Efficient ID-based Signature Without Trusted PKG

Jingwei Liu[1], Rong Sun[2], Weidong Kou[2], and Xinmei Wang[2]

1. The Key Laboratory of Computer Network and Information Security, Ministry of Education, Xidian University, P.O. Box 119, 710071 Xi'an, China.
2. State Key Laboratory of Integrated Service Networks, Xidian University, P.O. Box 119, 710071 Xi'an, China.
jwliu@mail.xidian.edu.cn, rong_sun@hotmail.com, kou_weidong@yahoo.com.cn, xmwang@mail.xidian.edu.cn

Abstract. In this paper, we introduce the exact concept of ID-based signature without trusted *Private Key Generater*(PKG), which solves the key escrow problem through binding two public key with a same identity. In this scheme, PKG is prevented from forging a legal user's signature because he only generates the partially private key. Using Gap Diffie-Hellman (GDH) groups, we construct an efficient ID-based signature scheme without trusted PKG, which security relies on the hardness of the Computational Diffie-Hellman Problem (CDHP). More precisely, under the random oracle model, our scheme is proved to be secure against existential forgery on adaptively chosen message and ID attack, assuming CDHP is intractable. Our scheme not only eliminates the inherent key escrow problem but also has a higher efficiency than the existing schemes.

Keywords: ID-based Signature, Key Escrow, Gap Diffie-Hellman Group, Bilinear Pairing, Dishonest PKG.

1 Introduction

With the widely use of open networks such as the Internet, digital signatures have become one of the most important cryptographic tools. In traditional CA-based Cryptosystems (CBC), the public key of a signer is essentially a random bit string. This leads to a problem of how the public key is associated with the signer. In these CBC, the binding between public key and identity of the signer is obtained via a digital certificate, issued by a Trusted Third Party (TTP) called Certifying Authority (CA). The traditional CBC requires huge efforts in terms of computing time and storage to manage the certificates. To simplify the certificate management process, an ID-based Cryptosystem (IBC) based on integer factorization problem was first proposed by Shamir in 1984 [1], which allows a user to use his identity as the public key. For the simple key management procedure compared to CBC, IBC can be an alternative for CA-based public key systems in some occasions, especially in the case that efficient key management is essential to the cryptosystem.

1.1 Related Works

Lots of research on ID-based schemes has been proposed after the initial work of Shamir [1], but most of them are impractical for low efficiency. In [2] Guillou and Quisquater proposed a paradoxical ID-based signature using their interactive zero-knowledge protocol in [2].

Recently, the bilinear pairings, namely the Weil pairing and the Tate pairing of algebraic curves, have initiated some completely new fields in cryptography, making it possible to realize cryptographic primitives that were previously unknown or impractical [3, 4]. More precisely, they are important tools for construction of ID-based signature schemes(IBS). Plenty of ID-based signature schemes from bilinear pairings have been proposed in recent years [5–12]. An ID-based signature using pairings was first proposed by Sakai, Ohgishi and Kasahara in [5], however they did not present the security analysis in their work. Paterson [6] proposed an IBS scheme based on pairings with brief security arguments but without rigorous proof. A provably secure IBS was proposed by Hess in [7], which is secure against existential forgery under adaptively chosen message and fixed ID attacks. In 2003, Cha-Cheon [8] proposed an IBS scheme based Gap Diffe-Hellman groups. They provided a definition of security for IBS schemes called security against existential forgery under adaptively chosen message and ID attacks and proved their scheme secure. In ASIACRYPT 2003, Al-Riyami et al. [9] introduced a concept of certificateless public key cryptography (CL-PKC), a model for the use of public key cryptographic which avoids the inherent key escrow of IBC. In the same year, Chen et al. [10] proposed an IBS scheme without trusted PKG, eliminating the inherent key escrow problem. In [11], Gorantla and Saxena proposed an efficient certificateless signature scheme which is more efficient than the IBS in [9]. In [12], Al-Riyami et al. present another CL-PKE scheme whose security is proven to rest on the hardness of the Bilinear Diffie-Hellman Problem (BDHP) and that is more efficient than the original scheme [9]. In [13], the authors propose a new CLPKE scheme that does not depend on the bilinear pairings and give a security proof in the random oracle model.

1.2 Our Contributions

In this paper, we introduce the exact concept of ID-based signature without trusted PKG, which solves the key escrow problem through binding two public key with a same identity. In this scheme, PKG is prevented from positive attacking because he only generates the partially private key. Using Gap Diffie-Hellman (GDH) groups, we construct an efficient ID-based signature scheme without trusted PKG, which security relies on the hardness of the Computational Diffie-Hellman Problem (CDHP). More precisely, under the random oracle model, our scheme is proved to be secure against existential forgery on adaptively chosen message and ID attack, which is a natural ID-based version of the standard adaptively chosen message attack, assuming CDHP is intractable. Our scheme not only eliminates the inherent key escrow problem but also has

a higher efficiency than the existing schemes. We will discuss the security and efficiency analysis of our scheme in section 5.

The main contributions in this paper are listed as follows:

1. Propose the generic definitions and the basic models of ID-based signature without trusted PKG.
2. Construct an efficient ID-based signature without trusted PKG.
3. Provide the security proof of the ID-based signature without trusted PKG in the random oracle model.

The rest of the paper is organized as follows. In the next section, we briefly describe some preliminary works. In section 3, we introduce the definition and basic model of ID-based signatures without trusted PKG. An efficient ID-based signature scheme without trusted PKG is constructed in detail in section 4. In section 5, the security analysis of our scheme is discussed and the efficiency comparison between our scheme and the existing schemes is shown. Finally, the concluding remarks are given in Section 6.

2 Preliminary Works

In this Section, we will briefly describe the basic definition and properties of bilinear pairings and Gap Diffie-Hellman Group.

2.1 Bilinear Pairings

The bilinear pairings namely Weil pairing and Tate pairing of algebraic curves is defined as a map $e : G_1 \times G_1 \rightarrow G_2$ where G_1 is a cyclic additive group generated by P, whose order is a prime q, and G_2 is a cyclic multiplicative group of the same order q. Let a, b be elements of $(\mathbb{Z}/q\mathbb{Z})^{\times}$. We assume that the discrete logarithm problems (DLP) in both G_1 and G_2 are hard. A bilinear pairings has the following properties:

1. Bilinear: $e(aR, bS) = e(R, S)^{ab}$, $\forall R, S \in G_1$ and $a, b \in (\mathbb{Z}/q\mathbb{Z})^{\times}$. This can be related as $\forall R, S, T \in G_1, e(R + S, T) = e(R, T)e(S, T)$ and $e(R, S + T) = e(R, S)e(R, T)$;

2. Non-degenerate: There exists $R, S \in G_1$ such that $e(R, S) \neq I_{G_2}$, where I_{G_2} denotes the identity element of the group G_2;

3. Computable: There is an efficient algorithm to compute $e(R, S)$ for all $R, S \in G_1$.

2.2 Gap Diffie-Hellman Group

Let G_1 be a cyclic additive group generated by P, whose order is a prime q, assume that the inversion and multiplication in G_1 can be computed efficiently. We first introduce the following problems in G_1.

1. Discrete Logarithm Problem (DLP): Given two elements $R, S \in G_1$, to find an integer $n \in (\mathbb{Z}/q\mathbb{Z})^{\times}$, such that $S = nR$ whenever such an integer exists.

2. Computational Diffie-Hellman Problem (CDHP): Given P, aP, bP for a, $b \in (\mathbb{Z}/q\mathbb{Z})^{\times}$, to compute abP.

3. Decisional Diffie-Hellman Problem (DDHP): Given P, aP, bP, cP for a, b, $c \in (\mathbb{Z}/q\mathbb{Z})^{\times}$ to decide whether $c \equiv ab \bmod q$.

We call G_1 a Gap Diffie-Hellman Group if DDHP can be solved in polynomial time but there is no polynomial time algorithm to solve CDHP or DLP with nonnegligible probability. Such group can be found in supersingular elliptic curve or hyperelliptic curve over finite field, and the bilinear pairings can be derived from the Weil or Tate pairings.

3 Basic Models

3.1 Definitions

In this section, we introduce some basic definitions of ID-based signatures without trusted PKG. The parameters involved in the schemes are depicted in the following.

- a plaintext message space M: a set of strings over some alphabet.
- a identity space ID: a set of possible id.
- a signature space S: a set of possible signatures.
- a partially private key space X_1: a set of possible partially private keys for signature creation, which is generated by the signer himself.
- a partially private key space X_2: a set of possible partially private keys for signature creation, which is generated by PKG.
- a partially public key space Y_1: a set of possible partially public keys for signature verification, which is generated by the signer himself.
- a partially public key space Y_2: a set of possible partially public keys for signature verification, which is generated from the signer's identity and Y_1 by PKG.
- A PKG involved in ID-based signature scheme maybe play three roles:

1. *Trusted*, Just as traditional, the trusted PKG wouldn't collude with an adversary;

2. *Negatively Dishonest*, The PKG would release the signer's partially private key $x_2 \in X_2$ to an adversary;

3. *Positively Dishonest*, The PKG would generate a valid public tuple $\langle y_1', y_2' \rangle$ and a valid private couple $\langle x_1', x_2' \rangle$ binding the id of a signer for an adversary.

Definition 1. *A full ID-based signature without trusted PKG consists of five procedures (Parameter Setup, Extract, Sign, Verify, Trace)*:

- an efficient probabilistic algorithm **Parameter Setup**:

$$k \rightarrow \text{description of system parameters } \langle M, S, X_1, X_2, Y_1, Y_2, y_{PKG}, x_{PKG} \rangle$$

where k is a security parameter and y_{PKG} is PKG's public key generated from x_{PKG} that is PKG's master secret key.

- an efficient probabilistic algorithm **Extract**:

Extract includes three sub-algorithms:

$$\begin{cases} \mathbf{Y}_1\mathbf{Gen} : X_1 \to Y_1, \text{ which is denoted by } y_1 = \mathbf{Y}_1\mathbf{Gen}(x_1); \\ \mathbf{Y}_2\mathbf{Gen} : ID \times Y_1 \to Y_2, \text{ which is denoted by } y_2 = \mathbf{Y}_2\mathbf{Gen}(id, y_1); \\ \mathbf{X}_2\mathbf{Gen} : Y_2 \to X_2, \text{ which is denoted by } x_2 = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(y_2). \end{cases}$$

where $\mathbf{Y}_2\mathbf{Gen}$ is a one-way function to binding y_1, y_2 with id at one time.

- an efficient probabilistic signing algorithm **Sign**:

$$M \times X_1 \times X_2 \to S,$$

for any message $m \in M$ and private key $x_1 \in X_1$, $x_2 \in X_2$, we denote by

$$s = \mathbf{Sign}_{(x_1,x_2)}(m)$$

where $s \in S$.

- an efficient signature verification algorithm **Verify**:

$$M \times S \times ID \times Y_1 \times Y_2 \to \{\text{True, False}\},$$

for any message $m \in M$ and public key $y_1 \in Y, y_2 \in Y_2$, it is necessary

$$\mathbf{Verify}_{(id,y_1,y_2)}(m, s) = \begin{cases} \text{True, if } s = \mathbf{Sign}_{(x_1,x_2)}(m) \text{ and } y_2 = \mathbf{Y}_2\mathbf{Gen}(id, y_1) \\ \text{False, elsewise.} \end{cases}$$

- an efficient interactive "knowledge proof" algorithm **Trace**:

$$Y_2 \times ID \times Y_1 \to \{\text{Honest, Dishonest}\},$$

if, for PKG' positively dishonesty, there exist two triple $\langle x_2, id, y_1 \rangle$ and $\langle x_2', id, y_1' \rangle$ and $id \in ID$, $y_1, y_1' \in Y_1$, $x_2, x_2' \in X_2$, it is necessary

$$\mathbf{Trace}(id, y_1, x_2, y_1', x_2') = \begin{cases} \text{Dishonest, if } x_2 = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(\mathbf{Y}_2\mathbf{Gen}(id, y_1)) \\ \qquad\qquad \text{and } x_2' = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(\mathbf{Y}_2\mathbf{Gen}(id, y_1')) \\ \text{Honest, elsewise.} \end{cases}$$

To protect user's partially secret key x_2, algorithm Trace should be an interactive "knowledge proof" algorithm that is performed by an arbiter and a legal user.

3.2 Basic Model of ID-Based Signature without Trusted PKG

In the normal case, most of ID-based signature schemes often involve a trusted PKG to generate a user's private key with his identity that is the correspond public key, so everyone can verify a signature using the signer's identity information. This property make ID-based signature advantageous over the traditional signature scheme, as the key distribution is far simplified. It needs a list for some public system parameters of PKG instead of a public key directory for all users. However, these schemes suffer from inherent drawback of key escrow that

PKG knows all users' private keys. But this weakness will be overcome in our signature scheme.

Following *Definition 1*, the detailed implement process of ID-based signature without trusted PKG is depicted as follows.

Parameter Setup:

PKG first choose a secret parameter k and the public system parameters $\langle M, S, ID, X_1, X_2, Y_1, Y_2, y_{PKG}\rangle$. y_{PKG} is computed from x_{PKG}.

Extract:

1. A user chooses a partially private key $x_1 \in X_1$ randomly and computes the partially public key $y_1 = \mathbf{Y}_1\mathbf{Gen}(x_1), y_2 = \mathbf{Y}_2\mathbf{Gen}(id, y_1)$, where $y_1 \in Y_1, y_2 \in Y_2$ and id is the user's identity information. He sends $\langle id, y_1, y_2\rangle$ to PKG.

2. After receiving user's $\langle id, y_1, y_2\rangle$, PKG verifies whether $y_2 = \mathbf{Y}_2\mathbf{Gen}(id, y_1)$. If hold, PKG computes $x_2 = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(y_2)$ and sends it back to the user in a secret channel. Couple $\langle x_1, x_2\rangle$ is the private key and couple $\langle y_1, y_2\rangle$ is the public key. Otherwise, PKG aborts.

Sign:

The user signs a message m with his private key $\langle x_1, x_2\rangle$ using algorithm **Sign**.

$$s = \mathbf{Sign}_{(x_1,x_2)}(m)$$

Where s is the signature of the message m.

Verify:

Everyone can verify whether $\mathbf{Verify}_{(id,y_1,y_2)}(m, s) = \text{True}$.

Trace:

Suppose positively dishonest PKG forged a "valid" public key $\langle y_1', y_2'\rangle$ and private key $\langle x_1', x_2'\rangle$ for a id of a legal user, whose public key is $\langle y_1, y_2\rangle$ and private key is $\langle x_1, x_2\rangle$, which fit

$$\begin{cases} x_2 = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(\mathbf{Y}_2\mathbf{Gen}(id, y_1)) \\ x_2' = \mathbf{X}_2\mathbf{Gen}_{x_{PKG}}(\mathbf{Y}_2\mathbf{Gen}(id, y_1')) \end{cases}$$

Using the interactive "knowledge proof" algorithm **Trace**, the legal user can show the proof that x_2 is really his legal partially private key, which is signed by PKG' private key, to an arbiter. Identity id correspond to x_2 and x_2' together, so the PKG is proved to be positively dishonest

Here we need to point out that our ID-based signature scheme can really prevent a positively dishonest PKG from binding several public keys with a same id, but it cannot avoid a Negatively dishonest PKG revealing the partially private key of a user. Fortunately, the other partially private key still can protect the users' right. In section 5, we will give the strict security proof for this case.

3.3 Attack Model for ID-based Signature without Trusted PKG

In traditional CA-based signature scheme, the property of secure against existential forgery on adaptively chosen message attack is necessary. In this model [14, 15], an adversary wins the game if he outputs a valid pair of a message

and a signature, where he is allowed to ask the signer to sign any message except the output. Here we will introduce an attack model for ID-based signature without trusted PKG, similarly to [8, 15]. We say that an ID-based signature scheme, which consists of five algorithms *Parameter Setup, Extract, Sign, Verify* and *Trace* playing the same role as ours, 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 $\mathcal{S}$ in the following game:

1. $\mathcal{S}$ runs *Parameter Setup* firstly and gives the public system parameters to $\mathcal{A}$.

2. $\mathcal{A}$ can require the following queries:

(a) Hash function query. $\mathcal{S}$ computes the value of the hash function for the requested input and sends the value to $\mathcal{A}$.

(b) *Extract* query. Given an identity *id* and the public key $\langle y_1, y_2\rangle$, $\mathcal{S}$ returns the partially private key x_2 corresponding to *id*, which is obtained by running $X_2 Gen$ algorithm.

(c) *Sign* query. Given a triple $\langle id, y_1, y_2\rangle$ and a message m, $\mathcal{S}$ returns a signature which is obtained by running *Sign* algorithm.

3. $\mathcal{A}$ outputs $\langle m, s\rangle$, where m is a message, and s is the signature. A wins the game if s is a valid signature of m for *id* in the following three cases:

Case 1: $\mathcal{A}$ outputs $\langle\langle id, y_1, y_2\rangle, m, s\rangle$, where *id* is a legal identity, $\langle y_1, y_2\rangle$ is the correspond legal public key, $\langle id, y_1, y_2\rangle$ and m are not equal to the inputs of any query to *Extract* and *Sign* respectively.

Case 2: $\mathcal{A}$ outputs $\langle\langle id, y_1, y_2\rangle, m, s\rangle$, where *id* is a legal identity, $\langle y_1, y_2\rangle$ is the correspond legal public key, $\langle id, y_1, y_2\rangle$ is an input of query *Extract* and m are not equal to the inputs of query to *Sign*.

Case 3: $\mathcal{A}$ outputs $\langle\langle id, y_1', y_2'\rangle, m, s\rangle$, where *id* is a legal identity, $\langle y_1', y_2'\rangle$ is the forged public key, $\langle id, y_1', y_2'\rangle$ is an input of query to *Extract* and m is an input of query to *Sign*.

Using this attack model, we can reduce the security of ID-based signature without trusted PKG to the hardness of CDHP (Case 1 and Case 2) and DLP (Case 3) in section 5.

4 An Efficient ID-based Signature without Trusted PKG

Let $(G_1, +)$ and $(G_2, \cdot)$ denote cyclic groups of prime order q, $P \in G_1$ a generator of G_1 and let $e : G_1 \times G_1 \to G_2$ be a pairing which satisfies the properties of *Bilinear* and *Non-degenerate*.

We also assume that $e(R, S)$ can be easily computed while, for any given random $S \in G_1$ and $T \in G_2$, it should be infeasible to compute $R \in G_1$ such that $e(R, S) = T$. We remark that the pairing e is required to be symmetric. Furthermore we define the hash functions $H : \{0,1\}^* \to G_1^*$ and $h : \{0,1\}^* \times G_2^* \to (\mathbb{Z}/q\mathbb{Z})^\times$, where $G_1^* := G_1 \setminus \{0\}$. We also abbreviate $G_2^* := G_2 \setminus \{1\}$.

The efficient ID-based signature scheme without trusted PKG also consists of five algorithms, **Setup**, **Extract**, **Sign**, **Verify** and **Trace**. There are three parties in the system, the authority PKG, the signer and the verifier.

Setup : The PKG picks a random integer $s_{PKG} \in (\mathbb{Z}/q\mathbb{Z})^{\times}$, computes $Q_{PKG} = s_{PKG}P$ and publishes Q_{PKG} while s_{PKG} is kept secret. The signer also selects a random integer $s_1 \in (\mathbb{Z}/q\mathbb{Z})^{\times}$ as his partially secret key and computes $Q_1 = s_1 P$ as his partially public key.

Extract : This algorithm is performed by the PKG when a signer requests the secret key corresponding to his identity. Suppose the signer's identity is given by the string id which is the other partially public key. The other partially secret key of the identity is then given by $S_2 = s_{PKG}Q_2$ where $Q_2 = H(id, Q_1)$, which is computed by the PKG and given to the signer. For a signer, $\langle Q_1, Q_2 \rangle$ is his public key and $\langle s_1, S_2 \rangle$ is his private key.

The extraction step is typically done once for every identity and uses the same setup data for many different identities.

Sign: To sign a message m, the signer chooses a random integer $k \in (\mathbb{Z}/q\mathbb{Z})^{\times}$ and computes:

1. $r = e(Q_2, Q_{PKG})^k$
2. $v = h(m, r)$
3. $U = kS_2 - vs_1Q_2$

The signature is then the pair $(v, U) \in ((\mathbb{Z}/q\mathbb{Z})^{\times}, G_1)$.

Verify : On receiving a message m and signature $\langle v, U \rangle$ the verifier computes:

1. $r = e(U, P) \cdot e(Q_2, Q_1)^v$.
2. Accept the signature if and only if $v = h(m, r)$ and $Q_2 = H(id, Q_1)$.

It is straightforward to check that the verification equation holds for a valid signature.

Trace : Suppose PKG (or colludes with a dishonest user) wants to impersonate an honest user whose identity information is id. He (or they) can do as follows:

-PKG randomly chooses an integer $s_1' \in (\mathbb{Z}/q\mathbb{Z})^{\times}$ and let $Q_1' = s_1'P$, $Q_2' = H(id, Q_1')$ and $S_2' = s_{PKG}Q_2'$;

-He then performs the above signing protocol for the message m;

-Outputs $(v', U') \in ((\mathbb{Z}/q\mathbb{Z})^{\times}, G_1)$.

Because $v' = h(m, r')$ and $Q_2' = H(id, Q_1')$, where $r' = e(U', P) \cdot e(Q_2, Q_1)^v$, PKG forged a "valid" signature of the honest user. However, the user can provide a proof to convince that the signature is forged by PKG, which is similar to CA-based systems. He firstly sends Q_1 to the arbiter, and then provides a "knowledge proof" that he knows $S_2 = s_{PKG}H(id, Q_1)$: the arbiter randomly chooses a secret integer $\alpha \in (\mathbb{Z}/q\mathbb{Z})^{\times}$ and sends αP to the user; the user then computes $e(S_2, \alpha P)$. If the equation $e(S_2, \alpha P) = e(Q_2, Q_{PKG})^{\alpha}$ holds, the arbiter deduces PKG dishonest because identity id corresponds to $\langle Q_1, Q_2 \rangle$ and $\langle Q_1', Q_2' \rangle$ and the master-key s_{PKG} is only known to PKG.

5 Security and Efficiency Analysis of Our Scheme

5.1 Security

In this section, we will discuss the security of our signature schemes in the random oracle model [16].

THEOREM 5.1. *(**Correctness**) Our scheme in section 4 is correct.*

proof: If

$$s = \mathbf{Sign}_{(s_1,S_2)}(m) = \langle v, U \rangle$$

where

$$\begin{aligned} r &= e(Q_2, Q_{PKG})^k \\ v &= h(m, r) \\ U &= kS_2 - vs_1Q_2 \end{aligned}$$

then

$$\begin{cases} v = h(m, e(U, P) \cdot e(Q_2, Q_1)^v) \\ Q_2 = H(id, Q_1) \end{cases} \Leftrightarrow \mathbf{Verify}_{(id,Q_1,Q_2)}(m, v, U) = \text{True}.$$

This theorem is proved. □

To prove the *Unforgeability* of our ID-based signature scheme without trusted PKG, we firstly introduce an important conclusion – *Forking Lemma* [15]. It gives a reductionist security proof for triplet ElGamal-family signature schemes which produce a signature (σ_1, h, σ_2) on an input message m, where σ_1 takes its value randomly from a large set, h is the hash of m and σ_1, and σ_2 depends only on σ_1, m and h.

LEMMA 5.2. *(**Forking Lemma**) Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and R the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 10(R+1)(R+Q)/2^k$ (where k is a security parameter), a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{A}$ replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time $T' \leq 120686QT/\varepsilon$.*

THEOREM 5.3. *(**Unforgeability**) Our scheme in section 4 is unforgeable under a chosen message attack in the random oracle model, assuming the hardness of the CDHP.*

proof: The proof is referred to the proof of unforgeability of the signature scheme by Pointcheval and Stern [14], and makes use of the forking lemma [14, 15]. Firstly, we note that our ID-based signature scheme in section 4, given the input message m, produces a valid signature of the form triple $\langle r, v, U \rangle$, where $r = e(Q_2, Q_{PKG})^k$ randomly takes its values in G_2 for k is randomly selected in $(\mathbb{Z}/q\mathbb{Z})^\times$, v is the hash value of $\langle m, r \rangle$ and U depends on r, the message m and v.

Because our ID-based signature scheme does not use a trusted PKG, the adversary $\mathcal{A}$ maybe forge a signature collude with a dishonest PKG. Thus there have two cases to discuss.

1. $\mathcal{A}$ forges a valid signature with no help of a trusted PKG;

2. $\mathcal{A}$ forges a valid signature with the help of a negatively dishonest PKG.

Case 1: We suppose that H and h are random oracles, and there exists a probabilistic polynomial time Turing machine $\mathcal{A}$ whose input only consists of public data. We assume that $\mathcal{A}$ can make Q_H queries to the random oracle H, Q_h queries to the random oracle h and R queries to the signing oracle $Sign$. Within time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 10(R+1)(R+Q_h)/2^q$ (where q is a security parameter), a valid signature $\langle m, \langle v, U\rangle\rangle$.

simulation: $\mathcal{S}$ gives the parameters $\langle G_1, G_2, Q_{PKG}, P, H, h\rangle$ and $\langle Q_1, Q_2\rangle$ to $\mathcal{A}$. $\mathcal{S}$ tries to simulate the challenger by simulating all the oracles to gain $\langle s_1 Q_2, kS_2\rangle$ for a fixed k, which is the random integer in $(\mathbb{Z}/q\mathbb{Z})^\times$ used in algorithm **Sign**, to have the ability of signing arbitrary message m as the real signer. $\mathcal{A}$ can query as follows:

H-Queries: $\mathcal{A}$ can query the random oracle H at any time. $\mathcal{S}$ simulates the random oracle by keeping list of couples $\langle \sum_i, Q_{(2,i)}\rangle$ which is called the H-List, where $\sum_i$ is a couple of $\langle id_i, Q_{(1,i)}\rangle$. When the oracle is queried with an input $\sum$, $\mathcal{S}$ responds as follows:

1. If the query $\sum$ is already on the H-List in the couple $\langle \sum, Q_{(2,i)}\rangle$, then $\mathcal{S}$ outputs $Q_{(2,i)}$.

2. Otherwise $\mathcal{S}$ selects a random $Q_2 \in G_1^*$, outputs Q_2 and adds $\langle \sum, Q_2\rangle$ to the H-List.

Extract-Queries: $\mathcal{A}$ can query the partially private key for any identity id_i and the public key $\langle Q_{(1,i)}, Q_{(2,i)}\rangle$. If $Q_{(2,i)} \neq H(id_i, Q_{(1,i)})$, $\mathcal{S}$ returns invalid. Otherwise, it outputs the partially private key $S_{(2,i)}$ corresponding to id_i which is obtained by running *Extract* algorithm.

h-Queries: $\mathcal{A}$ can query the random oracle h at any time. $\mathcal{S}$ simulates the random oracle by keeping list of couples $\langle \sum_i, v_i\rangle$ which is called the h-List, where $\sum_i$ is a couple of $\langle m_i, r_i\rangle$. When the oracle is queried with an input $\sum$, $\mathcal{S}$ responds as follows:

1. If the query $\sum$ is already on the h-List in the couple $\langle \sum, v_i\rangle$, then $\mathcal{S}$ outputs v_i.

2. Otherwise $\mathcal{S}$ selects a random $v \in (\mathbb{Z}/q\mathbb{Z})^\times$, outputs v and adds $\langle \sum, v\rangle$ to the h-List.

Sign-Queries: $\mathcal{S}$ simulates the signature oracle by accepting signature queries of the message m to be signed. $\mathcal{S}$ answers the query as follows:

1. $\mathcal{S}$ picks a random $U \in G_1$ `and` $v \in (\mathbb{Z}/q\mathbb{Z})^\times$ which v isn't equal to some previous output for the h oracle.
2. $\mathcal{S}$ computes $r = e(U, P) \cdot e(Q_2, Q_1)^v$. If $\sum = \langle m, r\rangle$ is some previous input for the h oracle, then return to step 1.
3. $\mathcal{S}$ adds a couple $\langle \sum, v\rangle$ to h-List.
4. $\mathcal{S}$ outputs $s = \langle v, U\rangle$ as the signature for message m.

NOTE: Here we must check whether the distributions of real signature δ and forged signature δ' are same.

$$\begin{cases} \delta = \{(r, v, U) | k, v \in (\mathbb{Z}/q\mathbb{Z})^{\times}, r = e(Q_2, Q_{PKG})^k, U = kS_2 - vs_1Q_2\} \\ \delta' = \{(r, v, U) | U \in G_2, v \in (\mathbb{Z}/q\mathbb{Z})^{\times}, r = e(U, P) \cdot e(Q_2, Q_1)^v\} \end{cases}$$

First we compute the probability of a real signature signed using secret key,

$$\begin{aligned} & \Pr_{\delta}[(r, v, U) = (\varepsilon, \beta, \gamma)] \\ = & \Pr_{k \neq 0, v}[r = e(Q_2, Q_{PKG})^k = \varepsilon, v = \beta, U = kS_2 - vs_1Q_2 = \gamma] \\ = & \frac{1}{q(q-1)}. \end{aligned}$$

The probability of a forged signature is

$$\begin{aligned} & \Pr_{\delta'}[(r, v, U) = (\varepsilon, \beta, \gamma)] \\ = & \Pr_{v, U}[v = \beta, U = \gamma, r = e(U, P) \cdot e(Q_2, Q_1)^v = \varepsilon \neq 1] \\ = & \frac{1}{q(q-1)}. \end{aligned}$$

So the triple $\langle r, v, U \rangle$ can be simulated without knowing the secret key, with an indistinguishable distribution probability. Thus, the signing oracle simulated by $\mathcal{S}$ is high quality, and thereby $\mathcal{A}$ is very satisfied with the Sign-Queries' answer. He can fully exert his forgery ability.

Output: Finally, with non-negligible probability, $\mathcal{A}$ outputs a signature $s = \langle r, v, U \rangle$ with a message $m \in M$, where $\mathbf{Verify}_{(id, Q_1, Q_2)}(m, v, U) = True$, in the case that $\mathcal{A}$ produces v through h queries but no $Extract$ query with input $\langle id, Q_1, Q_2 \rangle$, no $Sign$ query with input m were made by $\mathcal{A}$.

Now $\mathcal{S}$ can play the simulation twice so that $\mathcal{A}$ should produce two valid signature $s = \langle r, v, U \rangle$ and $s' = \langle r, v', U' \rangle$ with $v \neq v'$. Then we have the following equations.

$$\begin{cases} U = kS_2 - vs_1Q_2 \\ U' = kS_2 - v's_1Q_2 \end{cases} \Leftrightarrow \begin{cases} R_1 = s_1Q_2 = (v' - v)^{-1}(U - U') \\ R_2 = kS_2 = U + v(v' - v)^{-1}(U - U') \end{cases}$$

Using $\langle R_1, R_2 \rangle$, $\mathcal{S}$ can sign a valid signature $\langle v, U \rangle$ for any message m with a fixed k (Note: Here r is also fixed for $r = e(Q_2, Q_{PKG})^k$.), where $v = h(m, r)$ and $U = R_2 - vR_1$, just as the real signer signs the signature using his private key $\langle s_1, S_2 \rangle$.

From above equations $\mathcal{S}$ can solve the hard CDHP:

$$\begin{cases} Q_1 = aP = s_1P \\ Q_2 = bP \end{cases} \Rightarrow abP = R_1 = s_1Q_2$$

within expected time less than $\frac{120686 \times 2^q \times Q_h T}{10 \times (R+1) \times (R+Q_h)}$. This contradicts the hardness of the CDHP.

Case 2: The random oracle assumption is same as before. We suppose there exists a probabilistic polynomial time Turing machine $\mathcal{A}$ whose input only consists of public data. We assume that $\mathcal{A}$ can make Q_H queries to the random oracle H, Q_h queries to the random oracle h and R queries to the signing oracle $Sign$. But, in this case, $\mathcal{A}$ has a piece of additional information S_2 from the dishonest PKG.

simulation: $\mathcal{S}$ gives the parameters $\langle G_1, G_2, Q_{PKG}, P, H, h\rangle$ and $\langle Q_1, Q_2, S_2\rangle$ to $\mathcal{A}$. $\mathcal{S}$ tries to simulate the challenger by simulating all the oracles to gain s_1Q_2 to have the ability of signing arbitrary message m as the real signer. $\mathcal{A}$ can query as in case 1.

Output: Finally, with non-negligible probability, $\mathcal{A}$ output a signature $s = \langle r, v, U\rangle$ with a message $m \in M$, where $\mathbf{Verify}_{(id,Q_1,Q_2)}(m, v, U) = True$, in the case that $\mathcal{A}$ produces v through h queries, S_2 through $Extract$ query with input $\langle id, Q_1, Q_2\rangle$ but no $Sign$ query with input m were made by $\mathcal{A}$.

Now $\mathcal{S}$ can play the simulation twice so that $\mathcal{A}$ should produce two valid signature $s = \langle r, v, U\rangle$ and $s' = \langle r, v', U'\rangle$ with $v \neq v'$. Then we have the following equations.

$$\begin{cases} U = kS_2 - vs_1Q_2 \\ U' = kS_2 - v's_1Q_2 \end{cases} \Rightarrow \quad R_1 = s_1Q_2 = (v' - v)^{-1}(U - U')$$

Using $\langle R_1, S_2\rangle$, $\mathcal{S}$ can sign a valid signature $\langle v, U\rangle$ for any message m, where $v = h(m, e(Q_2, Q_{PKG})^k), U = kS_2 - vR_1$ and k is chose randomly in $(\mathbb{Z}/q\mathbb{Z})^\times$, just as the real signer uses $\langle s_1, S_2\rangle$. From above equations $\mathcal{S}$ can solve the CDHP:

$$\begin{cases} Q_1 = aP = s_1P \\ Q_2 = bP \end{cases} \Rightarrow abP = R_1 = s_1Q_2$$

within expected time less than $\frac{120686\times 2^q \times Q_h T}{10\times(R+1)\times(R+Q_h)}$. This contradicts the hardness of the CDHP. □

Before THEOREM 5.4, we first introduce the definition-Traceability.

Definition 2. In ID-based signature, if PKG forged a "valid" public key $\langle Q_1', Q_2'\rangle$ and private key $\langle s_1', S_2'\rangle$ for a id of a legal user, whose public key is $\langle Q_1, Q_2\rangle$ and private key is $\langle s_1, S_2\rangle$, the user can provides a proof, using the interactive "knowledge proof" algorithm Trace, to the arbiter that PKG is positively dishonest, we call the scheme has the property of Traceability.

THEOREM 5.4. ***(Traceability)*** *Our scheme in section 4 is traceable.*

proof: We suppose that H is a random oracle, and there exists a probabilistic polynomial time Turing machine $\mathcal{A}$ whose input only consists of public data. We assume that $\mathcal{A}$ can make following two queries.

H-Queries: is defined as in THEOREM 5.3.

Extract-Queries: is defined as in THEOREM 5.3.

Output: Finally, with non-negligible probability, $\mathcal{A}$ outputs S_2' being a "valid" partially private key for $\langle id, Q_1', Q_2'\rangle$, which fit

$$\begin{cases} Q_2' = H(id, Q_1') \\ S_2' = s_{PKG}Q_2' \end{cases}$$

but the legal partially private key and public key of id is S_2 and $\langle Q_1, Q_2\rangle$. One of the following two cases holds:

Case 1. PKG is really trusted, which means that no *Extract* query with input $\langle id, Q_1, Q_2\rangle$ and $\langle id, Q_1', Q_2'\rangle$, but adversary can always compute $S_2' = s_{PKG}Q_2'$ for $Q_2' = H(id, Q_1')$.

However, this comes into collision with the hardness of DLP, thus we are sure that the next case must have occurred.

Case 2. PKG is positively dishonest, triple $\langle id, Q_1', Q_2'\rangle$ must have been an input of *Extract* query:

$$\begin{cases} Q_2' = H(id, Q_1') \\ S_2' = s_{PKG}Q_2' \end{cases}$$

.

The user can provides a proof, using the interactive "knowledge proof" algorithm *Trace*, that he knows $S_2 = s_{PKG}H(id, Q_1)$. The arbiter deduces PKG dishonest because identity id corresponds to $\langle Q_1, Q_2\rangle$ and $\langle Q_1', Q_2'\rangle$ and the master-key s_{PKG} is only known to PKG. □

5.2 Efficiency

In our ID-based signature without trusted PKG, there is some pre-computation, which makes our scheme efficient. Before signing any message, the user can pre-compute $e(Q_2, Q_{PKG})$ for $r = e(Q_2, Q_{PKG})^k$ in step 1 and s_1Q_2 for $U = kS_2 - vs_1Q_2$ in step 3. Similarly, before verifying any signature, the verifier can pre-compute $e(Q_2, Q_1)$ for $r = e(U, P)e(Q_2, Q_1)^v$ in step 1 and $Q_2 = H(id, Q_1)$.

Because our ID-based signature scheme has much pre-computation , so it has higher efficiency than the existing schemes. Executive efficiency comparison between our scheme and the existing schemes is given in Table 1. In the table 1, " e " denotes the number of pairing operation, " E_{G_2} " denotes the number of exponentiation in G_2, " M_{G_1} " denotes the number of multiplication in G_1, " M_{G_2} " denotes the number of multiplication in G_2, " A_{G_1} " denotes the number of addition in G_1, " H " denotes the number of H hash operation, " h " denotes the number of h hash operation, "$A_{(\mathbb{Z}/q\mathbb{Z})^\times}$" denotes the number of addition in $(\mathbb{Z}/q\mathbb{Z})^\times$. In the Table 1, only for signing, our scheme performs as bad as [9] and worse than [10, 11], but the verify algorithm is simplified very much. For a simple signature scheme, in general, is one-time-sign and multi-time-verify, so our scheme obtains a higher efficiency in the whole.

6 Conclusions

In this paper, we introduce the exact concept of ID-based signature without trusted PKG, which solves the key escrow problem through binding two public key with a same identity. In this scheme, PKG is prevented from forging a legal user's signature because he only generates the partially private key. Using Gap Diffie-Hellman (GDH) groups, we construct an efficient ID-based signature

Algorithm	Pre-Sign	Sign
[AP03]	$1e$	$1E_{G_2}, 2M_{G_1}, 1h, 1A_{G_1}$
[CZK03]	/	$3M_{G_1}, 1H, 1h, 1A_{G_1}, 1A_{(\mathbb{Z}/q\mathbb{Z})^\times}$
[GS05]	/	$2M_{G_1}, 1h, 1A_{G_1}, 1A_{(\mathbb{Z}/q\mathbb{Z})^\times}$
Our Scheme	$1e, 1M_{G_1}$	$1E_{G_2}, 2M_{G_1}, 1h, 1A_{G_1}$
Algorithm	Pre-Verify	Verify
[AP03]	/	$4e, 1E_{G_2}, 1M_{G_2}, 1h$
[CZK03]	$1H$	$4e, 1H, 1h, 1M_{G_1}, 2A_{G_1}$
[GS05]	/	$3e, 1h, 1M_{G_2}, 1M_{G_1}, 1A_{G_1}$
Our Scheme	$1e, 1H$	$1e, 1E_{G_2}, 1M_{G_2}, 1h$

Table 1. Efficiency Comparison with Our Scheme and the Existing Schemes

scheme without trusted PKG, which security relies on the hardness of the Computational Diffie-Hellman Problem (CDHP). More precisely, under the random oracle model, our scheme is proved to be secure against existential forgery on adaptively chosen message and ID attack, which is a natural ID-based version of the standard adaptively chosen message attack, assuming CDHP is intractable. From the comparison, we conclude that our scheme not only eliminates the inherent key escrow problem but also has a higher efficiency than the existing schemes, which make it have many applications in different environment.

We have taken the directions for future research to transform the basic scheme to various more efficient transmutations.

7 Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant No. 90304008, No. 60373104 and No. 90604009.

References

1. A. Shamir, *Identity-based Cryptosystems and Signature Schemes*, In Advances in Cryptology-Crypto'84, LNCS vol. 196, Springer-Verlag, 1984, pp. 47-53.
2. L. C. Guillou and J. -J. Quisquatar, *A "paradoxical" identity-based signature scheme resulting from zero knowledge.*, In Advances in Cryptology-Crypto'88, LNCS vol. 403, Springer-Verlag, 1990, pp. 216-231.
3. D. Boneh. and M. Franklin, *Identity-based Encryption from the Weil pairing*, Advances in Crptology-Crypto'01, LNCS vol. 2139, Springer-Verlag, 2001, pp. 213-229.
4. D. Boneh, B. Lynn and H. Shacham, *Short Signatures from the Weil Pairing*, In Advances in Crptology-Asiacrypt'01, LNCS vol. 2248, Springer-Verlag, 2001, pp. 514-532.
5. R. Sakai, K. Ohgishi and M. Kasahara, *Cryptosysytems based on pairing*, In Symposium on Cryptography and Information Security-SCIS'00, Okinawa, Japan, 2000, pp. 26-28.

6. K. G. Paterson, *Id-based signatures from pairings on elliptic curves*, Electronics Letters, 38(18):1025–1026, 2002.
7. F. Hess, *Efficient Identity Based Signature Schemes Based on Pairings*, In Selected Areas in Cryptography-SAC'02, LNCS vol. 2595, Springer-Verlag, 2003, pp. 310-324.
8. J. C. Cha and J. H. Cheon, *An Identity-Based Signature from Gap Diffie-Hellman Groups*, In Public Key Cryptography-PKC'03, LNCS vol. 2567, Springer-Verlag, 2003, pp. 18-30.
9. S. S. Al-Riyami and K. G. Paterson, *Certificateless Public Key Cryptography*, In ASIACRYPT 2003, LNCS vol. 2894, Springer-Verlag, 2003, pp. 452-473.
10. X. Chen, F. Zhang, K. Kim, *A New ID-based Group Signature Scheme from Bilinear Pairings*, In Proceedings of WISA'03, LNCS vol. 2908, Springer-Verlag, 2003, pp.585-592.
11. M. C. Gorantla and A. Saxena, *An Efficient Certificateless Signature Scheme*, CIS 2005, Part II, LNAI vol. 3802, 2005, pp. 110-116.
12. Sattam S. Al-Riyami and Kenneth G. Paterson, *CBE from CL-PKE: A Generic Construction and Efficient Schemes*, Public Key Cryptography - PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, LNCS vol. 3386, Les Diablerets, Switzerland, January 23-26, 2005, pp.398.
13. Joonsang Baek, Reihaneh Safavi-Naini and Willy Susilo, *Certificateless Public Key Encryption Without Pairing*, Information Security: 8th International Conference, ISC 2005, LNCS vol. 3650, Singapore, September 20-23, 2005, pp. 134.
14. D. Pointcheval and J. Stern, *Security proofs for signature schemes*, In Advances in Cryptology - EUROCRYPT 1996, LNCS vol. 1070, Springer-Verlag, 1996, pp. 387-398.
15. D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, In Journal of Cryptology, vol. 13(2000), pp. 361-396.
16. M. Bellare, and P. Rogaway, *Random oracles are practical: a paradigm for designing efficient protocols*, In Proc. of the 1st CCCS, ACM press, 1993, pp. 62-73.