_*, \sigma_{*,2})$ and $(M_j, \sigma_{j,2})$ in (C.2-1.2).

Since $\varepsilon'_{1,2} = \varepsilon_{2,1}$ and $\varepsilon_{1,1} + \varepsilon'_{1,2} = 1$, we have

$$\begin{aligned} &\Pr[\mathcal{B} \text{ breaks } (q_0,t_0,\varepsilon_0)\text{-Assumption I}\ ] \\ &\quad = \frac{1}{2}(\varepsilon_{1,1} + \varepsilon_{1,2} + \varepsilon_{2,1} + \varepsilon_{2,2})\ \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{S}] \\ &\quad \ge \frac{1}{2}(\varepsilon_{1,1} + \varepsilon'_{1,2})\ \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{S}] \\ &\quad = \frac{1}{2}\ \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{S}] \ . \end{aligned}$$

□

From the assumption in Theorem 3,

$$\Pr[\mathcal{B} \text{ breaks } (q_0,t_0,\varepsilon_0)\text{-Assumption I}\ ] < \varepsilon_0 \ . \tag{10}$$

From (8), (10) and Lemma 2, we have

$$\begin{aligned} \frac{\varepsilon}{2} &\le \frac{1}{2}\ \Pr[\mathcal{A} \text{ breaks } (q,t,\varepsilon)\text{-SUF of } \mathcal{S}] \\ &\le \Pr[\mathcal{B} \text{ breaks } (q_0,t_0,\varepsilon_0)\text{-Assumption I}\ ] < \varepsilon_0 \ , \end{aligned}$$

which is a contradiction to the assumption $\varepsilon/2 \ge \varepsilon_0$ in Theorem 3.

Therefore, we have $\Pr[\mathcal{A}$ breaks (q,t,ε)-SUF of $\mathcal{S}] < \varepsilon$. □