Identity-Based Universal Designated Verifier Signatures*

Fangguo Zhang[1], Willy Susilo[2], Yi Mu[2], and Xiaofeng Chen[3]

[1] Department of Electronics and Communication Engineering,
Sun Yat-sen University, Guangzhou 510275, P.R. China
isdzhfg@zsu.edu.cn
[2] School of Information Technology and Computer Science,
University of Wollongong, Australia
{wsusilo, ymu}@uow.edu.au
[3] Department of Computer Science,
Sun Yat-sen University, Guangzhou 510275, P.R. China
isschxf@zsu.edu.cn

Abstract. The notion of Universal Designated Verifier Signatures (UDVS) was introduced in the seminal paper of Steinfeld *et. al.* in [6]. In this paper, we firstly propose a model of identity-based (ID-based) UDVS schemes. We note that there are two methods to achieve an ID-based UDVS scheme. We provide two constructions of ID-based UDVS schemes based on bilinear pairings that use the two methods that we have identified. We provide our security proof based on the random oracle model.

1 Introduction

In a certificate-based public key system, before a user's public key is used, the participants must firstly verify the user's certificate. As a consequence, this system requires a large storage and computing time to store and verify each user's public key and the corresponding certificate. In 1984, Shamir [5] proposed ID-based cryptosystem to simplify key management procedures in certificate-based public key setting. Since then, many ID-based encryption and signature schemes have been proposed. The main idea of ID-based cryptosystems is that the identity information of each user serves as his/her public key.

In [6], Steinfeld *et. al.* proposed a special type of digital signature scheme called *Universal Designated Verifier Signatures* (UDVS), which directly addresses the user privacy issue in user certification systems. On one hand, UDVS scheme protects user's privacy, and on the other hand, it maintains a similar convenience of use for the user and for the certificate issuer CA as in certification systems using standard digital signatures. The scenario of UDVS schemes is as follows. A user Alice is issued a signed certificate by the CA. When Alice wishes to send her certificate to a verifier Bob, she uses Bob's public key to *transform* the CA's signature

* This work is supported by the National Natural Science Foundation of China (No. 60403007) and ARC Discovery Grant DP0557493.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 825–834, 2005.

into a designated signature for Bob, using the UDVS scheme's designation algorithm, and sends the transformed CA's signature to Bob. Bob can use the CA's public key to verify the designated signature on the certificate, but is unable to use this designated signature to convince any other third party that the certificate was indeed signed by the CA, even if Bob is willing to reveal his secret-key to the third party. This is achieved because Bob's secret-key allows him to forge designated signatures by himself, so the third party is unable to tell who produced the signature (whereas Bob can, because he knows that he did not produce it). Therefore, through the use of a UDVS scheme, Alice's privacy is preserved in the sense that Bob is unable to disseminate convincing statements about Alice (of course, nothing prevents Bob from revealing the certificate statements themselves to any third party, but the third party will be unable to tell whether these statements are authentic, i.e. whether they have been signed by the CA or not). A question that directly arises from this model is "how could one design an ID-based UDVS scheme that allows Alice to convince Bob, by only knowing Bob's identity, such as email address, IP, etc."?

Our Contribution
In this paper, firstly we introduce the notion of ID-based UDVS schemes. We provide a model for such schemes together with its security requirements. We also propose two concrete constructions of ID-based UDVS schemes.

2 Preliminaries

2.1 Bilinear Pairings

Let $\mathbb{G}_1$ be a cyclic additive group generated by P, whose order is a prime q, and $\mathbb{G}_2$ be a cyclic multiplicative group with the same order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a map with the following properties:

1. **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q$
2. **Non-degeneracy:** There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$, in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. **Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

In our setting of prime order groups, the **Non-degeneracy** property is equivalent to $e(P, Q) \neq 1$ for all $P, Q \in \mathbb{G}_1$. So, when P is a generator of $\mathbb{G}_1$, $e(P, P)$ is a generator of $\mathbb{G}_2$.

Definition 1. Bilinear Diffie-Hellman (BDH) Problem:
Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP and cP (for unknown randomly chosen $a, b, c \in \mathbb{Z}_q$), compute $e(P, P)^{abc}$.

For the BDH problem to be hard, $\mathbb{G}_1$ and $\mathbb{G}_2$ must be chosen so that there is no known algorithm for efficiently solving the Diffie-Hellman problem in either $\mathbb{G}_1$ or $\mathbb{G}_2$. We note that if the BDH problem is hard for a pairing e, then it follows that e is non-degenerate.

Definition 2. Bilinear Diffie-Hellman Assumption:
If $\mathcal{IG}$ is a BDH parameter generator, the advantage $\mathtt{Adv}_{\mathcal{IG}}(\mathcal{A})$ that an algorithm $\mathcal{A}$ has in solving the BDH problem is defined to be the probability that the algorithm $\mathcal{A}$ outputs $e(P,P)^{abc}$ on inputs $\mathbb{G}_1, \mathbb{G}_2, e, P, aP, bP, cP$, where $(\mathbb{G}_1, \mathbb{G}_2, e)$ is the output of $\mathcal{IG}$ for sufficiently large security parameter k, P is a random generator of $\mathbb{G}_1$ and a, b, c are random elements of $\mathbb{Z}_q$. The BDH assumption is that $\mathtt{Adv}_{\mathcal{IG}}(\mathcal{A})$ is negligible for all efficient algorithms $\mathcal{A}$.

Throughout this paper, we define the system parameters in all schemes as follows: Let P be a generator of $\mathbb{G}_1$ with order q. The bilinear pairing is given by $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Define two cryptographic hash functions $H_1 : \{0,1\}^* \to \{0,1\}^\lambda$, in general, $|q| \geq \lambda \geq 160$, and $H_0 : \{0,1\}^* \to \mathbb{G}_1^*$. Denote PARAMS= $\{\mathbb{G}_1, \mathbb{G}_2, e, q, \lambda, P, H_0, H_1\}$, and let $|q|$ denote size of q in bits.

2.2 ID-Based Chameleon Hash Functions

A chameleon hash function is associated with a pair of public and private keys and has the following properties [4]: (1) Anyone who knows the public key can compute the associated hash function. (2) For people who do not have the knowledge of the trapdoor (i.e. the secret key), the hash function is collision resistant: it is infeasible to find two inputs which are mapped to the same output. (3) The trapdoor information's holder can easily find collisions for every given input.

The idea of chameleon hashing has been recently extended in [1] to construct an identity-based chameleon hash. An ID-based chameleon hash scheme is defined by a family of efficiently computable algorithms (Setup, Extract, Hash, Forge).

A number of ID-based Chameleon hash functions have been proposed, following the first paper proposed in [1]. In the setting of any ID-based system, there is a trusted party PKG, who only exists to initialize the system. In the following, we will review an ID-based Chameleon hash function from bilinear pairings in [8]. The four computable algorithms are defined as follows.

- Setup. PKG chooses a random number $s \in Z_q^*$ and sets $P_{pub} = sP$. PKG publishes PARAMS $= \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$, and keeps s as the MASTER KEY, which is known only by the PKG.
- Extract. A user submits his identity information ID to PKG. PKG computes the user's public key as $\mathsf{Q}_{\mathsf{ID}} = H_0(ID)$, and returns $\mathcal{S}_{\mathsf{ID}} = s\mathsf{Q}_{\mathsf{ID}}$ to the user as his private key.
- Hash. Given a message m, choose a random element R from $\mathbb{G}_1$.
  Define the hash as $\mathsf{Hash}(\mathsf{ID}, m, R) = e(R,\ P)e(H_1(m)H_0(\mathsf{ID}),\ P_{pub})$.
- Forge. $\mathsf{Forge}(\mathsf{ID}, \mathcal{S}_{\mathsf{ID}}, m, R, m') = R' = (H_1(m) - H_1(m'))\mathcal{S}_{\mathsf{ID}} + R$. One can verify that $\mathsf{Hash}(\mathsf{ID}, m, R) \stackrel{?}{=} \mathsf{Hash}(\mathsf{ID}, m', R')$ holds with equality.

3 ID-Based Universal Designated Verifier Signature Schemes

An ID-based Universal Designated Verifier Signature scheme ID-UDVS consists of six algorithms, namely (Setup, Extract, Sign, Public Verification, Designation, Designated Verification). There are four parties involved in the scheme:

- a PKG: is a trusted party who executes two operations: system setup (Setup) and user's private key generation (Extract).
- a *signer* S: who issued an ID based signature to be given to a *signature holder*.
- a *signature holder* SH: is a party who has a valid signature provided by a *signer*.
- a *designated verifier* DV: is any third party whose ID is published. *Anyone* who obtains a signature signed by the *signer* can always designate this signature to any third party, and this third party is referred as the designated verifier. In our scenario, any signature holder SH(who does not have any access to the signer's secret key) can designate the original signer's signature to a designated verifier DV, such that DV can be convinced with the authenticity of the signature, but he cannot convince any other third party about this fact, since he can always generate such a signature by himself which is indistinguishable from the original one.

The six algorithms defined in ID-UDVS are as follows.

1. Setup is a probabilistic polynomial algorithm, run by the PKG, that takes a security parameter k and returns PARAMS (system parameters) and MASTER-KEY.
2. Extract is a probabilistic polynomial algorithm, run by the PKG, that takes as input PARAMS, MASTER-KEY, and an arbitrary $\mathsf{ID} \in \{0,1\}^*$. It returns a private key $\mathcal{S}_{\mathsf{ID}}$. Here ID is the signer's identity and will be used as the signer's public key.
3. Sign is a probabilistic polynomial algorithm that is executed by the signer S. It takes PARAMS, a private key $\mathcal{S}_{\mathsf{ID}}$, an identity ID_S corresponding to the secret key $\mathcal{S}_{\mathsf{ID}}$, and a message m. The algorithm outputs a signature $\sigma(m)$ for m.
4. Public Verification is a deterministic polynomial algorithm that takes PARAMS, an identity of the signer ID_S , a message m and its signature $\sigma(m)$, and outputs either `accept` or `reject` as the verification decision.
5. Designation is a deterministic polynomial algorithm that takes as input PARAMS, a message m, a valid signature on m, $\sigma(m)$, and an identity of the designated verifier ID_DV, and outputs a designated signature $\sigma'(m)$ for m.
6. Designated Verification is a deterministic polynomial-time algorithm that takes a message m, a designated signature $\sigma'(m)$, the identity of the signer S, ID_S, and the secret key of the designated verifier $\mathcal{S}_\mathsf{DV}$ and outputs either `accept` or `reject`.

There are essentially two ways to achieve an ID-UDVS scheme. We note that these methods *do not* imply a generic construction of an ID-UDVS scheme.

1. *By incorporating the identity or public key of the designated verifier to encrypt the signature.* Using this mechanism, a signature holder can *encrypt* a signature that he has with the designated verifier's ID (or public key), such that only the designated verifier can be convinced with the authenticity of the message. This way, only the designated verifier can verify the authenticity of the signature. We call this method as an ID-UDVS *scheme with PK encryption.*

2. *By incorporating a chameleon hash function.* Using this mechanism, a signature holder uses a published chameleon hash function that is owned by the designated verifier. The designated verifier can be convinced with the authenticity of the signature, but no any other third party can, since the designated verifier can always generate another valid message signature pair by himself. We call this method as an ID-UDVS *scheme with a Chameleon Hash.*

In section 4 and 5, we provide two schemes that use the above two mechanisms.

3.1 Security Requirements

Security Against Existential Forgery on Adaptively Chosen Message and ID *Attacks.* We say an ID − UDVS scheme, which consists of six algorithms (Setup, Extract, Sign, Public Verification, Designation, Designated Verification), is *secure against existential forgery on adaptively chosen message and* ID *attacks* if no polynomial time algorithm $\mathcal{A}$ has a non-negligible advantage against a challenger $\mathcal{C}$ in the following game.

1. $\mathcal{C}$ runs Setup of the scheme. The resulting PARAMS is given to $\mathcal{A}$. MASTER KEYis kept secret from $\mathcal{A}$.
2. $\mathcal{A}$ issues the following queries as he wants.
   (a) Extract query: Given an identity ID, $\mathcal{C}$ returns the private key $\mathcal{S}_{\mathsf{ID}}$ corresponding to ID which is obtained by executing Extract.
   (b) Sign query: Given an identity ID and a message m, $\mathcal{C}$ returns a signature $\sigma(m)$ which is obtained by running Sign.
3. $\mathcal{A}$ outputs $(\mathsf{ID_S}, \mathsf{ID_{DV}}, m, \sigma'(m))$ where $\mathsf{ID_S}$ is the identity of a signer, $\mathsf{ID_{DV}}$ is the identity of a designated verifier, $\mathsf{ID_S}$ and $\mathsf{ID_{DV}}$ have never been queried to the Extract query and $(\mathsf{ID_S}, m)$ has never been queried before to the Sign query. $\mathcal{A}$ wins the game if $\sigma'(m)$ is a valid designated signature on m. That is, $\mathsf{DesignatedVerification}(m, \sigma'(m), \mathsf{ID_S}, \mathcal{S}_{\mathsf{DV}}) \stackrel{?}{=} \mathtt{accept}$ holds with equality.

We define $\mathcal{A}$'s guessing advantage $\mathtt{Adv}_{ID-UDVS}(\mathcal{A}) = |Pr[\beta' = \beta] - \frac{1}{2}|$.

4 An ID-UDVS Scheme with a PK Encryption from Bilinear Pairings

In this section, we provide our first construction of an ID-based UDVS (ID-UDVS) scheme based on bilinear pairings. Our ID-UDVS scheme functions as a standard Cha-Cheon signature [3] scheme when no designation is performed. Hence, it is compatible with the key generation, signing and verifying algorithms of the Cha-Cheon signature scheme [3]. The scheme is as follows.

1. Setup: PKG chooses a random number $s \in Z_q^*$ and sets $P_{pub} = sP$. PKG publishes system parameters PARAMS= $\{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$, and keeps s as the MASTER KEY, which is known only by itself.

2. Extract: A user submits his/her identity information ID to PKG. After a valid identification, PKG computes the user's public key as $\mathsf{Q_{ID}} = H_0(\mathsf{ID})$, and returns $\mathcal{S}_{\mathsf{ID}} = sQ_{ID}$ to the user as his/her private key.
3. Sign. Given a secret key $\mathcal{S}_{\mathsf{ID}}$, and a message m, perform the following.
   - Compute $U = r\mathsf{Q_{ID}}$, where $r \in_R \mathbb{Z}_q^*$, $h = H_1(U||m)$
   - Compute $V = (r+h)\mathcal{S}_{\mathsf{ID}}$.
   - Output the signature on m as (U, V).
4. Public Verification. Given ID, a message m, and a signature (U, V), verify if

$$e(V, P) \stackrel{?}{=} e(U + H_1(U||m)\mathsf{Q_{ID}}, P_{pub})$$

   holds with equality. If so, then output `accept`. Otherwise, output `reject`.
5. Designation. Given the signer's public key ID, a verifier's public key $\mathsf{ID_{DV}}$ and a message-signature pair (m, U, V), compute $\sigma' = e(V, \mathsf{Q_{ID_{DV}}})$, where $\mathsf{Q_{ID_{DV}}} = H_0(\mathsf{ID_{DV}})$. The designated verifier signature is (U, σ').
6. Designated Verification. Given the signer's public key ID, a verifier's secret key $\mathcal{S}_{\mathsf{ID_{DV}}}$ and a message/designated signature pair (m, U, σ'), `accept` if and only if

$$e(U + H_1(U||m)\mathsf{Q_{ID}}, \mathcal{S}_{\mathsf{ID_{DV}}}) \stackrel{?}{=} \sigma'$$

   holds with equality. Otherwise, output `reject`.

4.1 Security Analysis

Correctness and Consistency.
The correctness and consistency of the scheme is justified as follows.

$$\begin{aligned} e(V, P) &= e((r+h)\mathcal{S}_{\mathsf{ID}}, P) = e((r+h)s\mathsf{Q_{ID}}, P) \\ &= e((r+h)\mathsf{Q_{ID}}, P_{pub}) = e(r\mathsf{Q_{ID}}, P_{pub})e(h\mathsf{Q_{ID}}, P_{pub}) \\ &= e(U, P_{pub})e(H_1(U||m)\mathsf{Q_{ID}}, P_{pub}) = e(U + H_1(U||m)\mathsf{Q_{ID}}, P_{pub}) \end{aligned}$$

$$\begin{aligned} e(U + H_1(U||m)\mathsf{Q_{ID}}, \mathcal{S}_{\mathsf{ID_{DV}}}) &= e(r\mathsf{Q_{ID}} + H_1(U||m)\mathsf{Q_{ID}}, s\mathsf{Q_{ID_{DV}}}) \\ = e((r+h)s\mathsf{Q_{ID}}, \mathsf{Q_{ID_{DV}}}) &= e((r+h)\mathcal{S}_{\mathsf{ID}}, \mathsf{Q_{ID_{DV}}}) = e(V, \mathsf{Q_{ID_{DV}}}) = \sigma' \end{aligned}$$

Theorem 1. *If a valid universal designated signature can be generated without the knowledge of a valid signature or a secret key of the signer, then the BDH problem may be solved in a polynomial time.*

Proof. Let us recall the BDH problem as follows. Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP and cP (for unknown randomly chosen $a, b, c \in \mathbb{Z}_q$), compute $e(P, P)^{abc}$. To show the proof, we assume there is a polynomial algorithm $\mathcal{A}$ that can generate a valid universal designated signature σ' for a message m, without the knowledge of a signature σ generated by the signer, and without the signer's secret key. The algorithm $\mathcal{A}$ accepts an ID of the signer,

ID_A, an ID of the designated verifier, ID_C, and a message m, and it outputs a valid universal designated signature (U, σ'), where

$$Pr[\mathsf{DesignatedVerification}(m, (U, \sigma'), \mathsf{ID}_C) = \mathtt{accept}] = 1.$$

We will show how to use this algorithm to solve the BDH problem.

In our setting, we know the public information P_{pub}, ID_A (the ID of the signer) and ID_C (the ID of the designated verifier). From this public information, we can obtain $\mathsf{Q}_{\mathsf{ID_A}} = H_0(\mathsf{ID}_A)$ and $\mathsf{Q}_{\mathsf{ID_C}} = H_0(\mathsf{ID}_C)$. Since P is a generator in $\mathbb{G}_1$, then we can rewrite these three parameters as

$$\mathsf{Q}_{\mathsf{ID_A}} = aP \qquad P_{pub} = bP \qquad \mathsf{Q}_{\mathsf{ID_C}} = cP$$

We note that $\mathcal{S}_{\mathsf{ID}_A} = b\mathsf{Q}_{\mathsf{ID_A}} = abP$ and $\mathcal{S}_{\mathsf{ID}_C} = b\mathsf{Q}_{\mathsf{ID_C}} = bcP$. Now, we construct an algorithm $\hat{A}$ to solve the BDH problem as follows. Algorithm $\hat{\mathcal{A}}$ will control $\mathcal{A}$ and replaces $\mathcal{A}$'s interaction with the signer by simulation. Firstly, $\hat{\mathcal{A}}$ generates a list of ID of its choice, together with a random s_i associated with it. The size of this set is 2^ℓ, where ℓ is the security parameter. The idea of the game is illustrated as follows. The purpose of $\hat{\mathcal{A}}$ is to inject the information above (aP, bP, cP) during the simulation. Without losing generality, we only show the interaction where $\mathcal{A}$ interacts with $\hat{A}$ for the information that $\hat{\mathcal{A}}$ wants. There is a probability that $\hat{\mathcal{A}}$ will fail, i.e. when $\mathcal{A}$ queries the secret key for either $\mathsf{ID_A}$ or $\mathsf{ID_C}$ that will match with the published P_{pub}. Since $\hat{\mathcal{A}}$ does not have this information, then $\hat{\mathcal{A}}$ will halt the game. The probability of this failure to happen is $\leq \frac{1}{2^\ell}$. $\mathcal{A}$ will be run twice with a different random query set, but from the same list of ID's generated at the first place. The attack is successful, when $\mathcal{A}$ outputs two signatures for the given parameters (forking lemma). More concretely, the algorithm is described as follows. Firstly, $\hat{\mathcal{A}}$ selects two random numbers $a', a'' \in \mathbb{Z}_q$, where $a' - a'' = 1 \pmod q$. Then, $\hat{\mathcal{A}}$ will control $\mathcal{A}$ as follows.

First Round

H_1 - *Hash Query.* When $\mathcal{A}$ requests the value of $H_1(U_1||m)$, for the targeted parameters, $\hat{\mathcal{A}}$ responds with $a'\mathsf{Q}_{\mathsf{ID_A}}$. Otherwise, responds with the list that he has generated.

Random Generation Query. When $\mathcal{A}$ requests $\hat{\mathcal{A}}$ to generate a random number $r \in \mathbb{Z}_q$ and returns U, if the targeted parameters are used, then $\hat{\mathcal{A}}$ responds by $r \in Z_q$, keeps this r in his separate list and returns rP.

Output. Eventually, the output of the first round is (U, σ_1') where $\sigma_1' = e(U + a'\mathsf{Q}_{\mathsf{ID_A}}, \mathcal{S}_{\mathsf{ID}_C})$.

Second Round

H_1 - *Hash Query.* When $\mathcal{A}$ requests the value of $H_1(U_1||m)$, for the targeted parameters, $\hat{\mathcal{A}}$ responds with $a''\mathsf{Q}_{\mathsf{ID_A}}$. Otherwise, responds with the list that he has generated.

Random Generation Query. When $\mathcal{A}$ requests $\hat{\mathcal{A}}$ to generate a random number $r \in \mathbb{Z}_q$ and returns U, if the targeted parameters are used, then $\hat{\mathcal{A}}$ responds returning rP, where r is the number that he kept from the first round.

Output. Eventually, the output of the first round is (U, σ_2') where $\sigma_2' = e(U + a''\mathsf{Q}_{\mathsf{ID}_A}, \mathcal{S}_{\mathsf{ID}_C})$.

Obtaining (U, σ_1') and (U, σ_2'), $\hat{\mathcal{A}}$ can solve the BDH problem by first computing $d = \frac{\sigma_1'}{\sigma_2'}$ and output d as the solution of BDH problem.

The correctness of this algorithm is justified as follows.

$$d = \frac{\sigma_1'}{\sigma_2'} = e(U + a'\mathsf{Q}_{\mathsf{ID}_A}, \mathcal{S}_{\mathsf{ID}_C})/e(U + a''\mathsf{Q}_{\mathsf{ID}_A}, \mathcal{S}_{\mathsf{ID}_C}) = e((a' - a'')\mathsf{Q}_{\mathsf{ID}_A}, \mathcal{S}_{\mathsf{ID}_C})$$
$$= e(\mathsf{Q}_{\mathsf{ID}_A}, \mathcal{S}_{\mathsf{ID}_C}) = e(aP, bcP) = e(P, P)^{abc}$$

This contradicts with the BDH assumption, and hence, we complete the proof. As mentioned earlier, the probability that the simulation will fail is $\leq \frac{1}{2^\ell}$, where ℓ is the security parameter. ■

Theorem 2. *Having received a UDVS signature (U, σ'), the designated verifier* DV *cannot convince any other third party about the authenticity of the designated signature.*

Proof. The designated verifier DV cannot convince anyone else about the authenticity of (U, σ') because he can always generate this signature by himself after observing U. More precisely, he can always generate $\hat{U} = r\mathsf{Q}_{\mathsf{ID}_A}$, for a random $r \in \mathbb{Z}_q$, and compute $\hat{\sigma'} = e(\hat{U} + H_1(\hat{U}||m')\mathsf{Q}_{\mathsf{ID}}, \mathcal{S}_{\mathsf{ID}_{\mathsf{DV}}})$, for a random $m' \in \mathbb{Z}_q$, where $m' \neq m$, which is indistinguishable from the original signature. We note that the new pair $(\hat{U}, \hat{\sigma'})$ will pass the Designated Verification algorithm. ■

5 An ID-UDVS Scheme with a Chameleon Hash from Bilinear Pairings

In this section, we present our second ID-based UDVS scheme. In contrast to our first scheme, our second scheme makes use of bilinear pairings together with an ID-based chameleon hash function. The scheme is as follows.

- Setup: PKG selects a random number $s \in Z_q^*$ and sets $P_{pub} = sP$. Define another cryptographic hash function: $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and ID-Based Chameleon Hash: Hash. The center publishes system parameters PARAMS= $\{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_0, H_1\}$ and the ID-based Chameleon Hash Hash.
- Extract: A user submits his/her identity information ID to PKG. PKG computes the user's public key as $\mathsf{Q}_{\mathsf{ID}} = H_0(\mathsf{ID})$, and returns $\mathcal{S}_{\mathsf{ID}} = s\mathsf{Q}_{\mathsf{ID}}$ to the user as his/her private key.
- Sign. Given a secret key $\mathcal{S}_{\mathsf{ID}}$, and a message $m \in \mathbb{Z}_q$, compute $r = e(P, P)^k$, where $k \in_R \mathbb{Z}_q^*$, $c = H_1(m||r)$ and $U = kP - c\mathcal{S}_{\mathsf{ID}}$. The signature on a message m is $\sigma = (c, U)$.
- Public Verification. Given ID, a message m, and a signature (c, U), verify if

$$c \stackrel{?}{=} H_1(m||e(U, P)e(\mathsf{Q}_{\mathsf{ID}}, P_{pub})^c)$$

holds with equality.

- Designation. Given the signer's public key ID, a verifier's public key $\mathsf{ID_{DV}}$ and a message/signature pair (m, c, U), create a UDVS signature as follows.
  - Compute $r = e(U, P)e(\mathsf{Q_{ID}}, P_{pub})^c$.
  - Compute $r' = H_1(e(P, P)^{k'})$ for a random $k' \in \mathbb{Z}_q^*$.
  - Compute $h = \mathsf{Hash}(\mathsf{ID_{DV}}, r', R)$ for a random $R \in \mathbb{G}_1$.
  - Compute $c' = H_1(m, c, r, h)$.
  - Compute $S' = k'P - c'U$.
  - Output the designated signature as $\sigma' = (r, R, c', S')$.
- Designated Verification. Given the signer's public key ID, a verifier's secret key $\mathcal{S}_{\mathsf{ID_{DV}}}$ and a message/UDVS signature pair $m, (r, R, c', S')$, accept if and only if

$$c' \stackrel{?}{=} H_1(m, c, r, h)$$

holds with equality. Here, $c = H_1(m||r)$, $h = \mathsf{Hash}(\mathsf{ID_{DV}}, R, r')$, and $r' = H_1(e(S', P)(r \cdot e(\mathsf{Q_{ID}}, P_{pub})^{-c})^{c'})$.

5.1 Security Analysis

Correctness and Consistency
The correctness and consistency of our second scheme are justified as follows.

$$\begin{aligned} c &= H_1(m||e(U, P)e(\mathsf{Q_{ID}}, P_{pub})^c) = H_1(m||e(kP - c\mathcal{S}_{\mathsf{ID}}, P)e(\mathsf{Q_{ID}}, P_{pub})^c \\ &= H_1(m||e(P, P)^k e(c\mathcal{S}_{\mathsf{ID}}, P)e(c\mathcal{S}_{\mathsf{ID}}, P)^{-1}) = H_1(m||e(P, P)^k) = H_1(m||r) \end{aligned}$$

It is easy to see that the following equation holds with equality when it is generated correctly: $c' \stackrel{?}{=} H_1(m, c, r, h)$ for $c = H_1(m||r)$, $h = \mathsf{Hash}(\mathsf{ID_{DV}}, R, r')$, and $r' = H_1(e(S', P)(r \cdot e(\mathsf{Q_{ID}}, P_{pub})^{-c})^{c'})$. ∎

Theorem 3. *The designated signature (r, R, c', S') on a message m cannot be used by the designated verifier* DV *to convince any other third party.*

Proof. DV can always generate the designated verifier (r, R, c', S') on a message $m' \in \mathbb{Z}_q$, where $m' \neq m$, by himself, which is indistinguishable from the original signature. The way to do this is as follows.

- Select a random message $m' \in \mathbb{Z}_q$, and a random number $r \in \mathbb{Z}_q$.
- Compute $c = H_1(m'||r)$.
- Compute $r' = e(P, P)^{k'}$, for a random $k' \in \mathbb{Z}_q$.
- Compute $h = \mathsf{Hash}(\mathsf{ID_{DV}}, R, r')$, for a random $R \in \mathbb{G}_1$.
- Compute $S' = k'P - c'U$.
- Output (r, R, c', S').

Moreover, after receiving a valid designated signature (r, R, c', S'), the designated signature still can modify this signature by executing the Forge algorithm. Due to the construction of the ID-based Chameleon Hash function used, he can always find a different $R' \neq R$ that will satisfy the DesignatedVerification algorithm. ∎

The formal security proof is omitted due to page limitation.

6 Conclusion

In this paper, we propose a formal definition for identity-based Universal Designated Verifier Signatures (ID-UDVS). We provide two secure ID-UDVS schemes based on bilinear pairings. Our first scheme uses the Cha-Cheon ID-based signature scheme, while our second scheme uses an ID-based Chameleon Hash function.

References

1. G. Ateniese and B. de Medeiros. Identity-based Chameleon Hash and Applications. *Financial Cryptography 2004, LNCS 3110*, pages 164 - 180, 2004.
2. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *Advanced in Cryptology - Asiacrypt 2001, LNCS 2248*, pages 514–532, 2001.
3. J. C. Cha and J. H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. *6th International Workshop on Theory and Practice in PKC (PKC 2003), LNCS 2567*, pages 18–30, 2003.
4. H. Krawczyk and T. Rabin. Chameleon hashing and signatures. *Network and Distr System Security Symp, The Internet Society*, pages 143 – 154, 2000.
5. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, LNCS 196*, pages 47–53, 1985.
6. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk. Universal designated-verifier signatures. *Advances in Cryptology - Asiacrypt 2003, LNCS 2894*, pages 523 – 543, 2003.
7. R. Steinfeld, H. Wang, and J. Pieprzyk. Efficient Extension of Standard Schnorr/RSA signatures into Universal Designated-Verifier Signatures. *7th International Workshop on Theory and Practice in PKC (PKC 2004), LNCS 2947*, pages 86–100, 2004.
8. F. Zhang, R. Safavi-Naini, and W. Susilo. ID-Based Chameleon Hashes from Bilinear Pairings. *Cryptology ePrint Archive, Report 2003/208*, 2003.