Int. J. Inf. Secur. (2008) 7:171–183
DOI 10.1007/s10207-007-0021-2

REGULAR CONTRIBUTION

Secure universal designated verifier signature without random oracles

Xinyi Huang · Willy Susilo · Yi Mu · Wei Wu

Published online: 5 April 2007

Abstract In Asiacrypt 2003, the concept of universal designated verifier signature (**UDVS**) was introduced by Steinfeld, Bull, Wang and Pieprzyk. In the new paradigm, *any* signature holder (not necessarily the signer) can designate the publicly verifiable signature to any desired designated verifier (using the verifier's public key), such that only the designated verifier can believe that the signature holder does have a valid publicly verifiable signature, and hence, believes that the signer has signed the message. Any other third party cannot believe this fact because this verifier can use his secret key to create a valid UDVS which is designated to himself. In ACNS 2005, Zhang, Furukawa and Imai proposed the first UDVS scheme without random oracles. In this paper, we give a security analysis to the scheme of Zhang et al. and propose a novel UDVS scheme without random oracles based on Waters' signature scheme, and prove that our scheme is secure under the Gap Bilinear Diffie Hellman assumption.

Keywords Universal designated verifier signature · Gap Bilinear Diffie Hellman problem · Security analysis · Random oracle

X. Huang (✉) · W. Susilo · Y. Mu · W. Wu
School of Computer Science and Software Engineering,
University of Wollongong, Northfields Avenue,
Wollongong NSW 2522, Australia
e-mail: xh068@uow.edu.au

W. Susilo
e-mail: wsusilo@uow.edu.au

Y. Mu
e-mail: ymu@uow.edu.au

W. Wu
e-mail: weiwu81@gmail.com

1 Introduction

Digital signature, as introduced in the pioneering paper of Diffie and Hellman [7], allows a party with a private key to sign a message such that anyone who has access to the corresponding public key can verify the authenticity of the message. The verifier of a signature can convince any third party about the fact by presenting a digital signature on a message. The public verifiability of digital signatures is of great convenience for many applications, but it is unsuitable for some other applications where a verifier does not want to present the publicly verifiable signatures to other parties, such as those associated with certificates for hospital records, income summary, etc.

Universal designated verifier signature, as introduced by Steinfeld et al. [11] in Asiacrypt 2003, is an important tool to protect the privacy of the signature holder from dissemination of signatures by verifiers. Given a publicly verifiable signature from the signer, a signature holder 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 parties cannot believe it because this verifier can use his secret key to create a valid UDVS which is the same as the one 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, as introduced by Jakobsson et al. [9]. 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 [5], Steinfeld et al. [11] proposed the first UDVS scheme in Asiacrypt 2003. Steinfeld et al.

also showed how to obtain a UDVS scheme from the Schnorr/RSA signature scheme in PKC 2004 [12]. Zhang et al. [16] extended this notion to the Identity-based setting and proposed two identity-based UDVS schemes. However, the security of all the above UDVS schemes are based on the random oracle model [16]. The first UDVS scheme without random oracle was proposed by Zhang et al. [15] in ACNS 2005, where a variant of BB's [4] short signature scheme without random oracle is used as the building block.

In Asiacrypt 2005, Baek et al. [2] introduced the notion of universal designated verifier signature proof (**UDVSP**) which removes the requirement that the designated verifier must create a public key using the parameters of signer's public key system. Baek et al. also provided two interactive protocols [2] based on BLS [5] and BB [4] publicly verifiable signature schemes, respectively.

Our contribution

In this paper, we firstly formalize the security models of UDVS. Then, we analyze the UDVS scheme without random oracle proposed in [15]. The distinguisher $\mathcal{D}$ against this scheme can have non-negligible advantage in the model of the non-transferability defined in this paper. However, this problem does not exist in the definition of the model of Zhang et al. [15]. We also provide a new UDVS scheme without random oracle which is secure in our stronger model. The security of our scheme is based on the difficulty of Gap Bilinear Diffie Hellman problem.

Organization

The rest of this paper is organized as follows. In the next section, we will provide some preliminaries and background required throughout the paper. In Sect. 3, we introduce the formal models of the universal designated verifier signature. We review and analyze the scheme of Zhang et al. [15] in Sect. 4. We provide our UDVS scheme without random oracle with security analysis in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Preliminaries

In this section, we will review some fundamental backgrounds used throughout this paper, namely bilinear pairings and their complexity assumptions.

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 a group $\mathbb{G}_T$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ as above, and e, and the group action in $\mathbb{G}_1$ and $\mathbb{G}_T$ can be computed efficiently. See [5] for more details on the construction of such pairings.

2.2 Complexity assumptions

Definition 1 Bilinear Diffie-Hellman (BDH) Problem in $(\mathbb{G}_1, \mathbb{G}_T)$
Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some unknown $a, b, c \in \mathbb{Z}_p$, compute out $w = e(g, g)^{abc} \in \mathbb{G}_T$.

Definition 2 Decisional Bilinear Diffie-Hellman (DBDH) Problem in $(\mathbb{G}_1, \mathbb{G}_T)$
Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some unknown $a, b, c \in \mathbb{Z}_p$ and $w \in \mathbb{G}_T$, decide whether $w \stackrel{?}{=} e(g, g)^{abc}$.

A DBDH oracle $\mathcal{O}_{DBDH}$ is that on input $g, g^a, g^b, g^c \in \mathbb{G}_1$ and $w \in \mathbb{G}_T$, outputs 1 if $w = e(g, g)^{abc}$ and 0 otherwise.

Definition 3 Gap Bilinear Diffie-Hellman (GBDH) Problem in $(\mathbb{G}_1, \mathbb{G}_T)$
Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some unknown $a, b, c \in \mathbb{Z}_p$, compute out $w = e(g, g)^{abc} \in \mathbb{G}_T$ with the help of $\mathcal{O}_{DBDH}$.

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

$$Succ^{GBDH}_{\mathcal{A},\mathbb{G}_1,\mathbb{G}_T} = \Pr[e(g, g)^{abc} \leftarrow \mathcal{A}(\mathbb{G}_1, \mathbb{G}_T, g, g^a, g^b, g^c, \mathcal{O}_{DBDH})].$$

Definition 4 Gap Bilinear Diffie-Hellman (GBDH) Assumption in $(\mathbb{G}_1, \mathbb{G}_T)$
Given $g, g^a, g^b, g^c \in \mathbb{G}_1$ for some unknown $a, b, c \in \mathbb{Z}_p$, $Succ^{GBDH}_{\mathcal{A},\mathbb{G}_1,\mathbb{G}_T}$ is negligible.

3 Formal models of UDVS

Our universal designated verifier signature scheme consists of the following algorithms: **UDVS**= (CPG, SKG, VKG, PS, PV, DS, $\overline{\text{DS}}$, DV, P_{KR}).

- **Common Parameter Generation** CPG: a probabilistic algorithm, given a security parameter k, outputs a strong cp which denotes the common scheme parameters (cp is shared by all the users in the system). That is: $cp \leftarrow$ CPG(k).

- **Signer Key Generation** SKG: a probabilistic algorithm, on input a common parameter cp, outputs a secret/public key-pair (sk_s, pk_s) for the $Signer$. That is: $(sk_s, pk_s) \leftarrow \mathsf{SKG}(cp)$.
- **Verifier Key Generation** VKG: a probabilistic algorithm, on input a common parameter cp, outputs a secret/public key-pair (sk_v, pk_v) for the $Verifier$. That is $(sk_v, pk_v) \leftarrow \mathsf{VKG}(cp)$.
- **Signing** PS: a probabilistic algorithm, on input the common parameter cp, $Signer's$ secret key sk_s and the message m, outputs $Signer's$ publicly verifiable (PV) signature σ_{PV}. That is: $\sigma_{\mathsf{PV}} \leftarrow \mathsf{PS}(cp, sk_s, m)$.
- **Public Verification** PV: a deterministic algorithm, on input the common parameter cp, $Signer's$ public key pk_s, the signed message m and the PV signature σ_{PV}, outputs verification decision $d \in \{Acc, Rej\}$. That is: $\{Acc, Rej\} \leftarrow \mathsf{PV}(cp, pk_s, m, \sigma_{\mathsf{PV}})$.
- **Designation by Signature Holder** DS: a probabilistic algorithm, on input the common parameter cp, $Signer's$ public key pk_s, $Verifier's$ public key pk_v, the signed message m and the PV signature σ_{PV}, outputs the designated verifier(DV) signature σ_{DV}. That is: $\sigma_{\mathsf{DV}} \leftarrow \mathsf{DS}(cp, pk_s, pk_v, m, \sigma_{\mathsf{PV}})$.
- **Designation by Verifier** $\overline{\mathsf{DS}}$: a probabilistic algorithm, on input the common parameter cp, $Signer's$ public key pk_s, $Verifier's$ secret key sk_v and the message m outputs the designated verifier(DV) signature $\overline{\sigma_{\mathsf{DV}}}$ which is designated to himself. That is: $\overline{\sigma_{\mathsf{DV}}} \leftarrow \overline{\mathsf{DS}}(cp, pk_s, sk_v, m)$.
- **Designation Verification** DV: a deterministic algorithm, on input the common parameter cp, $Signer's$ public key pk_s, $Verifier's$ secret key sk_v, the signed message m and the DV signature σ_{DV}, outputs the verification decision $d \in \{Acc, Rej\}$. That is: $\{Acc, Rej\} \leftarrow \mathsf{DV}(cp, pk_s, sk_v, m, \sigma_{\mathsf{DV}})$.
- **Verifier Key-Registration** $\mathsf{P_{KR}}$(KRA,VER): a protocol between a "Key Registration Authority(KRA)" and a "Verifier(VER)" who wishes to register a verifier's public key. On common input cp, the algorithm KRA and VER interact by sending messages alternately from one to another. At the end of the protocol, KRA outputs a pair $(pk_v, Auth)$, where pk_v is the $Verifier's$ public key, and $Auth \in \{Acc, Rej\}$ is a key registration authorization decision. We write $\mathsf{P_{KR}}(\mathsf{KRA}, \mathsf{VER}) = (pk_v, Auth)$ to denote this protocol's output.

  The purpose of the **Verifier Key-Registration** is to force the $Verifier$ to "know" the secret key corresponding to his public key, in order to enforce the non-transferability privacy property which will be defined later [11].

Remark Compared with the models defined in [11,15], we add an additional algorithm $\overline{\mathsf{DS}}$ to describe directly how a designated verifier can create a valid UDVS which is designated to himself. It is also for the convenience to analyze the *non-transferability privacy* later.

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

- PV **Consistency**: this property requires that the PV signature produced by the PS algorithm is accepted as valid by the PV algorithm. That is:

$$\Pr[\mathsf{PV}(cp, pk_s, m, \mathsf{PS}(cp, sk_s, m)) = Acc] = 1.$$

- DV **Consistency of** DS: this property requires that the DV signature produced by the DS algorithm is accepted as valid by the DV algorithm. That is:

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

- DV **Consistency of** $\overline{\mathsf{DS}}$: this property requires that the DV signature produced by the $\overline{\mathsf{DS}}$ algorithm is accepted as valid by the DV algorithm. That is:

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

3.1 Security properties of UDVS

Unforgeability
Actually, there are two types of unforgeability properties that can be used [11]. The first property, *publicly verifiable signature unforgeability* PV-Unforgeability, is just the usual existential unforgeability notion under chosen message attacker [8] for the standard publicly verifiable signature scheme PS, which states that anyone should not be able to forge a PV signature of the signer. The second property, *designated verifier signature unforgeability* (DV-Unforgeability), requires that it is difficult for an attacker to forge a DV signature σ^*_{DV} by the signer on a new message m^*, such that the pair (M^*, σ^*_{DV}) passes the DV algorithm with respect to a designated verifier's public key pk^*_v, which states that for any message, an adversary without the PV signature should not be able to convince a designated verifier of holding such a PV signature. DV-Unforgeability always implies the PV-Unforgeability [11]. Thus, it is enough to consider only DV-Unforgeability.

Let **UDVS** = (CPG, SKG, VKG, PS, PV, DS, $\overline{\mathsf{DS}}$, DV, $\mathsf{P_{KR}}$) be a UDVS scheme. We define the existential unforgeability of the UDVS against adaptive chosen public key and chosen message attacker $\mathcal{A}^{CMA,\ CPKA}_{EUF,\ UDVS}$. In the defined model, we allow adversaries to submit SecretKey(SK) queries adaptively, thus the adversaries can corrupt some designated verifiers and adaptively choose the target designated verifier, which reflects more essence of real world adversaries [15]. We will define it via the following game with the challenger $\mathcal{C}$:

- Setup: The challenger $\mathcal{C}$ runs the CPG algorithm to obtain the common parameters cp. $\mathcal{C}$ also generates $Signer's$ secret/public key-pair (sk_s, pk_s) from the SKG. Additionally, $\mathcal{C}$ runs VKG some times to obtain n potential $Verifier's$ secret/public key-pairs (sk_{v_i}, pk_{v_i}). $\mathcal{C}$ then sends the common parameters cp, $Signer's$ public key pk_s and all $Verifier's$ public keys $pk_{v_i}, i \in \{1, 2, \ldots, n\}$ to the adversary $\mathcal{A}$.
- PS queries: $\mathcal{A}$ can ask the publicly verifiable signature σ_{PV} on the message m he chooses. In response, $\mathcal{C}$ runs PS algorithm to obtain the signature σ_{PV}. $\mathcal{C}$ then returns σ_{PV} to $\mathcal{A}$ as the answer.
- DS queries: $\mathcal{A}$ can ask the designated verifier signature σ_{DV} on the message m and under the verifier's public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ he chooses. In response, $\mathcal{C}$ runs PS algorithm firstly to obtain the publicly verifiable signature σ_{PV} if this signature does not exist, then runs DS algorithm to obtain the designated verifier signature σ_{PV}. $\mathcal{C}$ then returns σ_{DV} to $\mathcal{A}$ as the answer.
- DV queries: $\mathcal{A}$ can ask the designation verification result of the message/signature pair $(m, \sigma_{\mathsf{DV}})$ with the designated verifier's public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$. In response, $\mathcal{C}$ runs DV algorithm to return the decision $d \in \{Acc, Rej\}$ to $\mathcal{A}$.
- SK queries: $\mathcal{A}$ can request the secret key queries of the public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ he chooses. In response, $\mathcal{C}$ returns the corresponding secret key sk to $\mathcal{A}$.

We say $\mathcal{A}$ wins the game if $\mathcal{A}$ outputs a forged message/ signature pair$(m^*, \sigma^*_{\mathsf{DV}})$ with a public key $pk^* \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ after all the queries, such that:

1. $Acc \leftarrow \mathsf{DV}(cp, pk_s, sk^*, m^*, \sigma^*_{\mathsf{DV}})$.
2. m^* has never been submitted as one of the PS queries.
3. (m^*, pk^*) has never been submitted as one of the DS queries.
4. pk^* has never been submitted as one of the SK queries.

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

Definition 5 We say an attacker $\mathcal{A}^{CMA,\, CPKA}_{EUF,\, UDVS}$ can $(t, q_{\mathsf{PS}}, q_{\mathsf{DS}}, q_{\mathsf{DV}}, q_{\mathsf{SK}}, \varepsilon)$-breaks the **UDVS** scheme if $\mathcal{A}^{CMA,\, CPKA}_{EUF,\, UDVS}$ runs in time at most t, makes at most q_{PS} PS queries, q_{DS} DS queries, q_{DV} DV queries, q_{SK} SK queries and Succ $\mathcal{A}^{CMA,\, CPKA}_{EUF,\, UDVS}$ is at least ε.

Remark The unforgeability model defined here is not the strong unforgeability model in the sense of [4,15]. However, we note that this model is *practical* and *widely used* [8]. Moreover, as we shall show in Sect. 4, the strong unforgeability model is somehow undesirable in some situation.

Non-transferability privacy
Let **UDVS** = (CPG, SKG, VKG, PS, PV, DS, $\overline{\mathsf{DS}}$, DV, $\mathsf{P_{KR}}$) be a UDVS scheme. We define the *non-transferability* of the **UDVS** against adaptive chosen public key and chosen message distinguisher $\mathcal{D}^{CMA,\, CPKA}_{TRANS,\, UDVS}$. As explained in [11], the goal of *non-transferability privacy* is that the signature holder provides many designated verifier signature σ_{DV}'s on message m, designated to many verifier public keys of the attacker's choice, however, the attacker cannot use these σ_{DV}'s to convince a third party that the signer has signed on the message m. In order to make the property of non-transferability privacy clearer, we classify the model into two stages. In the first stage, the distinguisher $\mathcal{D}$ can submit PS, DS, $\overline{\mathsf{DS}}$, DV, SecretKey(SK) queries adaptively. At the end of the first stage, $\mathcal{D}$ can submit a challenge message m^* and the public key pk^* to the challenger. In response, the challenger will choose a random $coin \in \{0, 1\}$. If $coin = 1$, $\mathcal{C}$ runs DS algorithm and returns the signature $\sigma^*_{\mathsf{DV}} = \sigma_{\mathsf{DV}}$ to $\mathcal{D}$. Otherwise, $\mathcal{C}$ runs $\overline{\mathsf{DS}}$ algorithm and returns the signature $\sigma^*_{\mathsf{DV}} = \overline{\sigma_{\mathsf{DV}}}$ to $\mathcal{D}$. After receiving σ^*_{DV}, $\mathcal{D}$ still can submit PS, DS, $\overline{\mathsf{DS}}$, DV, SK queries in the second stage except that $\mathcal{D}$ cannot submit m^* as one of PS queries or he cannot submit (m^*, pk^*) as one of the DS queries or $\overline{\mathsf{DS}}$ queries. At last, $\mathcal{D}$ outputs his guess of $coin$. Compared with the models defined in [11,15], we allow the distinguisher to obtain the designated verifier signatures of the challenging message which is designated to other verifiers except the challenging verifier.

- Setup: The challenger $\mathcal{C}$ runs the CPG algorithm to obtain the common parameters cp. $\mathcal{C}$ also generates $Signer's$ secret/public key-pair (sk_s, pk_s) from the SKG. Additionally, $\mathcal{C}$ runs VKG some times to obtain n potential $Verifier's$ secret/public key-pairs (sk_{v_i}, pk_{v_i}). $\mathcal{C}$ then sends the common parameters cp, $Signer's$ public key pk_s and all $Verifier's$ public keys $pk_{v_i}, i \in \{1, 2, \ldots, n\}$ to the distinguisher $\mathcal{D}$.
- Stage 1
    - PS queries: $\mathcal{D}$ can ask the publicly verifiable signature σ_{PV} on the message m he chooses. In response, $\mathcal{C}$ runs PS algorithm to obtain the signature σ_{PV}. $\mathcal{C}$ then returns σ_{PV} to $\mathcal{D}$ as the answer.
    - DS queries: $\mathcal{D}$ can ask the designated verifier signature σ_{DV} on the message m and under the verifier's public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ he chooses. In response, $\mathcal{C}$ runs PS algorithm firstly to obtain the publicly verifiable signature σ_{PV} if this signature does

not exist, then runs DS algorithm to obtain the designated verifier signature σ_{PV}. $\mathcal{C}$ then returns σ_{DV} to $\mathcal{D}$ as the answer.

- $\overline{\mathsf{DS}}$ queries: $\mathcal{D}$ can ask the designated verifier signature $\overline{\sigma_{\mathsf{DV}}}$ on the message m and under the verifier's public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ he chooses. In response, $\mathcal{C}$ runs $\overline{\mathsf{DS}}$ to obtain the signature $\overline{\sigma_{\mathsf{PV}}}$ designated by the verifier. $\mathcal{C}$ then returns $\overline{\sigma_{\mathsf{DV}}}$ to $\mathcal{D}$ as the answer.
- DV queries: $\mathcal{A}$ can ask the designation verification result of the message/signature pair $(m, \sigma_{\mathsf{DV}})$ with the designated verifier's public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$. In response, $\mathcal{C}$ runs DV algorithm to return the decision $d \in \{Acc, Rej\}$ to $\mathcal{A}$.
- SK queries: $\mathcal{A}$ can request the secret key queries of the public key $pk \in \{pk_{v_1}, pk_{v_2}, \ldots, pk_{v_n}\}$ he chooses. In response, $\mathcal{C}$ returns corresponding secret key sk to $\mathcal{A}$.

- Challenge: Once $\mathcal{D}$ decides that Stage 1 is over, $\mathcal{D}$ outputs a message m^* and a Verifier pk^* such that (m^*, pk^*) has not been submitted as one of the PS queries, DS queries or $\overline{\mathsf{DS}}$ queries. Then the challenger $\mathcal{C}$ chooses a random $coin \in \{0, 1\}$. If $coin = 1$, $\mathcal{C}$ runs the algorithm DS and returns σ_{DV} to $\mathcal{D}$. Otherwise $coin = 0$, $\mathcal{C}$ runs the algorithm $\overline{\mathsf{DS}}$ and returns $\overline{\sigma_{DV}}$ to $\mathcal{D}$.
- Stage 2: Upon receiving the challenging message/signature pair from $\mathcal{C}$, $\mathcal{D}$ still can submit PS, DS, $\overline{\mathsf{DS}}$, DV, SK queries, except that
  1. He cannot submit m^* as one of PS queries.
  2. He cannot submit (m^*, pk^*) as one of the DS queries or $\overline{\mathsf{DS}}$ queries.
- Guess: Finally, $\mathcal{D}$ outputs his guess $coin'$ of $coin$. $\mathcal{D}$ wins the game if $coin' = coin$.

The advantage of an adaptive chosen message and public key distinguisher $\mathcal{D}$ has in the above game is defined as

$$\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS} = |\Pr[coin' = coin] - 1/2|.$$

Definition 6 We say a **UDVS** scheme is non-transferable against a $(t, q_{\mathsf{PS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}})$ adaptive chosen message and public key distinguisher $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ if Adv $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ is negligible after making at most q_{PS} PS queries, q_{DS} DS queries, $q_{\overline{\mathsf{DS}}}$ $\overline{\mathsf{DS}}$ queries, q_{DV} DV queries, q_{SK} SK queries in time t.

4 Analysis of UDVS scheme of Zhang et al. [15] without random oracle

Recently, Zhang et al. [15] proposed the first construction of the UDVS scheme without random oracles. In the scheme, they use BB [4] short signature scheme as the PS algorithm to obtain the UDVS without random oracle. Zhang et al. also refined the unforgeability definitions of UDVS such that the adversaries have more freedom to select target verifiers and target messages. Moreover, the notion "strong unforgeability" in the sense of [1] was firstly introduced to the UDVS. In this section, we will give a security analysis to the scheme of Zhang et al.

4.1 Review of UDVS scheme of Zhang et al. [15]

Here, we give a brief review of their scheme, please refer to [15] for more details. Let $\sigma_{\mathsf{PV}} = (\sigma, r)$ denote BB's signature of a message m from a signer whose public key is $pk_s = (u_1, v_1)$. Then the algorithms DS and $\overline{\mathsf{DS}}$ in [15] are:

1. DS: $\sigma_{\mathsf{DV}} = (\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2}, \sigma_{\mathsf{DV}_3}) = (\sigma, g^r, e(u_3, v_3^r))$, where u_3, v_3 is the public key of the designated verifier (Here we let $\mathbb{G}_1 = \mathbb{G}_2$ and the generator of $\mathbb{G}_1(\mathbb{G}_2)$ is g in their scheme for convenience.)
2. $\overline{\mathsf{DS}}$: The designated verifier himself can output a valid UDVS signature for himself using his secret key $sk_v = (x_3, y_3) \in \mathbb{Z}_p \times \mathbb{Z}_p$. For a message m, he chooses $s \in_R \mathbb{Z}_p$ and computes:

$$\sigma_{\mathsf{DV}_1} = g^s, \sigma_{\mathsf{DV}_2} = g^{1/s} u_1^{-1} v_1^{-m}, \quad \sigma_{\mathsf{DV}_3} = e(g, \sigma_{\mathsf{DV}_2})^{x_3 y_3}.$$

Remark Actually, Zhang et al. do not give the definition of the $\overline{\mathsf{DS}}$ directly; however, we can extract this algorithm from their proof of non-transferability privacy in the Theorem 3 [15].

3. DV: Given Signer's public key (u_1, v_1), $Verifier's$ secret key (x_3, y_3), the signed message m and the DV signature $\sigma_{\mathsf{DV}} = (\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2}, \sigma_{\mathsf{DV}_3})$, the verifier checks whether

$$e(\sigma_{\mathsf{DV}_1}, u_1 \sigma_{\mathsf{DV}_2} v_1^m) \overset{?}{=} e(g, g) \text{ and } \sigma_{\mathsf{DV}_3} \overset{?}{=} e(g, \sigma_{\mathsf{DV}_2})^{x_3 y_3}.$$

If all the two equations hold, he accepts the signature. Otherwise, rejects it.

4.2 Analysis of unforgeability

The unforgeability of the scheme of Zhang et al. is based on the Strong Diffie Hellman (SDH) problem [15]. However, as pointed out by Cheon very recently in [6], SDH-related assumption has some inherent drawbacks. To ensure the hardness of the SDH problem, Cheon suggested to increase the key size or use a prime p such that both of $p+1$ and $p-1$ have no small divisor greater than $(\log p)^2$ [6]. Unfortunately, the distribution of such primes is unknown. This is one of the reasons why we do not use BB signature as the underlying PS algorithm in our scheme proposed in this paper.

4.3 Analysis of the non-transferability

In this section, we will analyze the non-transferability of the scheme of Zhang et al. Our analysis shows that the distinguisher $\mathcal{D}$ can have non-negligible advantage in the model of the non-transferability defined in Sect. 3. However, it does not mean $\mathcal{D}$ could have the same advantage in the model defined in [15].

Suppose that $\mathcal{D}$ chooses verifier V_A as the target verifier and gets the challenging signature $\sigma_{(DV,V_A)}$ on message m^*. Then, as defined in the model, $\mathcal{D}$ can also choose another verifier V_B and submit (m^*, pk_{V_B}) as one of DS queries. Therefore, $\mathcal{D}$ will get another signature $\sigma_{(DV,V_B)}$ which is output by DS algorithm.

1. If $\sigma_{(\mathsf{DV},V_A)}$ is output by DS algorithm, the first two parts of these two signatures, $\sigma_{(\mathsf{DV},V_A)}$ and $\sigma_{(\mathsf{DV},V_B)}$, must be identical. Namely, $\sigma_{(\mathsf{DV}_1,V_A)} = \sigma_{(\mathsf{DV}_1,V_B)} = \sigma$ and $\sigma_{(\mathsf{DV}_2,V_A)} = \sigma_{(\mathsf{DV}_2,V_B)} = g^r$ where (σ, r) is BB's signature on the message m. Therefore
$$\Pr[\sigma_{(\mathsf{DV}_1,V_A)} = \sigma_{(\mathsf{DV}_1,V_B)} \wedge \sigma_{(\mathsf{DV}_2,V_A)} = \sigma_{(\mathsf{DV}_2,V_B)} | \sigma_{(\mathsf{DV},V_A)} \leftarrow \mathsf{DS}(pk_s, pk_{v_A}, m, \sigma, r)] = 1.$$
2. However, if $\sigma_{(\mathsf{DV},V_A)}$ is output by $\overline{\mathsf{DS}}$ algorithm, $\sigma_{(\mathsf{DV}_1,V_A)} = g^s$ and $\sigma_{\mathsf{DV}_2} = g^{1/s}u_1^{-1}v_1^{-m}$ where s is randomly chosen in $\mathbb{Z}_p$ and (u_1, v_1) is the public key of the signer. Therefore
$$\Pr[\sigma_{(\mathsf{DV}_1,V_A)} = \sigma_{(\mathsf{DV}_1,V_B)} \wedge \sigma_{(\mathsf{DV}_2,V_A)} = \sigma_{(\mathsf{DV}_2,V_B)} |\sigma_{(\mathsf{DV},V_A)} \leftarrow \overline{\mathsf{DS}}(pk_s, sk_{v_A}, m)] = 1/p,$$
which is negligible.

Therefore, the distinguisher $\mathcal{D}$ is only required to check the equality of the first two parts and will have non-negligible advantage in the game defined in the non-transferability model.

However, $\mathcal{D}$ could not have the same advantage in practice. If $\mathcal{D}$ has two signatures: $\sigma_{(\mathsf{DV},V_A)}$ and $\sigma_{(\mathsf{DV},V_B)}$ of the scheme of Zhang et al., $\mathcal{D}$ cannot be convinced that these two signatures are generated by DS algorithm. The reason is that these two verifiers, V_A and V_B, could cooperate and use the same random number s in $\overline{\mathsf{DS}}$ algorithm to generate $\sigma_{(\mathsf{DV},V_A)}$ and $\sigma_{(\mathsf{DV},V_B)}$, such that the first two parts of these two signatures are still identical. The model defined in [15] allows such a "cooperation" and therefore, the scheme of Zhang et al. still satisfies the notion of non-transferability defined in their paper. However, as we have shown, their scheme does not satisfy the non-transferability defined in this paper, where the "cooperation" is not allowed. Therefore, it is still worthwhile to construct a UDVS scheme which is also secure in our model.

In [15], Zhang et al. use BB signature scheme as the PS algorithm. BB scheme is strong unforgeable which means given a valid signature σ_{PV} of a message m, one cannot output another signature σ'_{PV} such that σ'_{PV} is a valid signature on the message m while $\sigma'_{\mathsf{PV}} \neq \sigma_{\mathsf{PV}}$. Therefore, the signature holder must designate the same signature of the message to different verifiers. This is the reason why the first two parts of the UDVS in their scheme [15] are *identical* and thus do not satisfy the non-transferability privacy property in the game defined in Sect. 3.

If the PS algorithm is not a strong unforgeable scheme but is unforgeable against chosen message attack in the sense of [8], then given a valid signature σ_{PV} of the message m, the signature holder can create many different valid signatures σ'_{PV} of the same message m. Therefore, the signature holder can use different PV signature σ_{PV} to create different σ_{DV} on the same message m and designated to a different verifier. We will show in Sect. 5 how to use this property to overcome the weakness in the scheme of Zhang et al. [15] to ensure that the non-transferable privacy of the UDVS is provided.

5 Secure universal designated verifier signature without random oracle

In this section, we incorporate Waters' signature scheme [14] to obtain a concrete secure UDVS scheme without random oracle. We also provide the formal security analysis of the proposed scheme. Details of Waters' signature scheme are provided in the Appendix.

5.1 The proposed scheme

1. CPG: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$ for some prime p, g is the generator of $\mathbb{G}_1$. e denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. The messages m to be signed in this scheme will be represented as bitstrings of length n, a separate parameter unrelated to p. Furthermore, picks $n+1$ random elements $u', u_1, u_2, \ldots, u_n \in_R \mathbb{G}_1$ and set $\mathbf{u} = (u_1, u_2, \ldots, u_n)$. Then the common parameter $cp = (\mathbb{G}_1, \mathbb{G}_T, p, g, e, n, u', \mathbf{u})$.
2. SKG: The $Signer$ picks two secret values $x_s, y_s \in_R \mathbb{Z}_p^*$ and sets the secret key $sk = (x_s, y_s)$. Then the signer computes the public key $pk_s = (pk_{sx}, pk_{sy}) = (g^{x_s}, g^{y_s})$.
3. VKG: The $Verifier$ picks two secret values $x_v, y_v \in_R \mathbb{Z}_p^*$ and sets the secret key $sk = (x_v, y_v)$. Then the

signer computes the public key $pk_v = (pk_{vx}, pk_{vy}) = (g^{x_v}, g^{y_v})$.

4. PS: Let m be an n-bit message to be signed by the signer, m_i denote the ith bit of m, and $\mathcal{M} \in \{1, \ldots, n\}$ be the set of all i for which $m_i = 1$, a signature is generated as follows. First, a random $r \in \mathbb{Z}_p$ is chosen. Then the signature is constructed as: $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2}) = (g^{x_s y_s}(u' \prod_{i\in\mathcal{M}} u_i)^r, g^r)$.
5. PV: Suppose we wish to check whether $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2})$ is a signature for a message M. The signature is accepted if $e(\sigma_{\mathsf{PV}_1}, g)/e(u' \prod_{i\in\mathcal{M}} u_i, \sigma_{\mathsf{PV}_2}) = e(pk_{sx}, pk_{sy})$ holds.
6. DS: Given the designated verifier's public key $(pk_v = (pk_{vx}, pk_{vy}))$, the signature holder selects $r' \in_R \mathbb{Z}_p$ and computes

$$\sigma_{\mathsf{DV}_1} = e\left(\sigma_{\mathsf{PV}_1} \cdot \left(u' \prod_{i\in\mathcal{M}} u_i\right)^{r'}, pk_{vx}\right) = e\left(g^{x_s y_s} \cdot \left(u' \prod_{i\in\mathcal{M}} u_i\right)^{r+r'}, pk_{vx}\right)$$

and $\sigma_{\mathsf{DV}_2} = \sigma_{\mathsf{PV}_2} \cdot g^{r'} = g^{r+r'}$. Then, the signature holder sends $\sigma_{\mathsf{DV}} = (\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2})$ to the designated verifier.

7. $\overline{\mathsf{DS}}$: The designated verifier can also produce a valid signature on any message m'. He only needs to select a random $r' \in Z_p$ and computes

$$\overline{\sigma_{\mathsf{DV}_2}} = g^{r'} \text{ and } \overline{\sigma_{\mathsf{DV}_1}} = e(pk_{sx}, pk_{sy})^{x_v} e\left(u' \prod_{i\in\mathcal{M}'} u_i, \sigma_{\mathsf{DV}_2}\right)^{x_v}.$$

8. DV: Given the signer's public key $pk_s = (pk_{sx}, pk_{sy})$, a message m, and a signature $(\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2})$, verify that $\sigma_{\mathsf{DV}_1} = e(pk_{sx}, pk_{sy})^{x_v} e(u' \prod_{i\in\mathcal{M}} u_i, \sigma_{\mathsf{DV}_2})^{x_v}$. If the equality holds, the result is Acc; otherwise the result is Rej.

Consistence:

1. PV **Consistency**: If the publicly verifiable signature $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2})$ of the message m is generated by the PS algorithm, then

$$\frac{e(\sigma_{\mathsf{PV}_1}, g)}{e(u' \prod_{i\in\mathcal{M}} u_i, \sigma_{\mathsf{PV}_2})} = \frac{e(g^{x_s y_s}(u' \prod_{i\in\mathcal{M}} u_i)^r, g)}{e(u' \prod_{i\in\mathcal{M}} u_i, g^r)} = e(g^{x_s y_s}, g) = e(pk_{sx}, pk_{sy}).$$

Therefore $\Pr[\mathsf{PV}(cp, pk_s, m, \mathsf{PS}(cp, sk_s, m)) = Acc] = 1$.

2. DV **Consistency of** DS: If the designated verifier signature $\sigma_{\mathsf{DV}} = (\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2})$ is generated by the DS algorithm, then

$$\begin{aligned}\sigma_{\mathsf{DV}_1} &= e\left(\sigma_{\mathsf{PV}_1} \cdot \left(u \prod_{i\in\mathcal{M}} u_i\right)^{r'}, pk_{vx}\right)\\ &= e\left(g^{x_s y_s}\left(u' \prod_{i\in\mathcal{M}} u_i\right)^{r}\left(u' \prod_{i\in\mathcal{M}} u_i\right)^{r'}, g^{x_v}\right)\\ &= e\left(g^{x_s y_s}, g^{x_v}\right) e\left(\left(u' \prod_{i\in\mathcal{M}} u_i\right)^{r+r'}, g^{x_v}\right)\\ &= e\left(pk_{sx}, pk_{sy}\right)^{x_v} e\left(u' \prod_{i\in\mathcal{M}} u_i, g^{r+r'}\right)^{x_v}\\ &= e\left(pk_{sx}, pk_{sy}\right)^{x_v} e\left(u' \prod_{i\in\mathcal{M}} u_i, \sigma_{\mathsf{DV}_2}\right)^{x_v}.\end{aligned}$$

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

3. DV **Consistency of** $\overline{\mathsf{DS}}$: If the designated verifier signature $\overline{\sigma_{\mathsf{DV}}} = (\overline{\sigma_{\mathsf{DV}_1}}, \overline{\sigma_{\mathsf{DV}_2}})$ is generated by the $\overline{\mathsf{DS}}$ algorithm, then

$$\overline{\sigma_{\mathsf{DV}_1}} = e(pk_{sx}, pk_{sy})^{x_v} e\left(u' \prod_{i\in\mathcal{M}'} u_i, \sigma_{\mathsf{PV}_2}\right)^{x_v}.$$

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

5.2 Unforgeability

Theorem 1 *If there is an adaptively chosen message and public key attacker $\mathcal{A}^{CMA,\ CPKA}_{EUF,\ UDVS}$ who can $(t, q_{\mathsf{PS}}, q_{\mathsf{DS}}, q_{\mathsf{DV}}, q_{\mathsf{SK}}, \varepsilon)$ break the proposed* **UDVS** *scheme, then there exists an algorithm $\mathcal{B}$ who can solve the GBDH problem in $(\mathbb{G}_1, \mathbb{G}_T)$ with probability:*

$$Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T} \geq \frac{\varepsilon}{8q_{\mathsf{SK}}(n+1)(q_{\mathsf{PS}} + q_{\mathsf{DS}} + q_{\mathsf{DV}})}\left(1 - \frac{1}{q_{\mathsf{SK}}+1}\right)^{q_{\mathsf{SK}}+1}.$$

in time $t' \leq t + 7n(q_{\mathsf{PS}} + q_{\mathsf{DS}} + q_{\mathsf{DV}})\rho_{\mathbb{G}_1} + (4q_{\mathsf{PS}} + 6q_{\mathsf{DS}})\tau_{\mathbb{G}_1} + (q_{\mathsf{DV}} + 1)\rho_{\mathbb{G}_T} + (q_{\mathsf{DS}} + 2q_{\mathsf{DV}} + 1)\varrho$ where $\rho_{\mathbb{G}_1}, \rho_{\mathbb{G}_T}$ are the time for a multiplication in $\mathbb{G}_1, \mathbb{G}_T$ respectively, $\tau_{\mathbb{G}_1}$ is the time for an exponentiation in $\mathbb{G}_1$ and ϱ is the time for pairing in $(\mathbb{G}_1, \mathbb{G}_T)$.

Proof See Appendix.

5.3 Non-transferability

Theorem 2 *The proposed* **UDVS** *scheme is non-transferable against a* $(t, q_{\mathsf{PS}}, q_{\mathsf{DS}}, q_{\overline{\mathsf{DS}}}, q_{\mathsf{DV}}, q_{\mathsf{SK}})$ *adaptive chosen message and public key distinguisher* $\mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$.

Proof See Appendix.

5.4 Delegatability

Non-delegatability is a stronger notion of the designated verifier signature schemes which is proposed by Lipmaa et al. [10]. Non-delegatability means that there exists an efficient knowledge extractor that can extract either the Signer's secret key or the designated verifier's secret key, when given oracle access to an adversary who can create valid signatures with a high probability. The proposed UDVS scheme in this paper does not satisfy this property because anyone who has the knowledge of the *trapdoor*: $e(pk_{sx}, pk_{sy})^{sk_{vx}}$ can compute a valid signature designated to a verifier V. Moreover, we note that to date, there is *no* known UDVS can satisfy this property in the standard model. However, we note that the ring signature scheme recently proposed in [3] might be used to construct a non-delegatable UDVS scheme without random oracles.

6 Conclusion

In this paper, we gave a security analysis to the universal designated verifier signature scheme without random oracle proposed in [15]. Then we constructed a new UDVS scheme without random oracle based on Waters' signature scheme proposed in [14]. We showed that a signature scheme which is unforgeable against chosen message attack in the sense of [8] but not strong unforgeable in the sense of [4] might be more suitable to construct a UDVS scheme. The new proposed scheme satisfies the privacy property of the UDVS and is unforgeable against an adaptively chosen message and chosen public key attacker based on the Gap Bilinear Diffie Hellman assumption.

Appendix

Waters' Signature Scheme [14]

1. CPG: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$ for some prime p, g is the generator of $\mathbb{G}_1$. e denotes the bilinear pairing $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$. The messages m to be signed in this scheme will be represented as bitstrings of length n, a separate parameter unrelated to p. Furthermore, picks $n+1$ random elements $u', u_1, u_2, \ldots, u_n \in_R \mathbb{G}_1$ and set $\mathbf{u} = (u_1, u_2, \ldots, u_n)$. Then the common parameter
$$cp = (\mathbb{G}_1, \mathbb{G}_T, p, g, e, n, u', \mathbf{u}).$$
2. SKG: The *Signer* picks two secret values $x_s, y_s \in_R \mathbb{Z}_p^*$ and sets the secret key $sk = (x_s, y_s)$. Then the signer computes the public key $pk_s = (pk_{sx}, pk_{sy}) = (g^{x_s}, g^{y_s})$.
3. PS: Let m be an n-bit message to be signed by the original signer Alice and m_i denote the ith bit of m, and $\mathcal{M} \in \{1, \ldots, n\}$ be the set of all i for which $m_i = 1$, a signature is generated as follows. First, a random $r \in \mathbb{Z}_p$ is chosen. Then the signature is constructed as:
$$\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2}) = \left(g^{x_s y_s}\left(u' \prod_{i \in \mathcal{M}} u_i\right)^r, g^r\right)$$
4. PV: Suppose we wish to check whether $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2})$ is a signature for a message M. The signature is accepted if
$$e(\sigma_{\mathsf{PV}_1}, g)/e\left(u' \prod_{i \in \mathcal{M}} u_i, \sigma_{\mathsf{PV}_2}\right) = e\left(pk_{sx}, pk_{sy}\right).$$

Given a valid Waters' signature $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2}) = (g^{x_s y_s}(u' \prod_{i \in \mathcal{M}} u_i)^r, g^r)$ of the message m, the signature holder can choose $r' \in_R \mathbb{Z}_p$ and obtain another valid signature σ'_{PV} on the same message m.

$$\begin{aligned}\sigma'_{\mathsf{PV}} &= \left(\sigma'_{\mathsf{PV}_1}, \sigma'_{\mathsf{PV}_2}\right)\\ &= \left(\sigma_{\mathsf{PV}_1} \cdot \left(u' \prod_{i \in \mathcal{M}} u_i\right)^{r'}, \sigma_{\mathsf{PV}_2} \cdot g^{r'}\right)\\ &= \left(g^{x_s y_s}\left(u' \prod_{i \in \mathcal{M}} u_i\right)^{r+r'}, \sigma_{\mathsf{PV}_2} \cdot g^{r+r'}\right)\end{aligned}$$

Proof of Theorem 1 Suppose there exists an attacker $\mathcal{A}$ who can $(t, q_{\mathsf{PS}}, q_{\mathsf{DS}}, q_{\mathsf{DV}}, q_{\mathsf{SK}}, \varepsilon)$ break our proposed UDVS scheme. We will construct an algorithm $\mathcal{B}$ which will use $\mathcal{A}$ to solve the Gap BDH problem. $\mathcal{B}$ will take Gap BDH challenge (g, g^a, g^b, g^c) of a bilinear group $(\mathbb{G}_1, \mathbb{G}_T)$ whose orders are both a prime p and output $e(g, g)^{abc}$ with the help of the oracle $\mathcal{O}_{DBDH}$. $\mathcal{B}$ will response $\mathcal{A}$'s queries as following.

– Setup: $\mathcal{B}$ sets an integer $\ell = 4(q_{\mathsf{PS}} + q_{\mathsf{DS}} + q_{\mathsf{DV}})$, and chooses an integer, k, uniformly at random between 0 and

n. It then chooses a value x' and a random n-vector, $\mathbf{x} = (x_i)$ where $x', x_i \in_R \mathbb{Z}_\ell$. Additionally, $\mathcal{B}$ chooses a value y' and a random n-vector $\mathbf{y} = (y_i)$ where $y', y_i \in_R \mathbb{Z}_p$. $\mathcal{B}$ keeps all the values secret.

For a message m, we let $\mathcal{M} \subseteq \{1, 2, \ldots, n\}$ be the set of all i for which $m_i = 1$. To make the notation easy to follow, we define three functions $F(m)$, $J(m)$ and $K(m)$ as [14]:

1. $F(m) = (p - \ell k) + x' + \Sigma_{i \in \mathcal{M}} x_i$
2. $J(m) = y' + \Sigma_{i \in \mathcal{M}} y_i$
3. $K(m) = \begin{cases} 0, & \text{if } x' + \Sigma_{i\in\mathcal{M}} x_i \equiv 0 \pmod{\ell} \\ 1, & \text{otherwise} \end{cases}$

$\mathcal{B}$ sets the public keys of the users and the common parameter as:

1. Firstly, $\mathcal{B}$ assigns the signer's public key $(pk_{sx}, pk_{sy}) = (g^a, g^b)$ where g^a, g^b are the inputs of the Gap BDH problem.
2. Then $\mathcal{B}$ maintains a list L to record all the secret/public key-pairs of the verifiers. To generate the ith verifier V_i's secret/public key pair, $\mathcal{B}$ chooses a random coin $c_i \in \{0, 1\}$ such that $\Pr[c_i = 1] = \delta$(the value of the δ will be determined later).
   - If $c_i = 0$, $\mathcal{B}$ chooses two random numbers $d_i, e_i \in \mathbb{Z}_p$ and computes
     $pk_{v_i} = (pk_{v_i x}, pk_{v_i y}) = (g^{d_i}, g^{e_i})$.
     Then $\mathcal{B}$ adds $(pk_{v_i x}, pk_{v_i y}, c_i, d_i, e_i)$ to the List L.
   - Else $c_i = 1$, $\mathcal{B}$ chooses two random numbers $d_i, e_i \in \mathbb{Z}_p$ and computes
     $pk_{v_i} = (pk_{v_i x}, pk_{v_i y}) = ((g^c)^{d_i}, (g^c)^{e_i})$
     where g^c is the input of the Gap BDH problem. $\mathcal{B}$ then adds $(pk_{v_i x}, pk_{v_i y}, c_i, \perp, \perp)$ to the List L. Here the notation $\perp$ means $\mathcal{B}$ does not know the corresponding value.
3. $\mathcal{B}$ then assigns $u' = pk_{sy}^{p-k\ell+x'} g^{y'}$ and $u_i = pk_{sy}^{x_i} g^{y_i}$ and sets $\mathbf{u} = (u_1, u_2, \ldots, u_n)$

$\mathcal{B}$ returns Signer's public key pk_s, all Verifiers' public keys pk_{v_i}, common parameter $cp = (\mathbb{G}_1, \mathbb{G}_T, p, g, e, n, u', \mathbf{u})$ to $\mathcal{A}$. From the perspective of the adversary all the distributions are identical to the real construction.

- PS queries: Suppose $\mathcal{A}$ issues an PS queries for the message m. If $K(m) \neq 0$(If we have $K(m) \neq 0$ this implies $F(m) \neq 0 \pmod{p}$, since we can assume $p > n\ell$ for any reasonable values of p, n, and ℓ [14]), $\mathcal{B}$ can construct the public verifiable signature by choosing a random $r \in \mathbb{Z}_p$ and compute:

$$\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2}) = \left(pk_{sx}^{\frac{-J(m)}{F(m)}} \left(u' \prod_{i\in\mathcal{M}} u_i \right)^r, pk_{sx}^{\frac{-1}{F(m)}} g^r \right).$$

Correctness

$$\begin{aligned} \sigma_{\mathsf{PV}_1} &= pk_{sx}^{\frac{-J(m)}{F(m)}} \left(u' \prod_{i\in\mathcal{M}} u_i \right)^r \\ &= pk_{sx}^{\frac{-J(m)}{F(m)}} (pk_{sy}^{F(m)} g^{J(m)})^r \\ &= pk_{sy}^a (pk_{sy}^{F(m)} g^{J(m)})^{\frac{-a}{F(m)}} (pk_{sy}^{F(m)} g^{J(m)})^r \\ &= pk_{sy}^a (pk_{sy}^{F(m)} g^{J(m)})^{r - \frac{a}{F(m)}} \\ &= pk_{sy}^a (pk_{sy}^{F(m)} g^{J(m)})^{\tilde{r}} = pk_{sy}^a \left(u' \prod_{i\in\mathcal{M}} u_i \right)^{\tilde{r}}. \end{aligned}$$

Note that: $\sigma_{\mathsf{PV}_2} = pk_{sx}^{\frac{-1}{F(m)}} g^r = g^{\frac{-a}{F(m)}} g^r = g^{r - \frac{-a}{F(m)}} = g^{\tilde{r}}$.

Otherwise, $K(m) = 0$. $\mathcal{B}$ terminates the simulation and reports failure.

- DS queries: Suppose $\mathcal{A}$ issues a DS query for a message m and the designated verifier pk_{v_i}. If $K(m) \neq 0$, $\mathcal{B}$ can obtain the publicly verifiable signature $\sigma_{\mathsf{PV}} = (\sigma_{\mathsf{PV}_1}, \sigma_{\mathsf{PV}_2})$ as above. Then $\mathcal{B}$ chooses a random $r' \in \mathbb{Z}_p$ and computes the designated verifier signature as

$$\sigma_{\mathsf{DV}_1} = e\left(\sigma_{\mathsf{PV}_1} \left(u' \prod_{i\in\mathcal{M}} u_i \right)^{r'}, pk_{v_i x} \right), \quad \sigma_{\mathsf{DV}_2} = \sigma_{\mathsf{PV}_2} g^{r'}.$$

and sends $(\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2})$ to $\mathcal{A}$ as the answer. Otherwise, $K(m) = 0$ and $\mathcal{B}$ terminates the simulation and reports failure.

- DV queries: Suppose $\mathcal{A}$ issues a DV queries for the message/signature pair $(m, \sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2})$ and the designated verifier whose public key is $pk_{v_i} = (pk_{v_i x}, pk_{v_i y})$.
  1. If $K(m) \neq 0$, $\mathcal{B}$ can compute a valid universal designated verifier signature on this message as he responses to DS queries. Let $(\sigma'_{\mathsf{DV}_1}, \sigma'_{\mathsf{DV}_2})$ denote the signature computes by $\mathcal{B}$. Then $\mathcal{B}$ submits

$$\left(g, u' \prod_{i\in\mathcal{M}} u_i, pk_{v_i x}, \frac{\sigma_{\mathsf{DV}_2}}{\sigma'_{\mathsf{DV}_2}}, \frac{\sigma_{\mathsf{DV}_1}}{\sigma'_{\mathsf{DV}_1}} \right).$$

  to the DBDH oracle $\mathcal{O}_{DBDH}$. $\mathcal{B}$ outputs Acc to $\mathcal{A}$ if the above tuple is a valid BDH tuple, otherwise outputs Rej.

  Correctness:

  If $(\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2} = g^r)$ is a valid signature, then

$$\sigma_{\mathsf{DV}_1} = e(g^{ab}, pk_{v_i x}) e\left(u' \prod_{i\in\mathcal{M}} u_i, pk_{v_i x} \right)^r.$$

Similarly, since $(\sigma'_{\mathsf{DV}_1}, \sigma'_{\mathsf{DV}_2} = g^{r'})$ is another valid signature produced by $\mathcal{B}$, then

$$\sigma'_{\mathsf{DV}_1} = e(g^{ab}, pk_{v_i x}) e\left(u' \prod_{i \in \mathcal{M}} u_i, pk_{v_i x}\right)^{r'}$$

Therefore,

$$\frac{\sigma_{\mathsf{DV}_2}}{\sigma'_{\mathsf{DV}_2}} = g^{r-r'} \text{ and } \frac{\sigma_{\mathsf{DV}_1}}{\sigma'_{\mathsf{DV}_1}} = e\left(u' \prod_{i \in \mathcal{M}} u_i, pk_{v_i x}\right)^{r-r'}.$$

which denotes that

$$\left(g, u' \prod_{i \in \mathcal{M}} u_i, pk_{v_i x}, \frac{\sigma_{\mathsf{DV}_2}}{\sigma'_{\mathsf{DV}_2}}, \frac{\sigma_{\mathsf{DV}_1}}{\sigma'_{\mathsf{DV}_1}}\right)$$

is a valid BDH tuple.

2. Else $K(m) = 0$ and $F(m) = 0$, $\mathcal{B}$ submits $\left(g, pk_{sx}, pk_{sy}, pk_{v_i x}, \frac{\sigma_{DV_1}}{e(pk_{v_i x}, \sigma_{DV_2})^{J(m)}}\right)$. to the DBDH oracle $\mathcal{O}_{DBDH}$. $\mathcal{B}$ outputs Acc to $\mathcal{A}$ if the above tuple is a valid BDH tuple, otherwise outputs Rej.

Correctness:

If $(\sigma_{\mathsf{DV}_1}, \sigma_{\mathsf{DV}_2} = g^r)$ is a valid signature, then

$$\sigma_{DV_1} = e(g^{ab}, pk_{v_i x}) e\left(u' \prod_{i \in \mathcal{M}} u_i, pk_{v_i x}\right)^{r}.$$

Note that $F(m) = 0$, which means

$$u' \prod_{i \in \mathcal{M}} u_i = g^{J(m)}.$$

Therefore,

$$\sigma_{DV_1} = e(g^{ab}, pk_{v_i x}) e(\sigma_{\mathsf{DV}_2}, pk_{v_i x})^{J(m)}.$$

which means

$$\left(g, pk_{sx}, pk_{sy}, pk_{v_i x}, \frac{\sigma_{DV_1}}{e(\sigma_{DV_2}, pk_{v_i x})^{J(m)}}\right).$$

is a valid BDH tuple.

3. Otherwise, $K(m) = 0$ and $F(m) \neq 0$, $\mathcal{B}$ terminates the simulation and reports failure.

– SK queries: Suppose $\mathcal{A}$ requests the secret key of the verifier V_i. In response, $\mathcal{B}$ firstly checks the tuple $(pk_{v_i x}, pk_{v_i y}, c_i, d_i, e_i)$ in the List L.
  1. If $c_i = 0$, which means $pk_{v_i x} = g^{d_i}$, $pk_{v_i y} = g^{e_i}$, $\mathcal{B}$ returns d_i, e_i to $\mathcal{A}$.
  2. Otherwise, $\mathcal{B}$ terminates the simulation and reports failure.

If $\mathcal{B}$ does not abort during the simulation, $\mathcal{A}$ will output a valid universal designated verifier signature $(\sigma^*_{\mathsf{DV}_1}, \sigma^*_{\mathsf{DV}_2})$ under the message m^* and the designated verifier V^* with success probability ε.

1. If $F(m^*) \neq 0$, $\mathcal{B}$ will abort.
2. Else, $\mathcal{B}$ checks the $(pk_{v^*x}, pk_{v^*y}, c^*, d^*, e^*)$. If $c^* = 0$, $\mathcal{B}$ will abort.
3. Otherwise, $F(m^*) = 0$ and $c^* = 1$ which means $pk_{v^*x} = (g^c)^{d^*}$. $\mathcal{B}$ computes

$$\begin{aligned}
&\left(\frac{\sigma^*_{\mathsf{DV}_1}}{e(pk_{v^*x}, \sigma^*_{\mathsf{DV}_2})^{J(m^*)}}\right)^{(d^*)^{-1}} \\
&= \left(\frac{e(pk_{sx}, pk_{sy})^{cd^*} e(u' \prod_{i \in \mathcal{M}} u_i, \sigma_{\mathsf{PV}_2})^{cd^*}}{e(pk_{v^*x}, \sigma^*_{\mathsf{DV}_2})^{J(m^*)}}\right)^{(d^*)^{-1}} \\
&= \left(\frac{e(pk_{sx}, pk_{sy})^{cd^*} e((g^b)^{F(m^*)} g^{J(m^*)}, \sigma_{\mathsf{PV}_2})^{cd^*}}{e(00(g^c)^{d^*}, \sigma^*_{\mathsf{DV}_2})^{J(m^*)}}\right)^{(d^*)^{-1}} \\
&= \left(\frac{e(g^a, g^b)^{cd^*} e(g^{J(m^*)}, \sigma_{\mathsf{PV}_2}))^{cd^*}}{e(g^{cd^*}, \sigma^*_{\mathsf{DV}_2})^{J(m^*)}}\right)^{(d^*)^{-1}} \\
&= e(g, g)^{abc}.
\end{aligned}$$

This completes the description of the simulation. It remains to analyze the probability of $\mathcal{B}$ not aborting. $\mathcal{B}$ will not abort if all the following cases happen:

$$\begin{aligned}
&\mathsf{A} : \mathcal{B} \text{ does not abort during } \mathsf{PS}, \ \mathsf{DS} \text{ and } \mathsf{DV} \text{ queries} \\
&\mathsf{B} : c_i = 0 \ during \ \mathsf{SK} \text{ queries} \\
&\mathsf{C} : c^* = 1 \\
&\mathsf{D} : F(m^*) = 0 \pmod p.
\end{aligned}$$

The success probability is $Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T}$=Pr$[A \wedge B \wedge C \wedge D]\varepsilon$.

Clearly, Case B and C are independent with other cases. Therefore

$$\begin{aligned}
Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T} &= \Pr[B \wedge C] \Pr[A \wedge D] \\
&= \delta(1-\delta)^{q_{\mathsf{SK}}} \Pr[A \wedge D]\varepsilon.
\end{aligned}$$

We can optimize this equation by setting $\delta = \frac{1}{q_{\mathsf{SK}}+1}$, then

$$Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T} = \frac{1}{q_{\mathsf{SK}}}\left(1 - \frac{1}{q_{\mathsf{SK}}+1}\right)^{q_{\mathsf{SK}}+1} \Pr[A \wedge D]\varepsilon.$$

Let $q_{\mathsf{PS}} + q_{\mathsf{DS}} + q_{\mathsf{DV}} = q$, then

$$\begin{aligned}
&\Pr[A \wedge D] = \Pr[A]\Pr[D|A] \\
&\quad \geq \Pr\left[\bigwedge_{i=1}^{q} K(m_i) \neq 0\right] \Pr\left[x + \sum_{i \in \mathcal{M}^*} x_i = \ell k | A\right] \\
&\quad = \left(1 - \Pr\left[\bigvee_{i=1}^{q} K(m_i) = 0\right]\right) \Pr\left[x + \sum_{i \in \mathcal{M}^*} x_i = \ell k | A\right]
\end{aligned}$$

$$\geq \left(1-\frac{q}{\ell}\right)\Pr\left[x+\sum_{i\in\mathcal{M}^*}x_i=\ell k|A\right]$$
$$=\frac{1}{n+1}\left(1-\frac{q}{\ell}\right)\Pr[K(m^*)=0|A]$$
$$=\frac{1}{n+1}\left(1-\frac{q}{\ell}\right)\frac{\Pr[K(m^*)=0]}{\Pr[A]}\Pr[A|K(m^*)=0]$$
$$\geq\frac{1}{(n+1)\ell}\left(1-\frac{q}{\ell}\right)\Pr[A|K(m^*)=0]$$
$$\geq\frac{1}{(n+1)\ell}(1-\frac{q}{\ell})\left(1-\Pr\left[\bigvee_{i=1}^{q}K(m_i)=0|K(m^*)=0\right]\right)$$
$$=\frac{1}{(n+1)\ell}\left(1-\frac{q}{\ell}\right)^2\geq\frac{1}{(n+1)\ell}\left(1-\frac{2q}{\ell}\right).$$

Therefore,

$$Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T}\geq\frac{1}{q_{\mathsf{SK}}}\left(1-\frac{1}{q_{\mathsf{SK}}+1}\right)^{q_{\mathsf{SK}}+1}\frac{1}{(n+1)\ell}\left(1-\frac{2q}{\ell}\right)\varepsilon.$$

We can optimize it by setting $\ell=4q=4(q_{\mathsf{PS}}+q_{\mathsf{DS}}+q_{\mathsf{DV}})$, then

$$Succ^{GBDH}_{\mathcal{B},\mathbb{G}_1,\mathbb{G}_T}\geq \frac{1}{8q_{\mathsf{SK}}(n+1)(q_{\mathsf{PS}}+q_{\mathsf{DS}}+q_{\mathsf{DV}})}\left(1-\frac{1}{q_{\mathsf{SK}}+1}\right)^{q_{\mathsf{SK}}+1}\varepsilon.$$

Proof of Theorem 2

- Setup: $\mathcal{C}$ sets the public keys of the users and the common parameters as:
  1. Firstly, $\mathcal{C}$ assigns the signer's public key $pk_s=(pk_{sx}, pk_{sy})=(g^{x_s},g^{y_s})$ where x_s, y_s are randomly chosen in $\mathbb{Z}_p$.
  2. Then $\mathcal{C}$ maintains a list L to record all the secret/public key-pairs of the verifiers. To generate the i^{th} verifier V_i's secret/public key pair, $\mathcal{C}$ chooses two random numbers $d_i, e_i\in\mathbb{Z}_p$ and computes $pk_{v_i}=(pk_{v_ix}, pk_{v_iy})=(g^{d_i},g^{e_i})$. Then $\mathcal{C}$ adds $(pk_{v_ix}, pk_{v_iy}, d_i, e_i)$ to the list L.
  3. $\mathcal{C}$ then chooses $u', u_i\in\mathbb{G}_1$ and sets $\mathbf{u}=(u_1,u_2,\ldots,u_n)$

  $\mathcal{C}$ returns $Signer's$ public key pk_s, all $Verifiers'$ public keys pk_{v_i}, common parameter $cp=(\mathbb{G}_1,\mathbb{G}_T, p, g, e, n, u', \mathbf{u})$ to $\mathcal{D}$. From the perspective of the adversary all the distributions are identical to the real construction.
- Stage 1:
  - Since $\mathcal{C}$ knows the secret keys of the signers and the verifiers, he can run PS algorithm, DS algorithm, $\overline{\mathsf{DS}}$ algorithm and DV algorithm to response PS queries, DS queries, $\overline{\mathsf{DS}}$ queries and DV queries, respectively.
  - SK queries: Suppose $\mathcal{D}$ requests the secret key of the verifier V_i. $\mathcal{C}$ firstly checks the list L to find the corresponding tuple $(pk_{v_ix}, pk_{v_iy}, d_i, e_i)$ in the list L and returns the corresponding secret key to $\mathcal{D}$.
- Challenge: At the end of Stage 1, $\mathcal{D}$ chooses a message m^* such that (m^*, V^*) has not been submitted as one of the DS queries or $\overline{\mathsf{DS}}$. Then the challenger $\mathcal{C}$ chooses a random coin $coin\in\{0,1\}$. If $coin=1$, $\mathcal{C}$ returns DS and sets $\sigma^*_{\mathsf{DV}}=\sigma_{\mathsf{DV}}$. Otherwise $coin=0$, $\mathcal{C}$ runs $\overline{\mathsf{DS}}$ and set $\sigma^*_{\mathsf{DV}}=\overline{\mathsf{DV}}$. Then $\mathcal{C}$ returns σ^*_{DV} to $\mathcal{D}$.
- Stage 2: After receiving the challenging message signature pair from $\mathcal{C}$, $\mathcal{D}$ still can submit PS, DS, $\overline{\mathsf{DS}}$, DV, SK queries, except that he cannot submit (m^*, V^*) as one of the DS queries or $\overline{\mathsf{DS}}$ queries.

1. Firstly, we show that the distribution of σ_{DV} which is output by DS algorithm is uniform.
   In the DS algorithm, given the designated verifier's public key $(pk_v=(pk_{vx}, pk_{vy}))$, the signature holder chooses $r'\in_R\mathbb{Z}_p$ and computes
   $$\sigma_{\mathsf{DV}_1}=e\left(\sigma_{\mathsf{PV}_1}\cdot\left(u'\prod_{i\in\mathcal{M}}u_i\right)^{r'}, pk_{vx}\right)$$
   $$=e\left(g^{x_sy_s}\cdot\left(u'\prod_{i\in\mathcal{M}}u_i\right)^{r+r'}, pk_{vx}\right)$$
   and $\sigma_{\mathsf{DV}_2}=\sigma_{\mathsf{PV}_2}\cdot g^{r'}=g^{r+r'}$. Therefore the value r' randomize the designated verifier $\sigma_{\mathsf{DV}}=(\sigma_{\mathsf{DV}_1},\sigma_{\mathsf{DV}_2})$ and σ_{DV} is *independent* with other DV signatures which are designated to other verifiers. The problem that exists in the scheme of Zhang et al. [15] will not happen in our scheme.
2. Then, we show that the signature simulated by the algorithm $\overline{\mathsf{DS}}$ is indistinguishable from algorithm DS, i.e. the following distributions are identical:
   $$\sigma_{\mathsf{DV}}=(\sigma_{\mathsf{DV}_1},\sigma_{\mathsf{DV}_2}):\begin{cases}\sigma_{\mathsf{DV}_1}=e(g^{x_sy_s}(u'\prod_{i\in\mathcal{M}}u_i)^r,\\ \quad pk_{vx}), r\in\mathbb{Z}_p\\ \sigma_{\mathsf{DV}_2}=g^r, r\in\mathbb{Z}_p\end{cases}$$
   and
   $$\overline{\sigma_{\mathsf{DV}}}=(\overline{\sigma_{\mathsf{DV}_1}},\overline{\sigma_{\mathsf{DV}_2}}):\begin{cases}\overline{\sigma_{\mathsf{DV}_1}}=e(g^{x_sy_s}(u'\prod_{i\in\mathcal{M}}u_i)^{\overline{r}},\\ \quad pk_{vx}), \overline{r}\in\mathbb{Z}_p\\ \overline{\sigma_{\mathsf{DV}_2}}=g^{\overline{r}}, \overline{r}\in\mathbb{Z}_p\end{cases}$$

Therefore

$$\Pr[\sigma_{\mathsf{DV}} = \sigma^*_{\mathsf{DV}}] = \Pr\begin{bmatrix}\sigma_{\mathsf{DV}_1} = \sigma^*_{\mathsf{DV}_1}\\ \sigma_{\mathsf{DV}_2} = \sigma^*_{\mathsf{DV}_2}\end{bmatrix} = \Pr[r = r^*] = 1/p$$

and

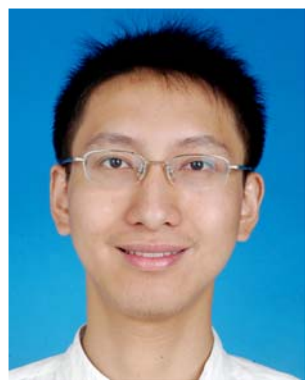
$$\Pr[\overline{\sigma_{\mathsf{DV}}} = \sigma^*_{\mathsf{DV}}] = \Pr\begin{bmatrix}\overline{\sigma_{\mathsf{DV}_1}} = \sigma^*_{\mathsf{DV}_1}\\ \overline{\sigma_{\mathsf{DV}_2}} = \sigma^*_{\mathsf{DV}_2}\end{bmatrix} = \Pr[\overline{r} = r^*] = 1/p,$$

which mean both distributions of probabilities are the same and $\mathsf{Adv}\ \mathcal{D}^{CMA,\ CPKA}_{TRANS,\ UDVS}$ is negligible.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. EUROCRYPT 2002. Lecture Notes in Computer Science, vol. 2332, pp. 83-107. Springer, Berlin (2002)
2. Baek, J., Safavi-Naini, R., Susilo, W.: Universal designated verifier signature proof (or how to efficiently prove knowledge of a signature). ASIACRYPT 2005. Lecture Notes in Computer Science, vol. 3788, pp. 644–661. Springer, Berlin (2005)
3. Bender, A., Katz, J., Morselli, R.: Ring signatures: stronger definitions, and constructions without random oracles. TCC 2006. Lecture Notes in Computer Science, vol. 3876, pp. 60–79. Springer, Berlin (2006)
4. Boneh, D., Boyen, X.: Short signatures without random oracles. EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 56–73. Springer, Berlin (2004)
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 514–532. Springer, Berlin (2001)
6. Cheon, J.H.: Security analysis of the strong diffie-hellman problem. EUROCRYPT (2006, to appear)
7. Diffie, W., Hellman, M.: New directions in cryptography. IEEE IT **22**, 644–654 (1976)
8. Goldwasser, S., Micali, S., Rivest, R.: A Digital signature scheme secure against adaptively chosen message attacks. SIAM J. Comput **17**(2), 281–308 (1988)
9. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. EUROCRYPT 1996. Lecture Notes in Computer Science, vol. 1070, pp. 143–154. Springer, Berlin (1996)
10. Lipmaa, H., Wang, G., Bao, F.: Designated verifier signature schemes: attacks, new securitynotions and a new construction. In: The 32nd International Colloquium on Automata, Languages and Programming ICALP 2005. Lecture Notes in Computer Science, vol. 3580, pp. 59–471. Springer, Berlin (2004)
11. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J.: Universal designated-verifier signatures. ASIACRYPT 2003. Lecture Notes in Computer Science, vol. 2894, pp. 523–543. Springer, Berlin (2003)
12. Steinfeld, R., Wang, H., Pieprzyk, J.: Efficient extension of standard schnorr/RSA Signatures into universal designated-verifier signatures. PKC 2004. Lecture Notes in Computer Science, vol. 2947, pp. 86–100. Springer, Berlin (2004)
13. Saeednia, S., Kramer, S., Markovitch, O.: An efficient strong designated verifier signature scheme. In: The 6th International Conference on Information Security and Cryptology (ICISC 2003). Lecture Notes in Computer Science, vol. 2971, pp. 40–54. Springer, Berlin (2003)
14. Waters, B.: Efficient identity-based encryption without random oracles. EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp.114–127. Springer, Berlin (2005)
15. Zhang, R., Furukawa, J., Imai, H.: Short signature and universal designated verifier signature without random oracles. Applied Cryptography and Network Security (ACNS 2005). Lecture Notes in Computer Science, vol. 3531, pp. 483–498. Springer, Berlin (2005)
16. Zhang, F., Susilo, W., Mu, Y., Chen, X.: Identity-Based universal designated verifier signatures. In: The First International Workshop on Security in Ubiquitous Computing Systems (SecUbiq 2005). Lecture Notes in Computer Science, vol. 3823, pp. 825–834. Springer, Berlin (2005)

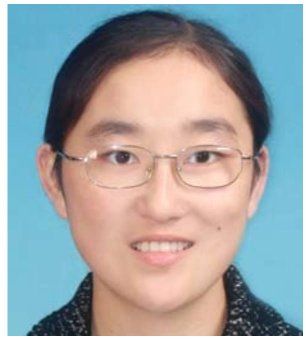
Author's biography



Xinyi Huang received his Bachelor and Master degree from Nanjing Normal University, China. He is currently a PhD candidate at the School of Computer Science and Software Engineering, University of Wollongong, Australia. His research interest is about cryptography, in particular digital signatures with limited verifiability.

Willy Susilo received a PhD in Computer Science from University of Wollongong, Australia. He is currently an Associate Professor at the School of Computer Science and Software Engineering. He is the director of Centre for Computer and Information Security Research (CCISR) at the University of Wollongong. His research interests include cryptography, information security, computer security and network security. His main contribution is in the area of digital signature schemes, in particular fail-stop signature schemes and short signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes.

Yi Mu received his PhD from the Australian National University in 1994. He currently is an associate professor in School of Computer Science and Software Engineering, University of Wollongong. Prior to joining University of Wollongong, he worked as a lecturer in the School of Computing and IT, the University of Western Sydney, and later as a senior lecturer in the Department of Computing, Macquarie University. Yi Mu is the director of Centre for Computer and Information Secure Research (CCISR) at University of Wollongong. His current research interests include network security, computer security, and cryptography. He is an editor-in-chief of International Journal of Applied Cryptography and an editor for other six international journals. He has served in program committees for a number of international conferences. He is a senior member of the IEEE and a member of the IACR.



Wei Wu received her Bachelor and Master degree from Nanjing Normal University, China. She is currently a research fellow at the School of Computer Science and Software Engineering, University of Wollongong, Australia. Her research interest is about cryptography, in particular new public key cryptography system.