Committal Deniable Signatures over Elliptic Curves

Song Han & Wanquan Liu
Department of Computing
Curtin University of Technology
WA 6102 Australia
{hans, wanquan}@cs.curtin.edu.au

Abstract

In this paper, a new deniable signature scheme, i.e. the committal deniable signature scheme is proposed. This scheme has been constructed by use of bilinear pairings over elliptic curves. In addition to the general property possessed by other deniable signature schemes, there are some new features for this new scheme. One important of them is that signer is not able to forge this type of deniable signatures on behalf of the verifier or any third party. Another feature is that any third party, even though she can obtain the committal deniable signatures by tapping, cannot distinguish the actual signer between the verifier and the signer in the underlying system.

1 Introduction

With the development of electronics commerce and wireless applications, the need for protecting the privacy of senders and the contents of signature is much more exigent. Since the first public key cryptography [13] was proposed in 1976, various public key cryptosystems or protocols have been developed. Some of them have also been applied in e-commerce and e-government where *information assurance* is needed. Among these protocols, deniable signature is one of the tools which would be used to maintain the privacy of the underlying signatures in cryptography. A deniable signature scheme allows the designated verifier to verify the validity of the underlying deniable signature; Furthermore, an adversary other than the verifier (resp. signer) is not able to distinguish the actual signer between the signer and the verifier. Roughly speaking, the deniable signatures can provide:

(1) verifying that a message (i.e. signature) really comes from the claimed sender;

(2) protecting the privacy of the signers in the underlying system only against the adversary outside of the system;

(3) the other party (i.e. the designated verifier) of this system can know who is the actual signer.

Due to these features, the deniable signatures are much more appropriate for some special applications in *information assurance*. For example, the following scenario is suitable for the deniable signatures.

In an electronics community, say, local area network, Alice and Bob communicate with each other. One wants to find out the other's preference and then provides better service for the other. They may allow the third party to know the exchanged information (might be digital signature) but they do not want the third party to know who is the actual sender (or signer). In addition, one can be convinced that the information is really signed by the other.

Huberman and Franklin et al proposed the first deniable signature scheme [19]. They made use of Schnorr identification protocol to implement the scheme. However, they did not furnish detailed algorithms on the new cryptographic primitive. Also, the HFH scheme was in the setting of multiplicative generic group modular a large prime, which is less secure complexity compared to the proposed scheme in this paper. Furthermore, the HFH scheme was not able to prevent one party from forging deniable signatures for the other party, which probably leads to illegal application. For example, one party may obtain something useful by forging deniable signatures on behalf of the other party. Thus, the latter party will probably harm the benefits of the former party. Surely, that was not what the former party intends to happen.

Recently, the pairings over elliptic curves have been used in various cryptographic systems due to its advantage in terms of bilinearity and secure complexity. For instance, Joux proposed a three-party key agreement protocol that requires only one round of communication [21]. Boneh and Franklin [1] by use of bilinear pairings built a notable encryption system which now have been implemented in some email systems. Han et al proposed an identity-based confirmer signature scheme based on pairings [18]. Smart proposed authenticated key agreement protocol [29]; Libert and Quisquater [24] proposed sign-encryption scheme based on pairings. Paterson [25] constructed an efficient general signature scheme. In this paper, we will propose a new deniable signature scheme based on pairings - a committal deniable signature scheme. Because of the widely affirmed features of cryptosystems over elliptic

0-7803-8396-6/04/$20.00 © 2004 IEEE

curves, the proposed new scheme in this paper will probably be used in some applications, i.e. e-commerce and e-government.

This new scheme will have the following significant characteristics.

(1) The concept of the committal deniable signatures is defined.

(2) This scheme is based on the settings of elliptic curves. ECC provides greater efficiency in terms of computational overheads, key sizes and bandwidth [10].

(3) This work brings the committal property into the deniable signatures for the first time.

(4) The provable security proposed in [11] can be guaranteed with the proposed committal deniable signature scheme. That is, the new scheme has the unforgeable property [17].

The organization of this paper is as follows. In section 2, we propose the new model for the committal deniable signatures. In section 3, some preliminaries for the underlying scheme are provided. In section 4, the committal deniable signature scheme will be described in detail. The analysis and discussions are addressed in section 5. The performance have been investigated in section 6. Finally, the conclusions are presented at the end of the paper.

2 Framework

In this section, we will propose the concept of a committal deniable scheme.

Definition 1. (Committal Deniable Signatures) *A committal deniable signature scheme is a digital signature scheme comprised of the following three procedures:*

(1) **Setup** *One inputs a security parameter* k, *a probabilistic algorithm will output a global parameter* Par_{sys}, *the private keys* SK_1 *and* SK_2 , *and the public keys* PK_1 *and* PK_2 *for the signer and verifier respectively.*

(2) **Sign** *A probabilistic algorithm with inputs of both the signer and the verifier's public keys* PK_1 *and* PK_2, *the signer's private key* SK_1, *the global parameters* Par_{sys} *and a document* m, *can generate* cdsig, *a deniable signature of the document (called a committal deniable signature).*

(4) **Verify** *An algorithm for establishing the validity of the alleged committal deniable signature* cdsig *of a message with respect to the underlying two public keys* PK_1 *and* PK_2 *of signer and verifier. This algorithm is executed only by the verifier. In the process of verification, the verifier will also use her/his private key* SK_2 *and the global parameters* Par_{sys}.

In addition, a secure committal deniable signature scheme must satisfy the following properties.

(1) **Correctness:** If the signer and the verifier correctly follow the precedure of the underlying scheme, then the verifier will be always accepting the validity of the committal deniable signatures ***cdsig***.

(2) **Deniable property:** Any adversary outside of the network is not able to distinguish who is the actual signer (between the verifier and the signer) even though she/he can get the signature ***cdsig*** by tapping. For the explicit depiction of the deniable property, see the subsection 5.2.

(3) **Committal property:** Given a valid committal deniable signature ***cdsig*** on message m, the verifier can be convinced that it is generated by the signer of the underlying network; On the other hand, the signer is not able to forge valid signatures on behalf of the verifier (see subsection 5.3 for the details).

(4) **Unforgeability:** Any adversary (other than the signer) is not able to forge valid committal deniable signatures for the signer. That is, the signatures have unforgeability property (see the subsection 5.4 for the details).

3 Preliminaries

3.1 Notations

Some notations used in this paper will be presented in this section. Let q be a large prime, and Z_q^* be $Z_q \backslash \{0\}$, where $Z_q = \{0, 1, 2, ..., q-1\}$. Let n be a positive integer, H and H_1 be two cryptographic hash functions:

$$H : \{0,1\}^* \rightarrow G_1$$

and

$$H_1 : \{0,1\}^* \times G_1 \rightarrow G_1,$$

where $\{0,1\}^*$ is a set of $\{0,1\}$-string with arbitrary length, and G_1 will be explained in the sequel.

Definition 2. (Negligible) *Let $g(n)$ be a rational function. We call $g(n)$ negligible, if for every constant $c > 0$ and all sufficiently large n, it always holds*

$$g(n) < \frac{1}{n^c}.$$

With the above definition, it is easy to derive the non-negligible property.

3.2 Pairings over Elliptic Curves

Let p be a sufficiently large prime that satisfies: ($\tilde{a}$) $p \equiv 2 \bmod 3$; ($\tilde{b}$) $p = 6q - 1$, where q is also a large prime. Consider respectively the elliptic curves E/F_p and E/F_{p^2} [5] [23] defined by the equation.

$$y^2 = x^3 + 1. \quad (1)$$

Let G_1 be an additive group of points of prime order q on an elliptic curve E/F_p and let G_2 be a multiplicative group of same order q of some finite field F_{p^2} [23]. Roughly speaking, an elliptic curve is a set of all points Q whose abscissa value and vertical value satisfy equation (3.1).

The modified Weil pairing is a bilinear mapping from $G_1 \times G_1$ to G_2,

$$e : G_1 \times G_1 \to G_2$$

satisfying that the Elliptic Curve Discrete Logarithm (ECDL) problems [23] are difficult in G_1 and the Computational Diffi-Hellman (CDH) problems [13] and the Inversion of Weil pairing (IWP) problems are difficult in G_2. All these requirements are needed in our new scheme.

The modified Weil pairings $e : G_1 \times G_1 \to G_2$ must have the following properties:

(1) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for every pair $P, Q \in G_1$ and for any $a, b \in Z_p$.

(2) Non-degeneracy: there exists at least one point $P \in G_1$ such that $e(P, P) \neq 1$.

(3) Efficient Computability: there are efficient algorithms to compute the bilinear pairings e.

In addition, the existence of the modified Weil pairings can be found at [1]. Also some computation issue has been addressed there. Next, we will describe the Elliptic Curve Discrete Logarithms.

3.3 Elliptic Curve Discrete Logarithms

Definition 3. (Elliptic Curve Discrete Logarithm Problem) Given G_1, one can choose P as a generator from G_1. With a given xP, where x is an unknown random element of Z_q^*, the Elliptic Curve Discrete Logarithm (ECDL) problem is to find x.

Assumption 1. (ECDLP Assumption) *Given xP and a generator P in G_1 with unknown $x \in Z_q^*$. An algorithm A has advantage δ in solving ECDLP in G_1 if*

$$Pr[A(P, xP) = x] \geq \delta$$

where the probability is taken over the random coin tosses(i.e.random choice, see [4]) of $P \in G_1$, the random coin tosses of x in Z_q^*, and the random bits of A. ECDLP Assumption implies there is no probabilistic polynomial time algorithm to solve the Elliptic Curve Discrete Logarithm problem with non-negligible advantage.

3.4 Inversion of Modified Weil Pairings

Definition 4. (Inversion of Modified Weil Pairings Problem) Given G_1, G_2 and $e(\cdot, \cdot)$ as above, choose P a generator from G_1, given $e(P, *)$, here $*$ is an unknown point of G_1, the Inversion of Modified Weil Pairings (IWP) problem is to find $Q \in G_1$ such that

$$e(P, Q) = e(P, *).$$

Assumption 2. (IWP Assumption) *Given G_1, G_2 and $e(\cdot, \cdot)$ as above, choose P a generator from G_1, given $e(P, *)$, here $*$ is an unknown point of G_1. An algorithm A has advantage δ in solving the Inversion of Modified Weil Pairings Problem(i.e.IWP) in G_1 and G_2 if*

$$Pr[A(P, e(P, *), G_1, G_2) = Q] \geq \delta$$

with the constraint of $e(P, Q) = e(P, *)$; and where the probability is taken over the random coin tosses(i.e.random choice) of $P, * \in G_1$, the random coin tosses of $e(P, *)$ in G_2, and the random bits of A. The IWP Assumption is that there is no probabilistic polynomial algorithm to solve the Inversion of Modified Weil Pairings problem with non-negligible advantage.

4 Committal Deniable Signature Scheme

With notations and definitions in previous section, we can present detail description of the committal deniable signature scheme in this section.

4.1 Setup

We assume there is a network with two parties: the signer and the verifier. They can obtain their own public and private keys by running the computer server of this network in some smart way. First,

(1) Choose p as a sufficient large prime satisfying: (a) $p \equiv 2 \bmod 3$; (b) $p = 6q - 1$, where q is also a large prime. Consider the two elliptic curves E/F_p and E/F_{p^2} defined by equation

$$y^2 = x^3 + 1.$$

Let G_1 be an additive group of points of prime order q on an elliptic curve E/F_p and let G_2 be a multiplicative group of same order q of F_{p^2}.

(2) Choose three cryptographic hash functions:

$$H : \{0,1\}^* \mapsto G_1.$$

$$H_1 : \{0,1\}^* \times G_1 \mapsto G_1.$$

$$H_2 : \{0,1\}^* \times (G_1)^4 \times (G_2)^2 \mapsto Z_q.$$

(3) Construct a bilinear function as defined in subsection 3.2:

$$e : G_1 \times G_1 \longmapsto G_2$$

As how to construct the modified Weil pairings, see [1] and [20].

(4) Select a generator element $P \in G_1$. Therefore $e(P, P)$ is a generator element of G_2.

(5) Select an integer a from Z_q^* as system secret key; Set $P_{pub} = aP$ as the public key of this network.

(6) The signer and the verifier choose respectively random strings $f_1 \in \{0,1\}^*$ and $f_2 \in \{0,1\}^*$. Then, they compute $Q_1 = H(f_1)$ and $Q_2 = H(f_2)$ as their public key respectively.

(7) The signer and the verifier obtain from the computer server respectively their private key: $S_1 = aQ_1$ and $S_2 = aQ_2$.

(8) Let $M = \{0,1\}^*$ (a set of strings of any length) be the message space.

Therefore, the global public parameters of this network are packed in the following set:

$$PK = \{P, P_{pub}, p, q, H_2, e(\cdot,\cdot), H, H_1\}.$$

The signer's private key is S_1 and public key is Q_1; the verifier's private key is S_2 and public key is Q_2.

In fact, as the two parties start to communicate, the signer and the verifier can respectively sign some important message for the other. That is, the signer can sign some message and have the verifier verify the signature (generated by the signer); The verifier can also sign some message and have the signer verify the signature (generated by the verifier).

4.2 SIGN

This is a generation algorithm of committal deniable signatures. In the signature generation algorithm, we make use of the Schnorr identification protocol [28] to generate the deniable part of the committal deniable signatures. Now the signer will be signing an important message and have the verifier to verify it.

Given a message $m \in M$, the signer performs the following operations.

(1) The signer respectively chooses randomly and uniformly U_2 from G_1 and c_2 from Z_q, and sets $t_2 = e(Q_2, P_{pub})^{-c_2} e(U_2, P)$.

(2) The signer selects a random $r \in Z_q$ and sets $t_1 = e(rP, P_{pub})$.

(3) She/He computes $c = H_2(P, Q_1, Q_2, t_1, t_2, m, P_{pub})$, and then sets $c_1 = (c - c_2) \bmod q$.

(4) She/He computes $U_1 = c_1 S_1 + r P_{pub}$ over G_1.

(5) She/He computes $SPKIWP(t, P)$, which is a signature of proof of knowledge of a solution to IWP (defined in section 3.4) on $t = e(P_{pub}, Q_1)$ and P. $SPKIWP(t, P)$ will be detailed in subsection 4.3.

Therefore, the committal deniable signature on message m will be

$$\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$$

Remark 1. *Given a committal deniable signature* $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ *on message* m, *we call*

$$\{c_1, c_2, t_1, t_2, U_1, U_2\}$$

the deniable part of the committal deniable signature on message m; *we call*

$$SPKIWP(t, P)$$

the committal part of the committal deniable signature on message m.

Remark 2. Similarly, the verifier can generate a committal deniable signature for the signer by performing the following signature generation algorithm.

(1) Respectively chooses randomly and uniformly U_1 *from* G_1 *and* c_1 *from* Z_q, *and sets* $t_1 = e(Q_1, P_{pub})^{-c_1} e(U_1, P)$.

(2) Selects a random $r \in Z_q$ *and sets* $t_2 = e(rP, P_{pub})$.

(3) Computes $c = H_2(P, Q_1, Q_2, t_1, t_2, m, P_{pub})$, *and then sets* $c_2 = (c - c_1) mod q$.

(4) Computes $U_2 = c_2 S_2 + r P_{pub}$.

(5) Computes $SPKIWP(t, P)$, *which is a signature of proof of knowledge of a solution to IWP (defined in section 3.4) on* $t = e(P_{pub}, Q_2)$ *and* P.

Hence, the signature is

$$\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$$

4.3 VERI

This is an algorithm executed only by the honest verifier on committal deniable signatures sent by the signer. Given a message m and its alleged committal deniable signature

$$\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$$

The verifier of the underlying network can validate the committal deniable signature by running the following procedures:

(1) Check whether $e(U_2, P) = e(Q_2, P_{pub})^{c_2} \cdot t_2 (\mathrm{mod} q)$.

(2) Check the following condition:

$$c_1 + c_2 = H_2(P, Q_1, Q_2, t_2, t_1, m, P_{pub}).$$

(3) Check the following condition:

$$e(Q_1, P_{pub})^{c_1} t_1 = e(U_1, P) \bmod q.$$

(4) check whether $SPKIWP(t, P)$ is a valid signature of proof of knowledge of a solution to IWP (defined in section 3.4) on t and P, where $t = e(P_{pub}, Q_1)$. In the process of checking the validity of $SPKIWP(t, P)$, the verifier will use her own private key S_2 as described below.

If the committal deniable signature satisfies all the above properties, then the verifier will be accepting the signature as valid; otherwise, he/she will reject it. By using the signer's public key, the verifier knows that the signature must come from the underlying signer. This is due to a fact that the deniable part (namely $\{c_1, c_2, t_1, t_2, U_1, U_2\}$) of the signature illustrates that the signature is efficiently created by someone who knows the private key corresponding to the public key Q_1 or Q_2. Further, because of the zero-knowledge property [28] of proof of knowledge of identifications (i.e. the zero-knowledge of the committal part, namely $SPKIWP(t, P)$), any adversary outside of the network is not able to distinguish the actual signer between the verifier and the signer, which will be explained in Remark 3.

It should be noted that for previous deniable signatures proposed in [19], it is possible to forge signatures without knowing the secret key, i.e., the signer may probably forge valid deniable signatures for the verifier in the underlying network.

In the proposed scheme in this paper, the verifier will use his/her own public and private keys, and the signer's public key to verify the signatures. Furthermore, the verifier must be interacting with the signer in order to verify the signatures. The interaction between the verifier and the signer can prevent the signer from forging valid deniable signatures for the verifier.

Remark 3. *Both the signing and the verifying algorithms of the proposed committal deniable signature scheme contain a committal part, i.e. $SPKIWP(t, P)$, which is a signature of proof of knowledge of a solution to IWP (defined in section 3.4) on $t = e(P_{pub}, Q_2)$ and P. Now we will explain the $SPKIWP(t, P)$ and show how to construct the committal part clearly. As a matter of fact, $SPKIWP(t, P)$ is an interactive protocol between the signer and the designated honest verifier of the underlying scheme as described below:*

(1) The signer chooses an element $w \in Z_q^*$ randomly, computes $F = wP$, and sends F to the verifier.

(2) The verifier chooses an element $x \in Z_q$ randomly, and sends it to the signer.

(3) After receiving x, the signer computes

$$Y = wP_{pub} + xS_1,$$

and sends it to the verifier.

(4) The verifier checks whether

$$e(Y, Q_2) = \{e(F, S_2) \times e(Q_1, S_2)^x\} mod q$$

If the above equation holds, then $SPKIWP(t, P)$ is a valid signature; Otherwise, it is regarded as an illegal signature.

5 Analysis and Discussions

In this section, we will come up with the security proofs and analysis of the proposed committal deniable signature scheme. First, we need to prove that the proposed scheme possesses the correctness property. Thereafter, we will investigate the deniable property and the committal property. Moreover, the unforgeability of the committal deniable signatures is also discussed.

5.1 Correctness

The correctness of the committal deniable signatures implies that if the signer correctly computes the signatures by the signature generation algorithm, then the verifier who correctly executes the verification algorithm will certainly accept the signature.

Theorem 1. Given any message $m \in M$, if the signer and the verifier follow all the procedures of the committal deniable signature scheme, then the verifier can be convinced that the signature $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ is valid.

Because of the space limitation, the proof of this theorem is provided in the full version of this paper.

5.2 Deniable Property

The deniable property of the committal deniable signature scheme implies that any adversary outside of the network can not distinguish who is the actual signer (between the verifier and the signer) even though she can get the signature by tapping.

Theorem 2. With the assumption of Elliptic Curve Discrete Logarithm Problem (ECDLP) in G_1 and the assumption of Discrete Logarithm Problem in G_2, the proposed committal deniable signature scheme has the deniable property.

Proof Following the definitions in **Remark 1** on the committal deniable signatures, one call

$$\{c_1, c_2, t_1, t_2, U_1, U_2\}$$

the deniable part of this signature. At the same time, the part

$$SPKIWP(t, P)\}$$

as the committal part of this signature.

As for the committal part, it is only a lip-deep signature. In fact, as soon as the (designated) verifier intends to validate this signature, the committal part is activated by the two communication parties, i.e. the verifier and the signer. Thereafter, the committal part becomes an interactive protocol executed only by the (designated) verifier and the signer. Therefore, the adversary can not obtain anything useful about the committal part even though she got the signature $\{SPKIWP(t, P)\}$.

With the deniable part, we need the following claim:

Claim 1. If the adversary can obtain the value r in the step 2 of signature generation with the advantage δ, then she/he will be able to distinguish the actual signer between the (designated) verifier and the signer with the probability being larger than $\frac{1}{2} + \delta$.

The proof of this claim is provided in the full version of this paper since the space is limited.

5.3 Committal Property

The committal property of the committal deniable signature (i.e., *CD signature*) scheme can be described as: Given a valid signature $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ on message m, the verifier can be convinced that it is generated by the signer of the underlying network; On the other hand, the signer can not forge a valid signatures on behalf of the verifier, i.e., the signer cannot create a valid signature $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$, where $t = e(P_{pub}, Q_2)$.

In order to clearly explain the reason that the new scheme has the committal property, we first give two definitions.

Definition 5. The CD signature w.r.t the verifier is valid. *Suppose* $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ *is a committal deniable signature created by the underlying system on message* m, *we call the* CD signature *w.r.t the verifier is* valid, *if the following conditions hold:*

1. *it is valid checked by the signer.*
2. $t = e(P_{pub}, Q_2)$.

This implies that the CD signature is a genuine signature generated by the verifier and can be validated by the signer.

Definition 6. The signer is successful in being able to forge CD signatures on behalf of the verifier. *Suppose* $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ *(on any message* m*) is a CD signature forged by the signer for the verifier, we call the signer is successful in being able to forge CD signatures on behalf of the verifier, if* $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ *is* a valid CD signature w.r.t to the verifier.

As a matter of fact, any one with the private key S_1 or S_2 can efficiently compute valid signatures in this network. Consequently, when the verifier receives a valid signature $\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$ on message m, she/he can be convinced that it is signed by the signer.

At the same time, because of the difficulty of the IWP, the signer cannot work out a solution to the IWP on t and P (where $t = e(Q_2, P)$) in which the private key S_2 is only secretly held by the verifier. Therefore, the signer can not construct a valid signature $SPKIWP(t, P)$ (where $t = e(Q_2, P)$). As a result, she is not able to forge valid signatures on behalf of the verifier. □

5.4 Security against Forgeability

In this section, we shall prove that any adversary other than the signer is not able to forge valid committal deniable signatures for the signer. That is, the new scheme has the unforgeability property. In the following, we will give a mathematical proof for this property.

Theorem 3. Under the assumption that the signer holds the private key securely and the assumption for Elliptic Curve Discrete Logarithm Problem, the proposed committal deniable signature scheme has unforgeability property.

Proof We need to prove that any adversary (other than the signer) is not able to forge valid committal deniable signatures for the signer. Now, Suppose that the adversary successfully forged a valid committal deniable signature

$$\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$$

for a message m, then the underlying adversary can compute the signer's private key S_1 in polynomial time.

In fact, since the forged signature

$$\{c_1, c_2, t_1, t_2, U_1, U_2, SPKIWP(t, P)\}$$

is valid, then it will certainly satisfy all the equations in the verification algorithm. Without loss of generality, we can check the first three equations in the verification algorithm.

(1) $e(U_2, P) = e(Q_2, P_{pub})^{c_2} \cdot t_2$.
(2) $c_1 + c_2 = H_2(P, Q_1, Q_2, t_2, t_1, m, P_{pub})$.
(3) $e(Q_1, P_{pub})^{c_1} t_1 = e(U_1, P)$.

It is easy to see that the adversary has no difficulty in forging correct

$$\{c_1, c_2, t_1, t_2, U_2\}$$

such that they satisfy the first two of the above three equations. Therefore, the main contribution of the adversary is to choose an appropriate r (as indicated in the **Sign** algorithm) and compute a correct U_1 satisfying the third equation in order to accomplish the forgery . On the other hand, if the adversary can guess the value U_1 correctly, then the success probability is around $\frac{1}{q}$ (this is derived from step 2, 3 and 4 of the signing algorithm). Now we suppose the adversary is able to successfully compute correct U_1 with non-negligible probability

$$\varepsilon,$$

where $\varepsilon > \frac{1}{q}$. Therefore, the adversary can compute c_1 and r_1 with different values for the same U_1. That is, there exist c_{11}, r_{11}, and c_{12}, r_{12} respectively such that

$$\begin{aligned} U_1 &= c_{11} S_1 + r_{11} P_{pub} \\ &= c_{12} S_1 + r_{12} P_{pub} \end{aligned}$$

By the step (3) of verifying algorithm, they all satisfy

$$e(Q_1, P_{pub})^{c_{11}} e(T_{11}, P_{pub}) = e(U_1, P)$$

and

$$e(Q_1, P_{pub})^{c_{12}} e(T_{12}, P_{pub}) = e(U_1, P),$$

where $T_{11} = r_{11} P$ and $T_{12} = r_{12} P$. Therefore,

$$(c_{11} - c_{12}) S_1 = (r_{12} - r_{11}) P_{pub}.$$

Hence,

$$S_1 = (c_{11} - c_{12})^{-1} (r_{12} - r_{11}) P_{pub}.$$

Where we might as well assume $(c_{11} - c_{12})^{-1}$ exists. Therefore, the adversary can work out the private key of the signer.

However, this is contradicting to the assumption of the theorem that the private key is secure. Therefore, the statement of the theorem must be true. □

6 Performance Analysis

As to the performance of this new scheme, the main workload is involved by the verification algorithm. And it consists of one hash evaluation, two bilinear pairing calculations, two modular multiplications, and one modular addition, as well as two bilinear pairing pre-calculations.

In practical implementation, we can use some existing tools for these computations (for example, [15], [3], [2], [16]) including scalar multiplication, bilinear pairing evaluation, and hash function evaluation over elliptic curves. Also the interaction protocol is required on line for the committable part of verification, which can provide better security.

7 Conclusions

A committal deniable signature scheme has been proposed in this paper.The scheme is based on the elliptic curves and thus it has high complexity with short key size. Also a zero knowledge proof interaction protocol is involved in the verification process of the committable part of the signature.

References

[1] D.Boneh & M.Franklin, *Identity-based encryption from the Weil pairing,* Proceedings of CRYPTO 2001, Springer-verlag, LNCS 2139, 213-229, 2001.

[2] S.L.Barreto & Y.Kim, *Fast hashing onto elliptic curves over fields of characteristic-3,* Cryptology ePrint Archive, Report 2001/098.

[3] P.S.L.M.Barreto, H.Y.Kim, B.Lynn & M.Scott, *Efficient algorithms for pairing-based cryptosystems,* Advances in Cryptology-Crypto 2002, Springer-Verlag, LNCS 2442, 354-368, 2002.

[4] M.Blum, *Probability and Graph Theory in CS,* Lecture Notes online, http://www.cs.berkeley.edu/blum/.

[5] I.F.Blake, G.Seroussi & N.P.Smart, *Elliptic Curves in Cryptography,* London Mathematical Society Lecture Notes Series 265, Cambridge University Press, 1999.

[6] D.Chaum, *Blind Signatures for Untraceable Payments,* Advances in Cryptology, Proceedings of Crypto 82, D. Chaum, R.L. Rivest, & A.T. Sherman (Eds.), Plenum, 199-203.

[7] D.Chaum & H.van Antwerpen, *Undeniable Signatures,* Advances in Cryptology-CRYPTO'89, G. Brassard (Ed.), Springer-Verlag, 212-216.

[8] D.Chaum & E.van Heyst, *Group Signatures,*Advances in Cryptology EUROCRYPT'91, D.W. Davies (Ed.), Springer-Verlag, 257-265.

[9] L.Chen, K.Harrison, N.Smart & D.Soldera, *Applications of multiple trust authorities in pairing based cryptosystems,* Proc.InfraSec 2002, Springer, LNCS 2437, 260-275, 2002.

[10] D.Clark, *Encryption advances to meet Internet challenges,* Computer 33 (8), 20-24, 2000.

[11] R.Cramer & V.Shoup, *A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack,* Advances in Cryptology-CRYPTO'98, LNCS1462, 13-25, Springer-Verlag, 1998.

[12] R.Dupont & A.Enge, *Practical non-interactive key distribution based on pairings,* Cryptology ePrint Archive, Report 2002/136.

[13] W.Diffie & M.E.Hellman, *New Directions in Cryptography,* IEEE Transactions on Information Theory, Vol. 22(6), 207-221, 1976.

[14] FIPS 186. Digital Signature Standard. U.S. Department of Commerce/N.I.S.T, National Technical Information Service, Springfield, VA, 1994.

[15] K.Eisentraeger, K.Lauter & P.L.Montgomery, *An efficient procedure to double and add points on an elliptic curve,* Cryptology ePrint Archive, Report 2002/112.

[16] S. D. Galbraith, K. Harrison, & D. Soldera, *Implementing the Tate pairing,* Algorithmic Number Theory Symposium-ANTS-V, Springer-Verlag, LNCS 2369, 324-337, 2002.

[17] S.Goldwasser, S.Micali & R.Rivest, *A digital signature scheme secure against adaptive chosen-message attacks,* SIAM Journal of Computing, 17(2), 281-308, 1988.

[18] S.Han, W.K.Y.Yeung & J.Wang, *Identity-based confirmer signatures from pairings over elliptic curves,* ACM Proceedings on Electronic Commerce, 2003.

[19] B.A.Huberman, M.Franlin & T.Hogg, *Enchancing privacy and trust in electronic communities,* Proceedings of the ACM Eelcetronics Commerce 1999.

[20] A.Joux & K.Nguyen *Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups,* Cryptology ePrint Report 2001/003.

[21] A. Joux, *A one-round protocol for tripartite Diffie-Hellman,* Algorithm Number Theory Symposium - ANTS-IV, Springer-Verlag, LNCS 1838, 385-394, 2000.

[22] A.W.Knapp, *Elliptic curves,* Mathematical Notes 40, Princeton University Press, 1992.

[23] N.Koblitz, *Algebraic aspects of cryptography,* Algorithms and Computation in Mathematics 3, Springer-Verlag, 1998.

[24] B. Libert & Jean-Jacques Quisquater, *New identity based signcryption schemes from pairings,* Proceedings of IEEE Information Theory Workshop 2003, 2003.

[25] K.G. Paterson, *ID-based signatures from pairings on elliptic curves,* Electronics Letters, Vol. 38 (18) (2002), 1025-1026.

[26] M.Rosing, *Implementing elliptic curve cryptography,* Maning Publications Co., 1998.

[27] R.Rivest, A.Shamir & L.Adleman, *A method for obtaining Digital Signatures and Public Key Cryptosystems,* Communications of the ACM, Vol.21(2) 1978, 120-126.

[28] C.Schnorr, *Efficient signature generation by smart cards,* Journal of Cryptology, Springer-Verlag, 4(3), 239-252, 1991.

[29] N.P.Smart, *An identity based authenticated key agreement protocol based on the Weil pairing,* Electronics Letters, Vol 38, 630-632, 2002.