# Universal Designated Verifier Signature Without Delegatability

Xinyi Huang, Willy Susilo, Yi Mu, and Wei Wu

Center for Information Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, Australia
{xh068, wsusilo, ymu}@uow.edu.au

**Abstract.** In Asiacrypt 2003, the notion of the universal designated verifier signature (UDVS) was put forth by Steinfeld, Bull, Wang and Pieprzyk. In the new paradigm, *any* signature holder (not necessarily the signer) can designate the standard signature to any desired designated verifier (using the verifier's public key), such that *only* the designated verifier will believe that the signature holder holds a valid standard signature, and hence, believe that the signer has indeed signed the message. When the signature holder is the *signer* himself, the UDVS scheme can be considered as a designated verifier signature (DVS) which was proposed by Jakobsson, Sako and Impagliazzo in Eurocrypt 1996. In the recent paper published in ICALP 2005, Lipmaa, Wang and Bao introduced a new security property, called "non-delegatability", as an essential property of (universal) designated verifier signature. Subsequently, Li, Lipmaa and Pei used this new property to "attack" four designated verifier signatures in ICICS 2005 and showed that *none* of them satisfy the required property. To date, there is no UDVS scheme that does not suffer from the delegatability problem. In this paper, we propose the *first* provably secure UDVS without delegatability, which can also be regarded as another DVS scheme *without* delegatability. We also refine the models of the UDVS schemes and introduce the notion of the strong universal designated verifier signature (SUDVS). We believe that the model itself is of an independent interest.

**Keywords:** Universal Designated Verifier Signatures, Designated Verifier Signatures, Non-delegatability, Bilinear Pairings.

## 1 Introduction

Digital signatures, introduced in the pioneering paper of Diffie and Hellman [3], allow a signer with a secret key to sign messages such that anyone with access to the corresponding public key can verify the authenticity of the message. A signature verifier can convince any third party about this fact by presenting a digital signature on a message. The ease of copying and transmitting digital signatures in some implementations is of great convenience, but it is unsuitable for many other applications in the real world where a verifier does not want to

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 479–498, 2006.

present the publicly verifiable signatures to other parties, such as certificates for hospital records, income summary, etc.

The notion of the *designated verifier signature* (DVS) was proposed by Jakobsson, Sako and Impagliazzo in [4]. In a DVS, the signature provides authentication of a message *without* providing a non-repudiation property of traditional signatures. A DVS can be used to convince a single third party, i.e. the designated verifier, and *only* the designated verifier who can be convinced about its validity or invalidity. This is due to the fact that the designated verifier can always create a signature intended for himself that is indistinguishable from an original signature. In the same paper, Jakobsson, Sako and Impagliazzo also introduced a stronger version of designated verifier signatures called *strong designated verifier signatures* (SDVS). In this concept, *no third party* can even verify the designated verifier signature as the designated verifier's secret key is *required* during the verification phase. Saeednia, Kremer and Markowitch firstly formalized the notion of strong DVS [15] and proposed an efficient scheme in the same paper. Some other recent papers discussing both DVS and SDVS include [5,6,8,9,10].

Universal designated verifier signature, which was introduced by Steinfeld, Bull, Wang and Pieprzyk [16] in Asiacrypt 2003, is a variant of DVS, in the sense that, given a standard signature from the signer, a signature holder (not necessarily the signer) can convert it to a UDVS which is designated to a verifier, such that only this designated verifier can believe that the message has been signed by the signer. However, any other third party cannot believe it since this verifier can use his secret key to create a valid UDVS which is designated to himself. Thus, one cannot distinguish whether a UDVS is created by the signature holder or the designated verifier himself. When the signature holder and the signer are the same user, a universal designated signature will form a designated verifier signature. Therefore, UDVS can be viewed as an application of general designated verifier signatures where the signer designates a non-interactive proof statement to a designated verifier.

From BLS short signature[2], Steinfeld, Bull, Wang and Pieprzyk [16] proposed the first UDVS scheme in Asiacrypt 2003. Steinfeld, Wang and Pieprzyk continued to show how to obtain a UDVS scheme from the Schnorr/RSA signature scheme in PKC 2004 [17]. Zhang, Susilo, Mu and Chen [21] extended this notion to the Identity-based setting and proposed two Identity-based UDVS schemes. The first UDVS scheme without random oracle was proposed by Zhang, Furukawa and Imai [20] in ACNS 2005, where a variant of BB's [1] short signature scheme without random oracle is used as the building block. Very recently, Vergnaud proposed two extensions of pairing-based signatures into universal designated verifier signatures [19].

Recently, Lipmaa, Wang and Bao introduced a new security notion for DVS schemes called *non-delegatability* [10]. They argued that this notion is necessary in many applications such as hypothetical e-voting protocol provided in [10]. They also showed that Saeednia-Kremer-Markowitch's scheme [15], Steinfeld-Bull-Wang-Pieprzyk's scheme [16] and Steinfeld-Wang-Pieprzyk's [17] are delegatable. In ICICS 2005, Li, Lipmaa and Pei presented an "attack" to another

four schemes, namely Susilo-Zhang-Mu's scheme [18], Ng-Susilo-Mu's scheme [11], Laguillaumie-Vergnaud's scheme [6] and Zhang-Furukawa-Imai's scheme [20], and show that they are delegatable [9]. Together with the analysis in [10] and [9], there are only two known provably secure DVS schemes without delegatability: one is the scheme proposed in [4] and the other one is in [10]. Nonetheless, there is *no* provably secure UDVS without delegatability. Therefore, whether the delegatability is an inherent problem of UDVS is an open research problem.

*Our Contribution*
In this paper, we firstly show that the two UDVS schemes which are very recently proposed by Vergnaud in ICALP 2006 [19] are delegatable. Then we refine the definitions of the UDVS and introduce the notion of the strong universal designated verifier signature (SUDVS). We proceed by proposing the *first* construction of non-delegatable UDVS scheme with formal security analysis in the random oracle model.

*Organization of The Paper*
In the next section, we will review some preliminaries required throughout the paper. In Section 3, we review the definition of the delegatability by analyzing two UDVS schemes which are very recently proposed in ICALP 2006. We provide the security models of UDVS in Section 4. In Section 5, we propose the *first* construction of UDVS without delegatabiity together with its security analysis. Finally, Section 6 concludes this paper.

## 2 Preliminaries

In this section, we will review some fundamental backgrounds used throughout this paper, namely bilinear pairing, complexity assumptions and the formal models of the universal designated verifier signature.

### 2.1 Bilinear Pairing

Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two groups of prime order $p$ and let $g$ be a generator of $\mathbb{G}_1$. The map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following three conditions hold true:

- $e$ is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- $e$ is non-degenerate, i.e. $e(g, g) \neq 1_{\mathbb{G}_T}$.
- $e$ is efficiently computable.

We say that $(\mathbb{G}_1, \mathbb{G}_T)$ are bilinear groups if there exists the bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ as above, and $e$, and the group action in $\mathbb{G}_1$ and $\mathbb{G}_T$ can be computed efficiently. See [2] for more details on the construction of such pairings.

### 2.2 Complexity Assumptions

**Definition 1. Computational Diffie Hellman(CDH) Problem in** $\mathbb{G}_1$
*Given* $g, g^a, g^b \in \mathbb{G}_1$ *for some unknown* $a, b \in \mathbb{Z}_p$*, compute* $g^{ab} \in \mathbb{G}_1$.

The probability that a polynomially bounded algorithm $\mathcal{A}$ can solve the CDH problem is defined as:

$$Succ^{CDH}_{\mathcal{A},\mathbb{G}_1} = \Pr[g^{ab} \leftarrow \mathcal{A}(\mathbb{G}_1, g, g^a, g^b)].$$

**Definition 2. Computational Diffie-Hellman(CDH) Assumption in $\mathbb{G}_1$** *Given $g, g^a, g^b \in \mathbb{G}_1$ for some unknown $a, b \in \mathbb{Z}_p$, $Succ^{CDH}_{\mathcal{A},\mathbb{G}_1}$ is negligible.*

### 2.3 Formal Model of Universal Designated Verifier Signature

There are three parties in the universal designated verifier signature: the Signer $S$, the Signature Holder $SH$ and the Verifier $V$ where

1. $S$ is the one who uses his/her secret key to generate a standard signature $\sigma_{\mathsf{SS}}$ on the message $m$.
2. $SH$ is the one who owns $S$'s standard signature $\sigma_{\mathsf{SS}}$ on the message $m$ and will generate a universal designated verifier signature $\sigma_{\mathsf{DV}}$ to convince $V$ that $S$ has signed the message $m$ and he owns $\sigma_{\mathsf{SS}}$.
3. $V$ is the designated verifier of the signature $\sigma_{\mathsf{DV}}$ and is convinced that $S$ has signed the message $m$. However, $V$ cannot convince anyone else that $S$ has signed the message $m$, even $V$ sharing his secret key with the one who wants to be convinced.

The universal designated verifier signature scheme UDVS consists of the following algorithms: ($\mathsf{CPG}$, $\mathsf{KG}$, $\mathsf{SS}$, $\mathsf{SV}$,$\mathsf{DS}$, $\overline{\mathsf{DS}}$, $\mathsf{DV}$)

1. Common Parameter Generation $\mathsf{CPG}$: a probabilistic algorithm, on input a security parameter $k$, outputs a string $cp \leftarrow \mathsf{CPG}(k)$ which denotes the common scheme parameters.
2. Key Generation $\mathsf{KG}$: a probabilistic algorithm, on input a common parameter $cp$, outputs a secret/public key-pair $(sk, pk) \leftarrow \mathsf{KG}(cp)$ for the signer $S$ and verifier $V$, respectively.
3. Standard Signing $\mathsf{SS}$: a probabilistic (deterministic) algorithm, on input the common parameter $cp$, $S$'s secret key $sk_s$ and the message $m$, outputs $S$'s standard signature $\sigma_{\mathsf{SS}} \leftarrow \mathsf{SS}(cp, sk_s, m)$.
4. Standard Verification $\mathsf{SV}$: a deterministic algorithm, on input the common parameter $cp$, $S$'s public key $pk_s$, the signed message $m$ and $S$'s standard signature $\sigma_{\mathsf{SS}}$, outputs verification decision $d \in \{Acc, Rej\}$ where $\{Acc, Rej\} \leftarrow \mathsf{SV}(cp, pk_s, m, \sigma_{\mathsf{SS}})$.
5. Designation by Signature Holder $\mathsf{DS}$: a probabilistic (deterministic) algorithm, on input the common parameters $cp$, $S$'s public key $pk_s$, $V$'s public key $pk_v$, $S$'s standard signature $\sigma_{\mathsf{SS}}$ of the message $m$, outputs the designated verifier ($\mathsf{DV}$) signature $\sigma_{\mathsf{DV}} \leftarrow \mathsf{DS}(cp, pk_s, pk_v, \sigma_{\mathsf{SS}}, m)$.
6. Simulation by Verifier $\overline{\mathsf{DS}}$: a probabilistic (deterministic) algorithm, on input the common parameter $cp$, $S$'s public key $pk_s$, $V$'s secret key $sk_v$ and the message $m$, outputs the designated verifier($\mathsf{DV}$) signature $\overline{\sigma_{\mathsf{DV}}} \leftarrow \overline{\mathsf{DS}}(cp, pk_s, sk_v, m)$ which is designated to himself.

7. Designation Verification $\mathsf{DV}$: a deterministic algorithm, on input the common parameter $cp$, $S$'s public key $pk_s$, $V$'s secret/public key pair $(sk_v, pk_v)$, the signed message $m$ and the $\mathsf{DV}$ signature $\sigma_{\mathsf{DV}}$, outputs the verification decision $d \in \{Acc, Rej\}$ where $\{Acc, Rej\} \leftarrow \mathsf{DV}(cp, pk_s, sk_v, pk_v, m, \sigma_{\mathsf{DV}})$.

*Consistency:*
In addition to the above algorithms, we also require three obvious consistency of the **UDVS** schemes.

1. $\mathsf{SV}$ Consistency: this property requires that the standard signature produced by the $\mathsf{SS}$ algorithm is accepted as a valid signature by the $\mathsf{SV}$ algorithm, i.e. $\Pr[\mathsf{SV}(cp, pk_s, m, \mathsf{SS}(cp, sk_s, m)) = Acc] = 1$
2. $\mathsf{DV}$ Consistency of $\mathsf{DS}$: this property requires that the $\mathsf{DV}$ signature produced by the $\mathsf{DS}$ algorithm is accepted as a valid signature by the $\mathsf{DV}$ algorithm, i.e.

$$\Pr[\mathsf{DV}(cp, pk_s, sk_v, pk_v, m, \mathsf{DS}(cp, pk_s, pk_v, \sigma_{\mathsf{SS}}, m)) = Acc] = 1.$$

3. $\mathsf{DV}$ Consistency of $\overline{\mathsf{DS}}$: this property requires that the $\mathsf{DV}$ signature produced by the $\overline{\mathsf{DS}}$ algorithm is accepted as a valid signature by the $\mathsf{DV}$ algorithm, i.e.

$$\Pr[\mathsf{DV}(cp, pk_s, sk_v, pk_v, m, \overline{\mathsf{DS}}(cp, pk_s, sk_v, m)) = Acc] = 1.$$

## 3 Delegatability of Universal Designated Verifier Signature Schemes

Let $(sk_s, pk_s), (sk_v, pk_v)$ denote the secret/public key pairs of the signer and the designated verifier, respectively. Delegatability of a UDVS [10] refers the case where the signer *delegates* the UDVS signing rights to $\mathcal{A}$ by disclosing some side information $y_{sv} = f_s(sk_s, pk_v)$ that will help $\mathcal{A}$ to generate valid signatures. Analogously, the designated verifier might delegate this signing rights by simulating capability by disclosing some side information $y_{sv'} = f_v(sk_v, pk_s)$. The implication of the delegatability of UDVS schemes will confuse the designated verifier, when he/she sees a valid universal designated verifier signature that is not generated by himself/herself, then he/she can only conclude that the signature is generated by someone who knows either $y_{sv}$ or $y_{sv'}$. To explain the delegatability more clearly, we analyze the following two UDVS schemes which are recently proposed by Vergnaud [19] in ICALP 2006. For the delegatability of other UDVS schemes [11,16,17,20], please refer to [9,10].

### 3.1 Vergnaud's UDVS-BB [19]

In [19], the author proposed two UDVS schemes which are designed for the devices with constrained computation capabilities since the $\mathsf{SS}$ and $\mathsf{DS}$ algorithms are pairing-free. The first UDVS scheme in [19] combines the BB short signature scheme without random oracle [1] to obtain a UDVS scheme without random oracle. The UDVS-BB [19] consists of the following algorithms. (We rewrite their

scheme with different notations in order to keep the consistence of the whole paper)

- CPG: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be a bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, $k$ be the system security number and $g$ be the generator of $\mathbb{G}_1$. $e$ denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. The message space $D_{\mathcal{M}} = \mathbb{Z}_p^*$. The system parameter $cp = \{\mathbb{G}_1, \mathbb{G}_T, p, k, e, g, D_{\mathcal{M}}\}$ which is shared by all the users in the system.
- KG: The Signer $S$ picks two secret numbers $u_a, v_a \in_R \mathbb{Z}_p^*$ and sets the secret key $sk_s = (u_a, v_a)$. Then $S$ computes the public key $pk_s = (U_a, V_a) = (g^{u_a}, g^{v_a})$. Similarly, the verifier $V$'s secret/public key-pair is $sk_v = (u_b, v_b), pk_v = (U_b, V_b) = (g^{u_b}, g^{v_b})$ where $u_b, v_b$ are randomly chosen in $\mathbb{Z}_q^*$.
- SS: For a message $m \in D_{\mathcal{M}}$ to be signed, $S$ chooses $r \in \mathbb{Z}_p^*$ and computes the standard signature $\sigma_{\mathsf{SS}} = (\sigma_{\mathsf{SS}_1}, \sigma_{\mathsf{SS}_2}) = (r, g^{\frac{1}{u_a+m+v_a r}})$.
- SV: Given a message $m$, the standard signature $\sigma_{\mathsf{SS}} = (\sigma_{\mathsf{SS}_1}, \sigma_{\mathsf{SS}_2})$ and $S$'s public key $pk_s$, one can check whether $e(\sigma_{\mathsf{SS}_2}, U_a \cdot g^m \cdot V_a^{\sigma_{\mathsf{SS}_1}}) \stackrel{?}{=} e(g, g)$. If the equality holds, outputs $Acc$, otherwise, $Rej$.
- DS: Given the standard signature signature $\sigma_{\mathsf{SS}} = (\sigma_{\mathsf{SS}_1}, \sigma_{\mathsf{SS}_2}) = (r, g^{\frac{1}{u_a+m+v_a r}})$ and the verifier's public key $pk_v$, the signature holder $SH$ selects $t \in_R \mathbb{Z}_p^*$ and computes $Q_1 = g^{\frac{t}{u_a+m+v_a r}}$, $Q_2 = (U_b)^t$ and $Q_3 = g^t$. The signature holder sends the universal designated verifier signature $\sigma_{DV} = (r, Q_1, Q_2, Q_3)$ to the verifier $V$.
- $\overline{\mathsf{DS}}$: Given the signer's public key $pk_s$ and the message $m$, the verifier $V$ chooses $t, r \in_R \mathbb{Z}_p^*$ and computes $R = (U_a \cdot g^m \cdot {V_a}^r)^t$. The universal designated verifier signature generated by the verifier is $\overline{\sigma_{DV}} = (r, Q_1, Q_2, Q_3)$ where $Q_1 = g^t$, $Q_2 = R^{u_b}$ and $Q_3 = R$.
- DV: Given the designated verifier signature $(r, Q_1, Q_2, Q_3)$, the verifier checks whether $e(Q_1, U_a \cdot g^m \cdot {V_a}^r) \stackrel{?}{=} e(Q_3, g)$ and $e(Q_3, g^{u_b}) \stackrel{?}{=} e(Q_2, g)$. If both equalities hold, output $Acc$, otherwise, $Rej$.

**Delegatability:**
We will show that the knowledge of $y_{sv} := (g^{u_b u_a}, g^{u_b v_a})$ is sufficient to generate a valid signature of Vergnaud's UDVS-BB scheme. Given a message $m$ and $y_{sv}$, anyone can choose $t, r \in_R \mathbb{Z}_q^*$ and compute $R = (U_a \cdot g^m \cdot V_a^r)^t$. Then he computes $Q_1 = g^t$, $Q_2 = (g^{u_b u_a} \cdot U_b^m \cdot (g^{u_b v_a})^r)^t$ and $Q_3 = R$. Note that $(r, Q_1, Q_2, Q_3)$ is a valid signature of Vergnaud's UDVS-BB since

$$\begin{aligned} & e(Q_1, U_a \cdot g^m \cdot {V_a}^r) = e(g^t, U_a \cdot g^m \cdot {V_a}^r) \\ = & e(g^{t(u_a+m+rv_a)}, g) = e((U_a \cdot g^m \cdot V_a^r)^t, g) \\ = & e(R, g) = e(Q_3, g). \end{aligned}$$

and

$$\begin{aligned} & e(Q_3, g^{u_b}) = e(R, g^{u_b}) = e(R^{u_b}, g) \\ = & e((U_a \cdot g^m \cdot V_a^r)^{tu_b}, g) \\ = & e((g^{u_a u_b} \cdot U_b^m \cdot (g^{u_b v_a})^r)^t, g) = e(Q_2, g). \end{aligned}$$

Note that both the signer and the verifier can compute $y_{sv}$. The signer can use his secret key $u_a, v_a$ to compute $y_{sv} = ((U_b)^{u_a}, U_b^{v_a})$ where $U_b$ is a part of the verifier's public key $pk_v = (U_b, V_b)$. Similarly, the verifier also can use his secret key $(u_b, v_b)$ to compute $y_{sv} = (U_a^{u_b}, V_a^{u_b})$ where $(U_a, V_a)$ is the public key of the signer. Therefore, a valid message signature pair $(m, \sigma_{\mathsf{DV}})$ of UDVS-BB can not convince the verifier that $S$ has signed this message.

### 3.2 Vergnaud's UDVS-BLS [19]

The seconde UDVS scheme UDVS-BLS in [19] combines the BLS short signature [2] to obtain a UDVS with shorter signature length compared with UDVS-BB. It consists of the following algorithms. (We rewrite their scheme with different notations in order to keep the consistence of the whole paper)

- CPG: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be a bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, $k$ be the system security number and $g$ be the generator of $\mathbb{G}_1$. $e$ denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Let $h : \{0,1\}^* \to \mathbb{G}_1^*$ be a secure cryptographic hash function. The message space $D_{\mathcal{M}} = \{0,1\}^*$. The system parameter $cp = \{\mathbb{G}_1, \mathbb{G}_T, p, k, e, h, D_{\mathcal{M}}\}$ which is shared by all the users in the system.
- KG: The Signer $S$ picks a secret value $x_s \in_R \mathbb{Z}_p^*$ and sets the secret key $sk_s = x_s$. Then, $S$ computes the public key $pk_s = g^{x_s}$. Similarly, the verifier $V$'s secret/public key-pair is $(sk_v, pk_v) = (x_v, g^{x_v})$ where $x_v$ is randomly chosen in $\mathbb{Z}_p^*$.
- SS: For a message $m$ to be signed, $S$ computes the standard signature $\sigma_{\mathsf{SS}} = h(m)^{sk_s} \in \mathbb{G}_1$.
- SV: Given a message $m$, the standard signature $\sigma_{\mathsf{SS}}$ and $S$'s public key $pk_s$, one can check the equation $e(\sigma_{\mathsf{SS}}, g) \stackrel{?}{=} e(h(m), pk_s)$. If the equality holds, outputs $Acc$, otherwise, $Rej$.
- DS: Given the standard signature signature $\sigma_{\mathsf{SS}}$ and the verifier's public key $pk_v$, the signature holder $SH$ selects $t \in_R \mathbb{Z}_p^*$ and computes $Q_1 = \sigma_{\mathsf{SS}}^t$ and $Q_2 = pk_v^{t^{-1}}$. Then $SH$ sends the universal designated verifier signature $\sigma_{DV} = (Q_1, Q_2)$ to the verifier $V$.
- $\overline{\mathsf{DS}}$: Given the signer's public key $pk_s$ and the message $m$, the verifier $V$ chooses $t \in_R \mathbb{Z}_p^*$ and computes $Q_1 = h(m)^{t^{-1}}$ and $Q_2 = (pk_s^{sk_v})^t$ The universal designated verifier signature generated by the verifier is $\overline{\sigma_{DV}} = (Q_1, Q_2)$.
- DV: Given the designated verifier signature $(Q_1, Q_2)$, the verifier checks whether $e(Q_1, Q_2) \stackrel{?}{=} e(h(m), pk_s^{sk_v})$. If the equality holds, outputs $Acc$, otherwise, $Rej$.

**Delegatability:**
We will show that the knowledge of $y_{sv} := g^{sk_s sk_v}$ is sufficient to generate a valid signature of Vergnaud's UDVS-BLS. Given a message $m$ and $y_{sv}$, anyone can choose $t \in_R \mathbb{Z}_p^*$ and compute $Q_1 = h(m)^t, Q_2 = y_{sv}^{t^{-1}}$. Note that $(Q_1, Q_2)$ is a valid signature of Vergnaud's UDVS-BLS since $e(Q_1, Q_2) = e(h(m)^t, y_{sv}^{t^{-1}}) = e(h(m), g^{sk_s sk_v}) = e(h(m), pk_s^{sk_v})$.

Note that both the signer and the verifier can compute $y_{sv}$. The signer can use his secret key $sk_s$ to compute $y_{sv} = pk_v^{sk_s}$. Similarly, the verifier also can use his secret key $sk_v$ to compute $y_{sv} = pk_s^{sk_v}$. Therefore, a valid message signature pair $(m, \sigma_{\mathsf{DV}})$ of UDVS-BLS can not convince the verifier that the signature holder holds the signer $S$'s signature on this message.

## 4 Security Models of Universal Designated Verifier Signature

In this section, we will define the security models of our UDVS scheme. Compared with the known security models of UDVS defined in [16,17,20], an important refinement of our model is that we allow the adversaries to adaptively corrupt the users in the system and adaptively choose the target signer and the designated verifier. In the defined models, we allow adversaries to adaptively submit Key Register (KR) queries to register the users in the system and obtain all the public keys he has registered. He can also submit the SS queries to obtain the standard signature of the message under the signer he chooses. In addition, the adversary can choose the message $m$, the signer $S$ and the verifier $V$ and submit $(m, S, V)$ as DS or $\overline{\mathsf{DS}}$ query to obtain the designated verifier signature. If necessary, the adversary can also submit DV queries to decide whether $\sigma_{\mathsf{DV}}$ is a valid designated verifier signature under the signer $S$ and the verifier $V$[1]. We also allow the adversary to submit SecretKey (SK) queries adaptively to obtain the secret keys of some users, thus the adversaries can corrupt some users and adaptively choose the target signer and designated verifier, which reflects more essence of real world adversaries.

**Unforgeability**

Actually, there are two types of unforgeability properties that can be used [16]. The first property, *standard signature unforgeability* (SS-Unforgeability), is just the usual existential unforgeability notion under chosen message attacker [7] for the standard signature scheme SS, which states that no one should be able to forge a standard signature of the signer $S$. The second property, *designated verifier signature unforgeability* (DV-Unforgeability), requires that it is difficult for an attacker to forge a DV signature $\sigma_{DV}^*$ on a new message $m^*$, such that the pair $(M^*, \sigma_{DV}^*)$ passes the DV algorithm with respect to a signer's public key $pk_s^*$ and a designated verifier's public key $pk_v^*$, which states that for any message, an adversary without the standard signature should not be able to convince a designated verifier of holding such a standard signature. DV-Unforgeability always implies the SS-Unforgeability [16]. Thus, it is enough to consider only DV-Unforgeability. The existential unforgeability of UDVS is defined via the following game between the simulator $\mathcal{S}$ and the adaptively chosen message and chosen public key adversary $\mathcal{F}_{EUF,\ UDVS}^{CMA,\ CPKA}$:

[1] Such queries are only needed when the execution of DV algorithm needs the secret key of the verifier. Otherwise, $\mathcal{F}$ can use the public keys of signer $S$ and the verifier $V$ to verify whether a $\sigma_{\mathsf{DV}}$ is valid.

- Setup: The simulator $\mathcal{S}$ runs the CPG to generate the common parameter $cp$. He then returns $cp$ to $\mathcal{F}$.
- Key Register (KR) queries: $\mathcal{F}$ can register for the users in the system. In response. $\mathcal{S}$ runs the KG algorithm to generate the secret/public key pair for this user. $\mathcal{S}$ returns the public key to $\mathcal{F}$.
- SS queries: $\mathcal{F}$ can ask the standard signature of the message $m$ under the public key $pk_s$ he chooses. In response, $\mathcal{S}$ runs the SS algorithm to generate the signature $\sigma_{\mathsf{SS}}$ and returns to $\mathcal{F}$ as the answer.
- DS queries: $\mathcal{F}$ can ask the universal designated verifier signature $\sigma_{\mathsf{DV}}$ which is generated by the algorithm DS on the message $m$ under the public keys $(pk_s, pk_v)$, where $pk_s$ denotes the signer and $pk_v$ denotes the verifier chosen by $\mathcal{F}$. In response, $\mathcal{S}$ firstly runs SS to generate the standard signature $\sigma_{\mathsf{SS}}$ on this message. Then $\mathcal{S}$ runs DS algorithm to generate the universal designated verifier signature $\sigma_{\mathsf{DV}}$. $\mathcal{S}$ returns $\sigma_{\mathsf{DV}}$ to $\mathcal{F}$ as the answer.
- $\overline{\mathsf{DS}}$ queries: $\mathcal{F}$ can ask the designated verifier signature $\overline{\sigma_{\mathsf{DV}}}$ which is generated by the algorithm $\overline{\mathsf{DS}}$ on the message $m$ and under the public keys $(pk_s, pk_v)$, where $pk_s$ denotes the signer and $pk_v$ denotes the verifier chosen by $\mathcal{F}$. In response, $\mathcal{S}$ runs $\overline{\mathsf{DS}}$ algorithm to obtain the designated verifier signature $\overline{\sigma_{\mathsf{DV}}}$. $\mathcal{S}$ then returns $\overline{\sigma_{\mathsf{DV}}}$ to $\mathcal{F}$ as the answer.
- DV queries: $\mathcal{F}$ can ask whether $\sigma_{\mathsf{DV}}$ is a valid universal designated verifier signature on the message $m$ under the public keys $(pk_s, pk_v)$, where $pk_s$ denotes the signer and $pk_v$ denotes the verifier chosen by $\mathcal{F}$. In response, $\mathcal{S}$ will run DV algorithm and return the decision $d \in \{Acc, Rej\}$ to $\mathcal{F}$.
- SK queries: $\mathcal{F}$ can request the secret key of the public key $pk$. In response, $\mathcal{S}$ returns corresponding secret key $sk$ to $\mathcal{F}$.

We say $\mathcal{F}$ wins the game if $\mathcal{F}$ outputs a forged messgae/signature pair$(m^*, \sigma^*_{\mathsf{DV}})$ under the public keys $(pk^*_s, pk^*_v)$ if:

1. $Acc \leftarrow \mathsf{DV}(cp, pk^*_s, sk^*_v, pk^*_v, m^*, \sigma^*_{\mathsf{DV}})$.
2. $(m^*, pk^*_s)$ has never been submitted as one of the SS queries.
3. $(m^*, pk^*_s, pk^*_v)$ has never been submitted as one of the DS or $\overline{\mathsf{DS}}$ queries.
4. Neither $pk^*_s$ nor $pk^*_v$ has been submitted as one of the SK queries.

The success probability of an adaptively chosen message and chosen public key attacker $\mathcal{F}$ wins the above game is defined as $\mathsf{Succ}\ \mathcal{F}^{CMA,\ CPKA}_{EUF,\ UDVS}$.

**Definition 3.** *We say* $\mathcal{F}^{CMA,\ CPKA}_{EUF,\ UDVS}$ *can* $(t, q_H, q_{\mathsf{KR}}, q_{SS}, q_{DS}, q_{\overline{DS}}, q_{DV}, q_{\mathsf{SK}}, \varepsilon)$*-break the* **UDVS** *scheme if* $\mathcal{F}^{CMA,\ CPKA}_{EUF,\ DVS}$ *runs in time at most* $t$*, makes at most* $q_H$ *queries to the random oracle,* $q_{\mathsf{KR}}$ *key registration queries,* $q_{SS}$ *SS queries,* $q_{DS}$ *DS queries,* $q_{\overline{DS}}$ $\overline{DS}$ *queries,* $q_{DV}$ *DV queries,* $q_{\mathsf{SK}}$ SK *queries and* $\mathsf{Succ}\ \mathcal{F}^{CMA,\ CPKA}_{EUF,\ UDVS}$ *is at least* $\varepsilon$*.*

**Non-delegatability**

A universal designated verifier signature can be regarded as a kind of non-interactive system of proofs of knowledge of the signer $S$'s standard signature

$\sigma_{\mathsf{SS}}$ or the verifier's secret key $sk_v$. Thus, both the signature holder and the designated verifier can generate this proof. However, as pointed out by Lipmaa, Wang and Bao in [10], the definition of the unforgeability does not cover the case when the signer (the verifier) is dishonest. Namely, without disclosing $sk_s(sk_v)$, the signer(verifier) can delegate the signing rights to some party $\mathcal{A}$ by disclosing some "side information" which helps the latter to produce valid universal designated verifier signatures on any message, as we have shown in Section 3. One can see a lot of concrete instances in [9,10].

**Definition 4.** *Let $(sk_s, pk_s), (sk_v, pk_v)$ be the secret/public key-pair of signer $S$ and verifier $V$. Let $\mathcal{A}$ be an algorithm, who does not necessarily know the signer's $\mathsf{SS}$ signature $\sigma_{\mathsf{SS}}$ of the message $m$ or the secret key $sk_v$, can produce a valid UDVS on the message $m$ with non-negligible probability $\varepsilon$, we say that a* **UDVS** *scheme is $(\tau, \kappa)$ non-delegatable if in time $\tau$, there exists a knowledge extractor $\mathcal{K}$ who can use $\mathcal{A}$ to obtain $\sigma_{\mathsf{SS}}$ or $sk_v$ with probability greater than $\kappa$.*

**Non-transferability**

Roughly speaking, the Non-Transferability of UDVS requires that: (1) Only the designated verifier can be convinced by the UDVS, even if he shares all the secret information with entities that want get convinced. (2) Even an entity can see many universal designated verifier signatures $\sigma_{\mathsf{DV}}$'s on the same message $m$ but with different designated verifiers, which is generated by the signature holder using the same standard signature $\sigma_{\mathsf{SS}}$, he can not be convinced that the signer has signed on this message. In other words, universal designated verifier signatures of the message $m$ with different designated verifiers must be independent. We define the existential Non-Transferability of the UDVS against adaptively chosen message and chosen public key distinguisher $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ via the game with the simulator $\mathcal{S}$. The model is divided into two phases.

- Phase 1: $\mathcal{D}$ can submit KR, SS, DS, $\overline{\mathsf{DS}}$, DV and SK queries as defined in the model of **Unforgeability**, the simulator $\mathcal{S}$ responses to these queries as same as defined in the **Unforgeability** model.
- Challenge: When the distinguisher $\mathcal{D}$ decides the first phase is over, he submits $m^*, pk_s^*, pk_v^*$ to $\mathcal{S}$ as the challenge with the constraints that
  1. $pk_s^*$ can not be submitted as one of the SK queries during Phase 1.
  2. $(m^*, pk_s^*)$ can not be submitted as one of the SS queries during Phase 1.
  3. $(m^*, pk_s^*, pk_v^*)$ has never been submitted as one of the DS during Phase 1.

  As response, the simulator $\mathcal{S}$ chooses a random bit $b \in \{0, 1\}$. If $b = 0$, $\mathcal{S}$ runs DS algorithm and returns $\sigma_{\mathsf{DV}}$ to $\mathcal{D}$. Otherwise $b = 1$, $\mathcal{S}$ runs $\overline{\mathsf{DS}}$ algorithm and returns $\overline{\sigma_{\mathsf{DV}}}$ to $\mathcal{D}$.
- Phase 2: On receiving the challenging signature, the distinguisher can submit more queries except that:
  1. $pk_s^*$ can not be submitted as one of the SK queries during Phase 1.
  2. $(m^*, pk_s^*)$ can not be submitted as one of the SS queries during Phase 1.
  3. $(m^*, pk_s^*, pk_v^*)$ has never been submitted as one of the DS during Phase 1.

– Guessing: Finally, the distinguisher $\mathcal{D}$ outputs a guess $b'$. The adversary wins the game if $b = b'$.

The advantage of an adaptively chosen message and chosen public key distinguisher $\mathcal{D}$ has in the above game is defined as $\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS} = |\Pr[b' = b] - 1/2|$.

**Definition 5.** *We say a* **UDVS** *scheme is Non-Transferable against a* $(t, q_H, q_{KR}, q_{SS}, q_{DS}, q_{\overline{DS}}, q_{DV}, q_{SK})$ *adaptively chosen message and chosen public key distinguisher* $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ *if* $\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ *is negligible after making at most* $q_H$ *queries to the random oracle,* $q_{KR}$ *key registration queries,* $q_{SS}$ *SS queries,* $q_{DS}$ *DS queries,* $q_{\overline{DS}}$ $\overline{DS}$ *queries,* $q_{DV}$ *DV queries and* $q_{SK}$ SK *queries in time* $t$.

### 4.1 Strong Universal Designated Verifier Signature: Privacy of Signer

Given a UDVS scheme satisfies the security requirements defined above, one can not decide who generates the universal designated verifier signature. Both the signature holder $SH$ and the designated verifier $V$ can generate valid universal designated verifier signatures. However, in order to protect the privacy of the signer $S$ in some cases as described in [4], the algorithm DV cannot be executed publicly. Therefore, an additional strong notion: **Privacy of Signer** is introduced into the universal designated verifier signature.

Informally speaking, this property requires that given a message $m$ and a $V$ designated UDVS $\sigma_{\mathsf{DV}}$, without the secret keys of the designated verifier $V$ and the possible two original signers $S_0, S_1$, one can not decide which original signer $S_0$ or $S_1$ generates the standard signature $\sigma_{\mathsf{SS}}$. It is defined using the following games between the distinguisher $\mathcal{D}^{CMA,CPKA}_{Privacy,SUDVS}$ and the simulator $\mathcal{S}$:

– Phase 1: $\mathcal{D}$ can submit KR, SS, DS, $\overline{\mathsf{DS}}$, DV and SK queries as defined in the model of **Unforgeability**, the simulator $\mathcal{S}$ responds to these queries in the same way as defined in the model of **Unforgeability**.
– Challenge: When the distinguisher $\mathcal{D}$ decides the first phase is over, he submits $(m^*, pk^*_{s_0}, pk^*_{s_1}, pk^*_v)$ to $\mathcal{S}$ as the challenge with the constraints that
  1. Neither $pk^*_{s_0}, pk^*_{s_1}$ nor $pk^*_v$ has been submitted as one of the SK queries during Phase 1.
  2. Neither $(m^*, pk^*_{s_0})$ nor $(m^*, pk^*_{s_1})$ has been submitted as one of the SS queries during Phase 1.
  3. Neither $(m^*, pk^*_{s_0}, pk^*_v)$ nor $(m^*, pk^*_{s_1}, pk^*_v)$ has been submitted as one of DS and $\overline{\mathsf{DS}}$ queries during Phase 1.

  In response, the simulator $\mathcal{S}$ chooses a random bit $b \in \{0, 1\}$. If $b = 0$, $\mathcal{S}$ firstly runs $SS(m, sk^*_{s_0})$ to obtain $S_0$'s standard signature $\sigma_{\mathsf{SS}_0}$ on message $m^*$, then he runs DS algorithm and sets $\sigma^*_{\mathsf{DV}} = \sigma_{\mathsf{DV}_0}$. Otherwise $b = 1$, $\mathcal{S}$ runs $SS(m, sk^*_{s_1})$ to obtain $S_1$'s standard signature $\sigma_{\mathsf{SS}_1}$ on message $m^*$, then he runs $\overline{\mathsf{DS}}$ algorithm and sets $\sigma^*_{\mathsf{DV}} = \sigma_{\mathsf{DV}_1}$.

- Phase 2: On receiving the challenging signature $\sigma^*_{\mathsf{DV}}$, the distinguisher can submit more queries except that:
  1. $pk^*_{s_0}, pk^*_{s_1}$ and $pk^*_v$ can not be submitted as one of the SK queries during Phase 2.
  2. $(m^*, pk^*_{s_0})$ and $(m^*, pk^*_{s_1})$ can not be submitted as one of the SS queries during Phase 2.
  3. $(m^*, pk^*_{s_0}, pk^*_v)$ and $(m^*, pk^*_{s_1}, pk^*_v)$ can not be submitted as one of the DS or $\overline{\mathsf{DS}}$ queries during Phase 2.
  4. $(m^*, \sigma^*_{\mathsf{DV}}, pk^*_{s_0}, pk^*_v)$ and $(m^*, \sigma^*_{\mathsf{DV}}, pk^*_{s_1}, pk^*_v)$ can not be submitted as one of the DV queries during Phase 2.
- Guessing: Finally, the distinguisher $\mathcal{D}$ outputs a guess $b'$. The adversary wins the game if $b = b'$.

The advantage of an adaptively chosen message and chosen public key distinguisher $\mathcal{D}$ has in the above game is defined as $\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{Privacy,\ UDVS} = |\Pr[b' = b] - 1/2|$.

**Definition 6.** *We say a* **UDVS** *scheme satisfies the property: privacy of signer against a* $(t, q_H, q_{KR}, q_{\mathsf{SS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}})$ *adaptively chosen message and chosen public key distinguisher* $\mathcal{D}^{CMA,\ CPKA}_{Privacy,\ UDVS}$ *if* $\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{Privacy,\ UDVS}$ *is negligible after making at most* $q_H$ *queries to the random oracle,* $q_{KR}$ *key registration queries,* $q_{\mathsf{SS}}$ SS *queries,* $q_{\mathsf{DS}}$ DS *queries,* $q_{\overline{\mathsf{DS}}}$ $\overline{\mathsf{DS}}$ *queries,* $q_{\mathsf{DV}}$ DV *queries and* $q_{\mathsf{SK}}$ SK *queries in time* $t$.

As we call DVS with Privacy of Signer as Strong DVS (SDVS), we call that UDVS with this property as Strong UDVS (SUDVS).

## 5 Proposed Scheme

In this section, we will firstly describe our universal designated verifier signature scheme without delegatability. Then we provide the formal security analysis of our scheme in the random oracle model. Our scheme consists of the following algorithms:

CPG: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be a bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, for some prime number $p \geq 2^k$, $k$ be the system security number and $g$ be the generator of $\mathbb{G}_1$. $e$ denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Let $h_0 : \{0,1\}^* \to \mathbb{G}_1^*, h_1 : \{0,1\}^* \to \mathbb{Z}_p$ be two secure cryptographic hash functions.

KG: The Signer $S$ picks a secret value $x_s \in_R \mathbb{Z}_p^*$ and sets the secret key $sk_s := x_s$. Then $S$ computes the public key $pk_s = g^{x_s}$. Similarly, the verifier $V$'s secret/public key-pair is $(sk_v, pk_v) = (x_v, g^{x_v})$ where $x_v$ is randomly chosen in $\mathbb{Z}_p^*$.

SS: For a message $m$ to be signed, $S$ computes the standard signature $\sigma_{\mathsf{SS}} = h_0(m)^{sk_s}$.

SV: Given a message $m$, the standard signature $\sigma_{\mathsf{SS}}$ and $S$'s public key $pk_s$, one can check the equation $e(\sigma_{\mathsf{SS}}, g) \stackrel{?}{=} e(h_0(m), pk_s)$. If the equality holds, output $Acc$, otherwise, $Rej$.

$\mathsf{DS}$: Given the standard signature $\sigma_{\mathsf{SS}}$ and the verifier's public key $pk_v$, the signature holder $SH$ selects $r, c_v, d_v \in_R \mathbb{Z}_p$ and computes

1. $z_s = e(g,g)^r, z_v = g^{d_v} pk_v{}^{c_v}$
2. $c = h_1(m, pk_s, pk_v, z_s, z_v)$
3. $c_s = c - c_v \pmod p, d_s = \frac{g^r}{(\sigma_{\mathsf{SS}})^{c_s}}$

Then, $SH$ sends the universal designated verifier signature $\sigma_{DV} = (c_s, c_v, d_s, d_v)$ to the verifier $V$.

$\overline{\mathsf{DS}}$: Given the signer's public key $pk_s$ and the message $m$, the verifier $V$ selects $r, c_s \in_R \mathbb{Z}_p, d_s \in \mathbb{G}_1$ and computes

1. $z_s = e(d_s, g)e(h_0(m), pk_s)^{c_s}, z_v = g^r$
2. $c = h_1(m, pk_s, pk_v, z_s, z_v)$
3. $c_v = c - c_s \pmod p, d_v = r - c_v sk_v \pmod p$

The universal designated verifier generated by the verifier is $\overline{\sigma_{DV}} = (c_s, c_v, d_s, d_v)$.

$\mathsf{DV}$: Given the designated verifier signature $(c_s, c_v, d_s, d_v)$, anyone can check whether

$$c_s + c_v \stackrel{?}{=} h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^{d_v} pk_v{}^{c_v}) \pmod p$$

If the equality holds, output $Acc$, otherwise, $Rej$.

*Consistency:*

- $\mathsf{SV}$ Consistency: If $\sigma_{\mathsf{SS}}$ is generated by the algorithm $\mathsf{SS}$, then $\sigma_{\mathsf{SS}} = h_0(m)^{sk_s}$. Therefore $e(\sigma_{\mathsf{SS}}, g) = e(h_0(m)^{sk_s}, g) = e(h_0(m), pk_s)$. That is: $\Pr[\mathsf{SV}(cp, pk_s, m, \mathsf{SS}(cp, sk_s, m)) = Acc] = 1$
- $\mathsf{DV}$ Consistency of $\mathsf{DS}$: If $\sigma_{DS}$ is generated by the algorithm $\mathsf{DS}$, then $\sigma_{\mathsf{DV}} = (c_s, c_v, d_s, d_v)$ where $c_v, d_v \in_R \mathbb{Z}_p$ and

$$c_s = h_1(m, pk_s, pk_v, e(g,g)^r, g^{d_v} pk_v{}^{c_v}) - c_v \pmod p, r \in \mathbb{Z}_p \text{ and } d_s = \frac{g^r}{(\sigma_{\mathsf{SS}})^{c_s}}.$$

Therefore,

$$\begin{aligned} &h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^{d_v} pk_v{}^{c_v}) \\ &= h_1(m, pk_s, pk_v, e(g,g)^r, g^{d_v} pk_v{}^{c_v}) = c_s + c_v \pmod p \end{aligned}$$

That is: $\Pr[\mathsf{DV}(cp, pk_s, pk_v, m, \mathsf{DS}(cp, pk_s, pk_v, \sigma_{\mathsf{SS}}, m)) = Acc] = 1$.

- $\mathsf{DV}$ Consistency of $\overline{\mathsf{DS}}$: If $\overline{\sigma_{DS}}$ is generated by the algorithm $\overline{\mathsf{DS}}$, then $\sigma_{\mathsf{DV}} = (c_s, c_v, d_s, d_v)$ where $c_s \in_R \mathbb{Z}_p, d_s \in_R \mathbb{G}_1$ and

$$c_v = h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^r) - c_s \pmod p, r \in \mathbb{Z}_p$$

and $d_v = r - c_v sk_v \pmod p$. Therefore,

$$\begin{aligned} &h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^{d_v} pk_v{}^{c_v}) \\ &= h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^r) = c_s + c_v \pmod p \end{aligned}$$

That is: $\Pr[\mathsf{DV}(cp, pk_s, pk_v, m, \overline{\mathsf{DS}}(cp, pk_s, sk_v, m)) = Acc] = 1$

### 5.1 Security Analysis

**Theorem 1.** *If there exists an algorithm* $\mathcal{F}^{CMA,\ CPKA}_{EUF,\ DVS}$ *can* $(t, q_{h_0}, q_{h_1}, q_{\mathsf{KR}}, q_{\mathsf{SS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}}, \varepsilon)$*-break our UDVS scheme, then there exists a simulator* $\mathcal{S}$ *who can solve a random instance of the Computational Diffie Hellman problem on* $\mathbb{G}_1$ *with probability* $Succ^{CDH}_{\mathcal{S},\mathbb{G}_1} \geq \frac{1}{9}\frac{1}{(q_{\mathsf{SS}}+q_{\mathsf{SK}})^2}$*, after running* $\mathcal{F}$ *by* $\frac{12}{\varepsilon} + \frac{56q_{h_1}}{\varepsilon}$ *times with assumption* $\varepsilon \geq 56q_{h_1}\frac{1}{2^k}(q_{\mathsf{DS}} + q_{\overline{\mathsf{DS}}})(q_{h_1} + q_{\mathsf{DS}} + q_{\overline{\mathsf{DS}}}) + 1$*, where* $k$ *is the system security number.*

*Proof*: See Appendix.

**Theorem 2.** *Let* $(pk_s, sk_s) \leftarrow \mathsf{KG}(k)$*,* $(pk_v, sk_v) \leftarrow \mathsf{KG}(k)$*. If* $\mathcal{A}$ *is an algorithm, who can produce a valid UDVS on the message* $m$ *with probability* $\epsilon$ *in time* $t$*, then our scheme UDVS scheme is* $(\tau, \kappa)$ *non-delegatable in the random oracle where* $\kappa \geq \frac{1}{9}, \tau \leq \frac{16tq_{h_1}}{\varepsilon}$ *if* $h_1$ *is regarded as the random oracle,* $\mathcal{A}$ *asks at most* $q_{h_1}$ *queries to the random oracle and* $\epsilon \geq \frac{7q_{h_1}}{2^k}$*, where* $k$ *is the system security number.*

*Proof*: See Appendix.

**Theorem 3.** *The proposed* **UDVS** *scheme is non-transferable against a* $(t, q_{h_0}, q_{h_1}, q_{KR}, q_{\mathsf{SS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}})$ *adaptively chosen message and chosen public key distinguisher* $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$*.*

*Proof*: See Appendix.

## 6 Conclusion

Non-delegatability is a property recently introduced by Lipmaa, Wang and Bao as an essential property of (universal) designated verifier signature. In this paper, we propose the first universal designated verifier signature scheme without delegatability. Additionally, we refine the security models of the universal designated verifier signature and introduce the notion of the strong universal designated verifier signature. However, as there is no secure non-delegatable strong designated verifier signature, the scheme proposed in this paper is not a strong universal designated verifier signature. How to construct a non-delegatable strong (universal) designated verifier signature remains as an open research problem.

**Acknowledgement.** The authors would like to thank the anonymous referees of International Conference on Information and Communications Security (ICICS 2006) for the suggestions to improve this paper.

## References

1. D. Boneh and X. Boyen. Short signatures without random oracles. EUROCRYPT 2004. Lectue Notes in Computer Science 3027, pp. 56-73, Springer–Verlag, Berlin, 2004.

2. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. ASIACRYPT 2001, Lectue Notes in Computer Science 2248, pp. 514-532, Springer–Verlag, Berlin, 2001.
3. W. Diffie and M. Hellman. New directions in cryptography. IEEE IT, 22: 644-654, 1976.
4. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated Verifier Proofs and Their Applications. Advances in Cryptology - Eurocrypt '96, Lecture Notes in Computer Science 1070, pp. 143 - 154, Springer-Verlag, Berlin, 1996.
5. F. Laguillaumie and D. Vergnaud. Designated Verifiers Signature: Anonymity and Efficient Construction from *any* Bilinear Map. Fourth Conference on Security in Communication Networks'04 (SCN 04), Lecture Notes in Computer Science 3352, pp. 107 - 121, Springer-Verlag, Berlin, 2004.
6. F. Laguillaumie and D. Vergnaud. Multi-Designated Verifiers Signatures. Information and Communications Security (ICICS 2004), Lecture Notes in Computer Science 3269, pp. 495 - 507, Springer-Verlag, Berlin, 2004.
7. S. Goldwasser, S. Micali and R. Rivest. A Digital signature scheme secure against adaptively chosen message attacks. SIAM Journal on Computing, 17(2):281-308, 1988
8. X. Huang, Y. Mu, W. Susilo, and F. Zhang. Short Designated Verifier Proxy Signature from Pairings. The First International Workshop on Security in Ubiquitous Computing Systems (SecUbiq'05), Lecture Notes in Computer Science 3823, pp. 835 - 844, Springer-Verlag, Berlin, 2005.
9. Y. Li, H. Lipmaa and Dingyi Pei. On Delegatability of Four Desiganted Verifier Signatures. ICICS 2005, Lecture Notes in Computer Science 3783, pp. 61 - 71, Springer-Verlag, Berlin, 2005.
10. H. Lipmaa, G. Wang and F. Bao. Designated Verifier Signature Schemes: Attacks, New Security Notions and A New Construction. The 32nd International Colloquium on Automata, Languages and Programming, ICALP 2005, Lecture Notes in Computer Science 3580, pp. 459 - 471, Springer-Verlag, Berlin, 2004.
11. C. N. g, W. Susilo and Y. Mu. Universal Designated Multi Verifier Signature Schemes. The First International Workshop on Security in Networks and Distributed Systems (SNDS2005), IEEE Press, pp. 305 - 309, 2005.
12. W. Ogata, K. Kurosawa, and S.-H. Heng. The Security of the FDH Variant of Chaums Undeniable Signature Scheme. Public Key Cryptography - PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Lecture Notes in Computer Science 3385, pp. 328 - 345, Springer-Verlag, Berlin, 2005.
13. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, Volume 13 - Number 3, pp. 361-396, Springer-Verlag, Berlin, 2000.
14. R. L. Rivest, A. Shamir, Y. Tauman. How to Leak a Secret. Proceedings of Asiacrypt 2001, Lecture Notes in Computer Science 2248, pp. 552 - 565, Springer-Verlag, Berlin, 2001.
15. S. Saeednia, S. Kramer, and O. Markovitch. An Efficient Strong Designated Verifier Signature Scheme. The 6th International Conference on Information Security and Cryptology (ICISC 2003), pp. 40 - 54, Springer-Verlag, Berlin, 2003.
16. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk. Universal Designated-Verifier Signatures. Proceedings of Asiacrypt 2003, Lecture Notes in Computer Science 2894, pp. 523 - 543, Springer-Verlag, Berlin, 2003.

17. R. Steinfeld, H. Wang, and J. Pieprzyk. Efficient Extension of Standard Schnorr/RSA Signatures into Universal Designated-Verifier Signatures. Proceedings of PKC 2004, Lecture Notes in Computer Science 2947, pp. 86 - 100, Springer-Verlag, Berlin, 2004.
18. W. Susilo, F. Zhang, and Y. Mu. Identity-based Strong Designated Verifier Signature Schemes. Information Security and Privacy, 9th Australasian Conference, ACISP 2004, Lecture Notes in Computer Science 3108, pp. 313 - 324, Springer-Verlag, Berlin, 2004.
19. D. Vergnaud. New Extensions of Pairing-based Signatures into Universal Designated Verifier Signatures. Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006. Lecture Notes in Computer Science 4052, pp. 58-69, Springer-Verlag, Berlin, 2006. Also available at `http://www.math.unicaen.fr/~vergnaud/publications.php`
20. R. Zhang, J. Furukawa and H. Imai. Short signature and universal designated verifier signature without random oracles. Applied Cryptography and Network Security (ACNS 2005), Lecture Notes in Computer Science 3531, pp. 483 - 498, Springer-Verlag, Berlin, 2005.
21. F. Zhang, W. Susilo, Y. Mu and X. Chen. Identity-based Universal Designated Verifier Signatures. The First International Workshop on Security in Ubiquitous Computing Systems (SecUbiq'05), Nagasaki, Japan, Lecture Notes in Computer Science 3823, pp. 825-834, Springer-Verlag, Berlin, 2005.

## Appendix

*Proof of Theorem 1*: Suppose there exists a forger $\mathcal{F}$ who can $(t, q_{\mathsf{KR}}, q_{h_0}, q_{h_1}, q_{\mathsf{SS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}}, \varepsilon)$ break our UDVS scheme. We will show there exists a simulator $\mathcal{S}$ who can use $\mathcal{F}$ to solve the Computational Diffie Hellman problem. In the proof, we assume that when $\mathcal{F}$ requests the signature of the signer $pk$ or asks the secret key corresponding to the public key $pk$, $\mathcal{F}$ has obtained the public key $pk$ from $KR$ queries. We will regard hash functions $h_0, h_1$ as the random oracles.

Let $(\mathbb{G}_1, \mathbb{G}_T)$ be a bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, for some prime number $p \geq 2^k$, $k$ be the system security number, $g$ be the generator of $\mathbb{G}_1$ and $e$ denote the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. Let $(g, g^a, g^b)$ be a random instance of the computational diffie hellman problem on $\mathbb{G}_1$.

- $\mathsf{Setup}$: $\mathcal{S}$ returns $cp = (\mathbb{G}_1, \mathbb{G}_T, g, p, e)$ to the forger $\mathcal{F}$.
- $\mathsf{KR}$ queries: At any time, $\mathcal{F}$ can register for the $i^{th}$ user. In response, $\mathcal{S}$ will maintain a *pk-list* which stores his responses to such queries. For a new query, $\mathcal{S}$ chooses a number $e_i \in \{0, 1\}$ such that $\Pr[e_i = 1] = \frac{1}{q_{\mathsf{SS}} + q_{\mathsf{SK}}}$. If $e_i = 0$, $\mathcal{S}$ sets $sk_i = f_i \in_R \mathbb{Z}_p^*, pk_i = g^{sk_i}$. Otherwise, $e_i = 1$ and $\mathcal{S}$ sets $pk_i = (g^a)^{f_i}$, where $f_i \in_R \mathbb{Z}_p^*$. For either case, $\mathcal{S}$ adds $(pk_i, e_i, f_i)$ into the *pk-list* and returns $pk_i$ to $\mathcal{F}$ as the answer.
- $h_0$ queries: $\mathcal{F}$ can issue $h_0$ queries for the message $m_i$. In response, $\mathcal{S}$ will maintain an $h_0$-*list* which stores his responses to such queries. For a new query, $\mathcal{S}$ chooses a number $x_i \in \{0, 1\}$ such that $\Pr[x_i = 1] = \frac{1}{q_{\mathsf{SS}} + q_{\mathsf{SK}}}$. Firstly, $\mathcal{S}$ chooses $y_i \in_R \mathbb{Z}_p^*$. If $x_i = 0$, sets $h_0(m_i) = w_i = g^{y_i}$. Otherwise,

$\mathcal{S}$ sets $h_0(m_i) = w_i = (g^b)^{y_i}$. $\mathcal{S}$ then adds $(m_i, x_i, y_i, w_i)$ to the $h_0$-*list* and returns $w_i$ to $\mathcal{F}$.

- $h_1$ queries: $\mathcal{F}$ can issue $h_1$ queries with the query $Q_i = (m_i, pk_s, pk_v, z_s, z_v)$. In response, $\mathcal{S}$ will maintain an $h_1$-*list* which stores his responses to such queries. For a new query $Q_i$, $\mathcal{S}$ chooses $u_i \in \mathbb{Z}_p$ and adds $(Q_i, u_i)$ to $h_1$-*list*. Then $\mathcal{S}$ returns $u_i$ to $\mathcal{F}$ as the answer.
- $\mathsf{SS}$ queries: On a standard sign query $(m_i, pk_j)$, $\mathcal{S}$ firstly checks $h_0$-*list* to obtain $w_i = h_0(m_i)$. If $m_i$ has not been submitted by $\mathcal{F}$ as one of the $h_0$ queries, $\mathcal{S}$ generates the value of $h_0(m_i)$ as he responds to $h_0$ queries and adds $(m_i, x_i, y_i, w_i)$ into the $h_0$-*list*. By assumption, $(pk_j, e_j, f_j)$ has been in the *pk-list*. (1) If $e_j = 1, x_i = 1$, $\mathcal{S}$ reports failure and aborts. (2) Else $e_j = 1, x_i = 0$, $\mathcal{S}$ can compute the standard signature $\sigma_{\mathsf{SS}} = (pk_j)^{y_i}$. (3) Else $e_j = 0$, $\mathcal{S}$ can compute the standard signature $\sigma_{\mathsf{SS}} = (w_i)^{f_j}$.
- $\mathsf{DS}$ queries: $\mathcal{F}$ can request a designated verifier signature $\sigma_{\mathsf{DV}}$ of the message $m_i$ under the public key $(pk_s, pk_v)$, where $pk_s$ denotes the signer and $pk_v$ denotes the designated verifier. $\mathcal{S}$ firstly tries to compute $\sigma_{\mathsf{SS}}$ of this message $m$. If $m_i$ has not been submitted by $\mathcal{F}$ as one of the $h_0$ queries, $\mathcal{S}$ generates $w_i = h_0(m_i)$ as its response to $h_0$ queries and adds $(m_i, x_i, y_i, w_i)$ into the $h_0$-*list*. By assumption, $(pk_s, e_s, f_s), (pk_v, e_v, f_v)$ have been in the *pk-list*. If $x_i \neq 1$ or $e_s \neq 1$, $\mathcal{B}$ can generate the signature $\sigma_{\mathsf{SS}}$ as its response to the $\mathsf{SS}$ queries, then he runs $\mathsf{DS}$ algorithm as defined in Section 5. If $x_i = 1, e_s = 1$, $\mathcal{S}$ chooses $c_s, c_v, d_v \in_R \mathbb{Z}_p$ and $d_s \in_R \mathbb{G}_1$ and computes $z_s = e(d_s, g)e(w_i, pk_s)^{c_s}, z_v = g^{d_v}(pk_v)^{c_v}$. Then $\mathcal{S}$ sets $Q_i = (m_i, pk_s, pk_v, z_s, z_v)$ and adds $(Q_i, c_s + c_v)$ into the $h_1$-*list*. If query $Q_i$ has been requested by $\mathcal{F}$, $\mathcal{S}$ reports failure and aborts.
- $\overline{\mathsf{DS}}$ queries: $\mathcal{F}$ can request designated verifier signature $\overline{\sigma_{\mathsf{DS}}}$ of the message $m_i$ under the public key $(pk_s, pk_v)$, where $pk_s$ denotes the signer and $pk_v$ denotes the designated verifier. In response, $\mathcal{S}$ firstly generates the value of $h_0(m_i)$ as its response to $h_0$ queries and adds $(m_i, x_i, y_i, w_i)$ into the $h_0$-*list*. By assumption, $(pk_s, e_s, f_s), (pk_v, e_v, f_v)$ have been in the *pk-list*. If $e_v = 0$, then $sk_v = f_v$. $\mathcal{S}$ can run the $\overline{\mathsf{DS}}$ algorithm as defined in the Section 5. Otherwise, $e_v = 1$. In this case, $pk_v = (g^a)^{f_v}$ and $\mathcal{S}$ does not know the secret key $sk_v$. Similarly as its response to $\mathsf{DS}$ queries, $\mathcal{S}$ chooses $c_s, c_v, d_v \in_R \mathbb{Z}_p$ and $d_s \in_R \mathbb{G}_1$ and computes $z_s = e(d_s, g)e(w_i, pk_s)^{c_s}, z_v = g^{d_v}(pk_v)^{c_v}$. Then, $\mathcal{S}$ sets $Q_i = (m_i, pk_s, pk_v, z_s, z_v)$ and adds $(Q_i, c_s + c_v)$ into the $h_1$-*list*. Similarly, if query $Q_i$ has been requested by $\mathcal{F}$, $\mathcal{S}$ reports failure and aborts.
- $\mathsf{DV}$ queries: $\mathcal{F}$ can execute the $\mathsf{DV}$ algorithm by himself since $\mathsf{DV}$ algorithm does not need the secret keys of the signer and the designated verifier.
- $\mathsf{SK}$ queries: $\mathcal{F}$ can request the secret key corresponding to the public key $pk_i$. By assumption, $(pk_i, e_i, f_i)$ has been in the the *pk-list*. If $e_i = 0$, $\mathcal{S}$ returns $f_i$ to $\mathcal{F}$. Otherwise, if $e_i = 1$, $\mathcal{B}$ reports failure and aborts.

Finally, $\mathcal{F}$ outputs a forged messgae/signature pair $(m^*, \sigma^*_{\mathsf{DV}})$ $(\sigma^*_{\mathsf{DV}} = (c^*_s, c^*_v, d^*_s, d^*_v))$ under the public keys $(pk^*_s, pk^*_v)$ such that:

1. $Acc \leftarrow \mathsf{DV}(cp, pk^*_s, pk^*_v, m^*, \sigma^*_{\mathsf{DV}})$.
2. $(m^*, pk^*_S)$ has never been submitted as one of the $\mathsf{SS}$ queries.

3. $(m^*, pk_s^*, pk_v^*)$ has never been submitted as one of the $\mathsf{DS}$ or $\overline{\mathsf{DS}}$ queries.
4. Neither $pk_s^*$ nor $pk_v^*$ has been submitted as one of the $\mathsf{SK}$ queries.

Therefore, if $\mathcal{S}$ does not abort during the simulation, $\mathcal{F}$ can output a valid forgery with probability greater than $\varepsilon$. Now it remains to compute the probability that $\mathcal{S}$ does not abort:

1. The probability $\mathcal{S}$ does not abort during the $\mathsf{SS}$ queries is greater than $(1-(\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^2)^{q_{\mathsf{SS}}}$.
2. The probability $\mathcal{S}$ does not abort during the $\mathsf{SK}$ queries is greater than $(1-\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^{q_{\mathsf{SK}}}$.
3. The probability that there is no collision happens during the $\mathsf{DS}$ or $\overline{\mathsf{DS}}$ queries is greater than $(1-\frac{q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}}}{2^k})^{q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}}}$.

Therefore, the probability $\mathcal{S}$ does not abort during the simulation is

$$\begin{aligned}&(1-(\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^2)^{q_{\mathsf{SS}}}(1-\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^{q_{\mathsf{SK}}}(1-\frac{q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}}}{2^k})^{q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}}}\\ \geq\ &(1-\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^{q_{\mathsf{SS}}+q_{\mathsf{SK}}}(1-\frac{(q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})(q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})}{2^k})\end{aligned}$$

Since $h_1$ is regarded as a random oracle, the probability that $\mathcal{F}$ succeeds and has not submitted $Q^* = (m^*, pk_s^*, pk_v^*, z_s^*, z_v^*)$ ($z_s^* = e(d_s^*, g)e(w^*, pk_s^*)^{c_s^*}$, $z_v^* = g^{d_v^*}(pk_v^*)^{c_v^*}$) to the random oracle $h_1$ is less than $\frac{1}{2^k}$. Therefore, $\mathcal{F}$ can output a valid forgery $(m^*, pk_s^*, pk_v^*, \sigma_{\mathsf{DV}}^*)$ such that $Q^*$ has been submitted to the random oracle with probability $\varepsilon' \geq \varepsilon(1-\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^{q_{\mathsf{SS}}+q_{\mathsf{SK}}}(1-\frac{(q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})(q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})}{2^k}) - \frac{1}{2^k} \geq \varepsilon(1-\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^{q_{\mathsf{SS}}+q_{\mathsf{SK}}} - \frac{(q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})(q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})+1}{2^k} \geq \frac{\varepsilon}{7}$. (with assumption $\varepsilon \geq 56q_{h_1}\frac{(q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})(q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})+1}{2^k}$). Therefore, the probability $\mathcal{F}$ succeeds with no collision happens is more than $\frac{\varepsilon}{7}$. Now, we apply the forking lemma [13]. Let $\mathcal{F}$ begins with the the random tape $\Omega$ and $h_1$ is regarded as the random oracle $\Theta$. If $\mathcal{S}$ runs $\mathcal{F}$ $12/\varepsilon$ times with the random oracle $\Omega$ and $\Theta$, $\mathcal{S}$ gets a least one pair $(\Omega, \Theta)$ with success probability $1-e^{-12/7} \geq \frac{4}{5}$ such that $Q^*$ has been requested as one of the $h_1$ queries. Let the $\beta^{th}$ queries $Q_\beta = Q^*$ and the response is $u_{\beta 1}^*$. If $\mathcal{S}$ replays forger $\mathcal{F}$ with different response $u_{\beta 2}^*$ to $Q_\beta$ and the same responses to $Q_i, i \leq \beta$, $\mathcal{S}$ can obtain another forged signature on the same message with probability more than $\frac{\varepsilon'}{4q_{h_1}} - 1/2^k \geq \frac{\varepsilon}{56q_{h_1}}$ (with assumption $\varepsilon \geq 56q_{h_1}\frac{(q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})(q_{h_1}+q_{\mathsf{DS}}+q_{\overline{\mathsf{DS}}})+1}{2^k}$).

Due to the forking lemma, after running $\mathcal{F}$ $\frac{12}{\varepsilon} + \frac{56q_{h_1}}{\varepsilon}$ times, $\mathcal{S}$ can obtain two valid universal designated verifier signatures $(m^*, c_{s1}^*, c_{v1}^*, d_{s1}^*, d_{v1}*)$ and $(m^*, c_{s2}^*, c_{v2}^*, d_{s2}^*, d_{v2}^*)$ with probability greater than $1/9$ and

$$e(d_{s1}^*, g)e(w^*, pk_s^*)^{c_{s1}^*} = e(d_{s2}^*, g)e(w^*, pk_s^*)^{c_{s2}^*} \quad (1)$$

$$g^{d_{v1}^*}(pk_v^*)^{c_{v1}^*} = g^{d_{v2}^*}(pk_v^*)^{c_{v2}^*} \quad (2)$$

but $c_{s1}^* + c_{v1}^* \neq c_{s2}^* + c_{v2}^* \pmod p$.

1. If $c^*_{s1} \neq c^*_{s_2} \pmod p$, with probability $\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}}$, $w^* = (g^b)^{y^*}$. Additionally, with probability $\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}}$, $pk^*_s = (g^a)^{f^*_s}$. Therefore, with probability $(\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^2$, the following equation holds from equation (1): $d^*_{s1}(g^{ab})^{f^*_s y^* c^*_{s1}} = d^*_{s2}(g^{ab})^{f^*_s y^* c^*_{s2}}$. Therefore $g^{ab} = (d^*_{s1}/d^*_{s2})^{(f^*_s y^*)^{-1}(c^*_{s2}-c^*_{s1})^{-1}}$
2. Otherwise, $c^*_{v1} \neq c^*_{v_2} \pmod p$, then $sk^*_v = (c^*_{v2} - c^*_{v1})^{-1}(d^*_{v1} - d^*_{v2})$ (due to the equation (2)). With probability $\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}}$, $pk^*_v = (g^a)^{f^*_v}$, therefore, $a = (f^*_v)^{-1}(c^*_{v2} - c^*_{v1})^{-1}(d^*_{v1} - d^*_{v2})$ and $g^{ab} = (g^b)^{(f^*_v)^{-1}(c^*_{v2}-c^*_{v1})^{-1}(d^*_{v1}-d^*_{v2})}$.

In either way, $\mathcal{S}$ can compute $g^{ab}$ with probability $\frac{1}{9}(\frac{1}{q_{\mathsf{SS}}+q_{\mathsf{SK}}})^2$ after running $\mathcal{F}$ by $\frac{12}{\varepsilon} + \frac{56q_{h_1}}{\varepsilon}$ times. □

*Proof of Theorem 2*: In the extraction, $\mathcal{K}$ will act as the random oracle to reply $\mathcal{A}$'s $q_{h_1}$ $h_1$ queries. Let $pk_s = g^{sk_s}, pk_v = g^{sk_v}$ be the public keys of the signer and the verifier. For each $h_1$ query $Q_i = (m_i, pk_s, pk_v, z_s, z_v)$, $\mathcal{K}$ chooses a random number $u_i \in \mathbb{Z}^*_p$ and sets $h_1(Q_i) = u_i$. If $\mathcal{A}$ can produce a valid universal designated verifier signature with probability $\varepsilon \geq \frac{7q_{h_1}}{2^k}$, due to the forking lemma, $\mathcal{K}$ can use $\mathcal{A}$ to obtain two valid signatures on the same message $m$ with probability $\kappa \geq \frac{1}{9}$ after running $\mathcal{A}$ by $\frac{2}{\varepsilon} + \frac{14q_{h_1}}{\varepsilon}$ times. Let $(m, c_s, c_v, d_s, d_v)$ and $(m, c'_s, c'_v, d'_s, d'_v)$ be these two valid signatures, then $e(d_s, g)e(h_0(m), pk_s)^{c_s} = e(d'_s, g)e(h_0(m), pk_s)^{c'_s}$, $g^{d_v} pk_v^{c_v} = g^{d'_v} pk_v^{c'_v}$ but $c_s + c_v \neq c'_s + c'_v \pmod q$. Then the following two equations hold:

$$e(h_0(m), pk_s)^{c_s - c'_s} = e(\frac{d_s}{d'_s}, g), sk_v(c_v - c'_v) = d'_v - d_v.$$

If $c_s \neq c'_s$, then $e(h_0(m), pk_s) = e(\frac{d_s}{d'_s}, g)^{(c_s - c'_s)^{-1}}$. Therefore $\mathcal{K}$ can compute $\sigma_{\mathsf{SS}} = (\frac{d_s}{d'_s})^{(c_s - c'_s)^{-1}}$. Otherwise, $c_s = c'_s, c_v \neq c'_v$. $\mathcal{K}$ can compute $sk_v = (d'_v - d_v)(c_v - c'_v)^{-1}$. Therefore, $\mathcal{K}$ can extract $\sigma_{\mathsf{SS}}$ or $sk_v$ with probability $\kappa \geq \frac{1}{9}$ in time $\tau \leq \frac{16tq_{h_1}}{\varepsilon}$. □

*Proof of Theorem 3*: In the proof, the simulator $\mathcal{S}$ runs $\mathsf{KG}$ algorithm to generate all the secret/public keys. He then sends the public keys to the distinguisher $\mathcal{D}$ and keep the secret keys only known to himself. Because $\mathcal{S}$ has the knowledge of all secret keys, he can run $\mathsf{SS}$, $\mathsf{DS}$ and $\overline{\mathsf{DS}}$ to response $\mathcal{D}$'s queries. Neither will he abort during $\mathcal{D}$'s $\mathsf{SK}$ queries. For each $h_0$ queries, $\mathcal{S}$ chooses a random element in $\mathbb{G}^*_1$ as the response. Similarly, $\mathcal{S}$ chooses a random number in $\mathbb{Z}_p$ as the response to each $h_1$ queries. Therefore, $\mathcal{S}$ will not abort during the simulations.

Firstly we show that the distribution of signature $\sigma_{\mathsf{DV}}$ generated by the algorithm $\mathsf{DS}$ is uniform in $\mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{G}_1 \times \mathbb{Z}_p$. Let $\sigma_{\mathsf{DV}} = (c_s, c_v, d_s, d_v)$ is the universal designated verifier signature generated by the $\mathsf{DS}$ algorithm,

$$\sigma_{\mathsf{DV}} = (c_s, c_v, d_s, d_v) : \begin{cases} c_s = h_1(m, pk_s, pk_v, e(g,g)^r, g^{d_v} pk_v{}^{c_v}) - c_v \pmod p, \\ \text{where } r, c_v, d_v \in_R \mathbb{Z}_p \text{ and} \\ d_s = \frac{g^r}{(\sigma_{\mathsf{SS}})^{c_s}}, r \in_R \mathbb{Z}_p \end{cases}$$

Therefore, for a randomly chosen signature $\sigma^* = (c_s^*, c_v^*, d_s^*, d_v^*)$, the probability $\Pr[\sigma_{\mathsf{DV}} = \sigma^*] = \frac{1}{p^4}$.

Then we show that the distribution of signature simulated by the algorithm $\overline{\mathsf{DS}}$ is also uniform $\mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{G}_1 \times \mathbb{Z}_p$.

$$\overline{\sigma_{\mathsf{DV}}} = (c_s, c_v, d_s, d_v) : \begin{cases} c_v = h_1(m, pk_s, pk_v, e(d_s, g)e(h_0(m), pk_s)^{c_s}, g^r) - c_s \\ \quad (\text{mod } p), \text{where } r, c_s \in_R \mathbb{Z}_p, d_s \in_R \mathbb{G}_1 \text{ and} \\ d_v = r - c_v sk_v \pmod p, r \in_R \mathbb{Z}_p \end{cases}$$

Therefore, for a randomly chosen signature $\sigma^* = (c_s^*, c_v^*, d_s^*, d_v^*)$, the probability $\Pr[\overline{\sigma_{\mathsf{DV}}} = \sigma^*] = \frac{1}{p^4}$.

Therefore, given a valid universal designated verifier signature, one can not distinguish whether it is generated by $\mathsf{DS}$ algorithm or $\overline{\mathsf{DS}}$ algorithm. Hence, our proposed scheme satisfies the untransferability property.