Efficient ID-based Threshold Signature Schemes without Pairings

Jun Shao, Zhenfu Cao, and Licheng Wang

Department of Computer Science and Engineering
Shanghai Jiao Tong University
200030, Shanghai, China P.R.C
chn.junshao@gmail.com, caozf@cs.sjtu.edu.cn, wanglc@sjtu.edu.cn

Abstract. The focus of this paper is to design an efficient and secure solution addressing the key escrow problem in ID-based signature schemes, i.e., the Private Key Generator (PKG) knows the user's private key, which damages the essential requirement–"non-repudiation" property of signature schemes. In this paper, we proposed two ID-based threshold signature schemes, which both reach Girault's trusted level 3, and in which there exists only one PKG in our ID-based threshold signature schemes. In particular, the second scheme has another good property: it does not require trusting any particular party at any time.
Compared with the previous schemes, our schemes do not need to compute pairings, which make them be more efficient than those schemes. Furthermore, our ID-based signature schemes increase the availability of the signing agency and the difficulty for the adversary to learn the private key.

1 Introduction

Certificate-based cryptography allows the user to use an arbitrary string, unrelated to his identity, as his public key. When another user wants to use this public key, she must obtain an authorized certificate that contains this public key. This creates the certificate management problem. For reasons of efficiency and convenience, it is desirable to design a signature scheme without certificate management.

To address this problem, Shamir introduced the concept of ID-based public key cryptography [Sha84], in 1984. In this kind of public key cryptography, the user's public key is the user's identity information, e.g., the email address, while the private key is computed from the public key by a Private Key Generator (PKG) who has knowledge of a master secret. As a result, the certificate management problem can be eliminated.

Shamir, in his breakthrough work [Sha84], proposed the first ID-based signature scheme. Since then, many ID-based signature schemes [SOK00,Hes02,Pat02,CC03] and ID-based encryption schemes [BF01,Coc01,BF03] were proposed. However, there are some drawbacks in these schemes. The most criticism against these schemes, called key escrow problem, is that the PKG knows the private key of

all users, so he is able to impersonate any user. In particular, in ID-based signature schemes, it removes the essential requirement, "non-repudiation", of digital signature schemes.

To address this key escrow problem, many researchers [BF03,BZ04a,BZ04b,CZKK04] suggested extending ID-based cryptography to be ID-based threshold cryptography. Using an ID-based threshold signature scheme, digital signatures can be produced by a group of players rather than by one party. Compared with the regular ID-based signature schemes where the signer is single entity which holds the private key, in ID-based threshold signature schemes the private key is shared by a group of n players. In order to produce a valid signature on a given message m, the number of the participant players must attain the given threshold value, the signature can be created. More precisely, a typical (t, n) ID-based threshold signature scheme follows the three basic properties:

- Any t or more players in the group can cooperate with each other to generate a valid group signature, while they don't reveal any information about their sub-secret keys or the private key.
- Any $t-1$ or fewer players in the group cannot create a valid group signature, even after producing many signatures on different messages.
- Any verifier can verify the group signature with only knowing the group identity and the public key of the PKG. In other words, the signature is produced in an ID-based threshold signature scheme is the same as if produced in a regular ID-based signature scheme. In particular, verification of the signature is dependent of the way the signature generation is implemented.

Besides removing the key escrow problem, an (t, n) ID-based threshold signature scheme has another two advantages: (1) increasing the availability of the signing agency, since if only t players perform the scheme honestly, the signature can be generated successfully; (2) increasing the difficulty for the adversary to learn the private key, since only the adversary corrupts t or more players, he can learn the private key.

1.1 Previous Work

In [BF01,BF03], Boneh and Franklin proposed the first ID-based threshold cryptosystem. In their scheme, the PKG's master key is shared among a number of PKGs. However, this method has the following disadvantages: (1) it requires each PKG not to be closed after key generation, which is against Shamir's original proposal of ID-based cryptography; (2) it imposes heavy loads on users to authenticate themselves to the multiple PKGs; (3) the value of threshold is not flexible for users. To solve these problems, Baek and Zheng [BZ04a,BZ04b] proposed another ID-based threshold cryptosystem. In their schemes, it shares a private key associated with an identity rather than sharing a master key of the PKG. However, their schemes still suffer from the key escrow problem, that is, the PKG knows the private key. In [CZKK04], by combining the advantages of both Certificate-based public key cryptography and ID-based public key cryptography, Chen *et al.* proposed a new ID-based threshold signature scheme, which

not only solves the key escrow problem, but also removes the disadvantages of Boneh and Franklin's method. To our best knowledge, though there exists a particular party knows the private key in Chen *et al.*'s scheme, it is considered as the best solution among the existing schemes. In this paper, we get better solutions: more efficient (in terms of computational complexity) and more secure (no particular party knows the private key).

1.2 Our Contribution

We present two ID-based threshold signature schemes which enjoy the following properties.

Provable security: Based on the Schnorr's signature scheme [Sch91], we first propose a new ID-based signature scheme without bilinear pairings, which is provably secure in the random oracle model and achieves the Girault's trusted level 3 [Gir91] (i.e., the PKG does not know (or cannot easily compute) the users private keys. Moreover, it can be proved that the PKG generated false public keys of users if it does so.). Based on the proposed ID-based signature scheme, we propose two provably secure ID-based threshold signature schemes.

Efficiency: Since there is no bilinear pairings in our schemes, our schemes are more efficient than the existing schemes. We will give the performance evaluation in Section 7.

Flexible thresholds: The value of threshold is up to the users.

Assumed trust: In our first ID-based threshold signature scheme, the group manager should be trusted, since he knows the private key. While our second ID-based threshold signature scheme does not require trusting any particular party at any time, including during the generation of private key.

Limited power of the PKG: In our both ID-based threshold signature schemes, the only thing the PKG can do is to cooperate with the user to generate the user's private key. However, the PKG cannot learn the user's private. If the PKG generates the user's private key by himself, it can be detected.

1.3 Organization

Section 2 introduces the model and definitions for ID-based threshold signatures and their security. Section 3 recalls the Schnorr's signature scheme. Section 4 describes some of the existing tools in the literature that we use in our solutions. Section 5 shows our provably secure ID-based signature scheme without pairings. Section 6 presents our two provably secure ID-based threshold signature schemes. Section 7 discusses the efficiency of our ID-based threshold signature schemes. Finally, Section 8 gives the conclusion.

2 Model and Definitions

In this section we review the communication model and the definitions of secure ID-based threshold signature scheme.

Communication model. The players in our schemes include a set of n players who are connected by an authenticated broadcast channel. In addition, all players are capable of private point-point communication over secure channels. Finally, we work in a synchronous communication model. These assumptions allow us to focus on high-level descriptions of the protocols. For the details, the reader can refer to [GJKR96][Section 2].

The adversary. Our (t, n) ID-based threshold signature schemes assume a non-adaptive adversary (static adversary) who may corrupt up to $t-1$ players in advance of protocol execution. The adversary has access to all information available to the corrupted players, including their sub-secrets, messages they receive, and messages broadcast to all players. Furthermore, the adversary can cause player to deviate arbitrarily from the protocol.

ID-based signature scheme. Recall that ID-based signature scheme $\mathcal{S}$ is consisted of four random protocols: `Setup`, `Extract`, `Sign`, `Verify`. The `Setup` protocol takes as input the security parameter k, and generates *params* (system parameters) and *master-key*. The `Extract` protocol takes as input a user's ID and the master key, and returns the user's private key, denoted by d_{ID}. The `Sign` protocol signs messages using the user's private key and the `Verify` protocol verifies signatures using the user's ID and *params*. However, in order to achieve Girault's trusted level 3, we modify the `Extract` protocol slightly in this paper, i.e., add the data chosen by the user to the input.

The notion of security for ID-based signature scheme was formally defined in [CC03]. The following definition captures the notion: existential unforgeability against adaptively chosen identity and message attack.

Definition 2.1. *We say that an ID-based signature scheme $\mathcal{S}$(`Setup`, `Extract`, `Sign`, `Verify`) is unforgeable if no adversary who is given the public key of the PKG generated by `Setup`, the private keys of k_1 identities $ID_{i_1}, \cdots, ID_{i_{k_1}}$ adaptively chosen, and the signatures of k_2 tuples $(ID_{j_1}, m_{j_1}), \cdots, (ID_{j_{k_2}}, m_{j_{k_2}})$ adaptively chosen, can produce the signature on a new message m under the identity ID (i.e., $ID \notin \{ID_{i_1}, \cdots, ID_{i_{k_1}}\}$, $(ID, m) \notin \{(ID_{j_1}, m_{j_1}), \cdots, (ID_{j_{k_2}}, m_{j_{k_2}})\}$) with non-negligible probability.*

ID-based threshold signature scheme. Like ID-based signature scheme, ID-based threshold signature scheme $\mathcal{TS}$ is also consisted of four random protocols: `Setup`, `Thresh-Extract`, `Thresh-Sign`, `Verify`. Furthermore, it should achieve Girault's trusted level 3.

Setup. Identical to that in ID-based signature scheme.

Thresh-Extract. According to whether there exists a particular party knowing the private key, this protocol can be categorized as *with* or *without* a group manager. In the former one, there are a PKG, a group manager, and n players. Firstly, the group manager gets the private key through an interactive protocol with the PKG. Secondly, the group manager distributes the private key to the n players through some verifiable secrete sharing scheme. In the latter one, there are only a PKG and n players. The private key is jointly generated by the PKG and these n players. In both of two, the public key is always the identity ID of these n players.

Thresh-Sign. It is the distributed signature protocol. The private input of player P_i is his share of the private key. The public key inputs consist of a message m and the identity ID of these n players. The output of the protocol is a signature for message m under the identity ID.

Verify. Identical to that in ID-based signature scheme.

From the verifier's viewpoint, ID-based threshold signature scheme is the same as ID-based signature scheme.

Secure ID-based threshold signature scheme. Following the idea in [GJKR96], the definition of security for ID-based threshold signature scheme includes both *unforgeability* and *robustness*.

Definition 2.2. *We say that a (t, n) ID-based threshold signature scheme $\mathcal{TS}$=(`Setup`, `Thresh-Extract`, `Thresh-Sign`, `Verify`) is unforgeable, if no adversary who corrupts at most $t-1$ players, and is given the view of $Thresh-Extract$ on input k_1 identities $ID_{i_1}, \cdots, ID_{i_{k_1}}$ adaptively chosen, and of `Thresh-Sign` on input k_2 tuples $(ID_{j_1}, m_{j_1}), \cdots, (ID_{j_{k_2}}, m_{j_{k_2}})$ adaptively chosen, can produce the signature on a new message m under the identity ID (i.e., $ID \notin \{ID_{i_1}, \cdots, ID_{i_{k_1}}\}$, $(ID, m) \notin \{(ID_{j_1}, m_{j_1}), \cdots, (ID_{j_{k_2}}, m_{j_{k_2}})\}$) with non-negligible probability.*

Definition 2.3. *An ID-based threshold signature scheme $\mathcal{TS}$=(`Setup`, `Thresh-Extract`, `Thresh-Sign`, `Verify`) is (t, n) robust if in a group of n players, even in the presence of an adversary who corrupts at most $t-1$ players, both `Thresh-Extract` and `Thresh-Sign` can complete successfully.*

3 The Schnorr's Signature Scheme.

The Schonrr's signature scheme [Sch91] is a signature scheme based on the discrete logarithm problem, which can be proved secure against the adaptively chosen message attack in the random oracle model. It consists of the following three protocols: `Setup`, `Sign`, `Verify`.

Setup. It takes as input a security parameter 1^k and outputs a public key $(p, q, g, H(\cdot), y)$ and a secret key x, where p and q are two large primes, $q|p-1$, g is a generator of order q in Z_p, $H(\cdot)$ is a cryptographic hash function: $\{0,1\}^* \rightarrow Z_q^*$, and $y = g^x \bmod p$.

Sign. To sign a message m, the user does the following performances. (1) choose a random $r \in Z_q^*$, (2) compute $R = g^r \bmod p$, and (3) set the signature to be (R, σ), where $\sigma = r + xH(m||R) \bmod q$.

Verify. To verify a signature (R, σ) for message m, the verifier does: check $g^\sigma \stackrel{?}{=} Ry^{H(m||R)} \bmod p$. If it holds, the signature is valid; otherwise, the signature is invalid.

4 Basic Tools

In this section we recall Feldman's verifiable secret sharing protocol [Fel87], denoted here by `Protocol VSS`, and the secure distributed key generation protocol [GJKR99], denoted here by `Protocol DKG`.[1]

We say an issue is a *complaint* if the share of a player cannot pass the verified equation, and the player asks the dealer to reveal his share. We say a dealer is *disqualified* if more than t players broadcast complaints against the dealer.

A high-level descriptions of the `Protocol VSS` and the `Protocol DKG` are given in Figure 1 and Figure 2, respectively. p, q are two large primes, and they satisfy $q|p-1$, and g , h are two random generators of Z_p of order q. For details, the reader can refer to [Fel87,GJKR99].

5 New ID-based Signature Scheme

In this section, we propose our ID-based signature scheme without pairings, which is based on the Schnorr's signature scheme. Follow the definition of ID-based signature schemes, it consists of the following four protocols.

Setup. Given security parameter $k_1, k_2 \in Z^+$, the protocol works as follows:

Step 1: Choose a k_1-bit prime p and a k_2-bit prime q, such that $q|p-1$.

Step 2: Choose a generator g of order q in Z_p.

Step 3: Choose a random $x \in Z_q^*$, and compute $y = g^x \bmod p$.

Step 4: Choose two cryptographic hash functions $H_1(\cdot)$ and $H_2(\cdot)$, such that $H_1 : \{0,1\}^* \to Z_q^*$, $H_2 : \{0,1\}^* \to Z_q^*$.

The public key of PKG is $(p, q, g, y, H_1(\cdot), H_2(\cdot))$, and the corresponding master key is x.

Extract. It is an interactive protocol between the user and the PKG.

1. The user first chooses a random $r_{ID} \in Z_q^*$, and computes $R_{ID} = g^{r_{ID}} \bmod p$. At last, the user sends (ID, R_{ID}) to the PKG.
2. Upon receiving (ID, R_{ID}), the PKG does: (1) choose a random $r_{PKG} \in Z_q^*$, (2) compute $R_{PKG} = g^{r_{PKG}} \bmod p$, and (3) set the private key $sk_{ID} \leftarrow (R_{PKG}, d_{ID})$, where $d_{ID} = r_{PKG} + xH_1(ID||R_{ID}||R_{PKG}) \bmod q$. At last, the PKG sends the private key to the user.

[1] The protocols in this paper are slightly different from the original ones, however, they have the same properties as the original ones.

Protocol VSS

participants: a dealer, n players $P_i (i = 1, \cdots, n)$.
input: r, R, p, q, g, such that $R = g^r \bmod p$.

1. The dealer performs as follows.
   (a) Chooses a random polynomial $f(x)$ over Z_q of degree $t-1$:

$$f(x) = \sum_{i=0}^{t-1} a_i x^i,$$

   such that $f(0) = a_0 = r$.
   (b) Broadcast $A_i = g^{f(i)} \bmod p$ for $i = 0, \cdots, n$. Notice that $A_0 = R \bmod p$.
   (c) Compute the shares $r_i = f(i) \bmod q$ for $i = 1, \cdots, n$ and sends r_i to player P_i by a secret channel.
2. Upon receiving r_i from the dealer, each player P_i does the following performances.
   (a) Choose randomly t A_{k_j}'s from $A_k (k = 0, \cdots, i-1, i+1, \cdots, n)$.
   (b) Check

$$g^{r_i} \stackrel{?}{=} A_i \stackrel{?}{=} \prod_{j=0}^{t-1} A_{k_j}^{\lambda_{i,k_j}} \bmod p, \quad (1)$$

   where λ_{i,k_j}'s are the Lagrange interpolation coefficients. If it does not hold, player P_i broadcasts a *complaint* against the dealer.

If more than t players complain then the dealer is clearly bad and he is disqualified. Otherwise, the dealer distributes the value of r among these n players.

Fig. 1. Secure verifiable secret sharing.

Protocol DKG

participants: n players $P_i (i = 1, \cdots, n)$.
input: p, q, g, h.

1. Each player P_i does the following performances.
   (a) Choose two random number r_i and r'_i over Z_q.
   (b) Choose two random polynomials $f_i(x)$, $f'_i(x)$ over Z_q of degree $t-1$:

$$f_i(x) = \sum_{j=0}^{t-1} a_{ij} x^j, \qquad f'_i(x) = \sum_{j=0}^{t-1} b_{ij} x^j,$$

   such that $f_i(0) = a_{i0} = r_i$, $f'_i(0) = b_{i0}$.
   (c) Broadcast $A_{ij} = g^{a_{ij}} h^{b_{ij}} \bmod p$ for $j = 0, \cdots, t-1$.
   (d) Compute the shares $x_{ij} = f_{ij} \bmod p$ and $x'_{ij} = f'_{ij} \bmod p$ for $j = 0, \cdots, t-1$, and send x_{ij}, x'_{ij} to player P_j.
   (e) On receiving x_{ji}, x'_{ji} from player P_j, player P_i checks

$$g^{x_{ji}} h^{x'_{ji}} \stackrel{?}{=} \prod_{k=0}^{t-1} (A_{jk})^{i^k} \bmod p.$$

   If it does not hold, player P_i broadcasts a *complaint* against player P_j.
   (f) Build the set of non-disqualified players `QUAL`.
   (g) If player P_i is in `QUAL`, he broadcasts $B_{ij} = g^{f_i(j)} \bmod p$ for $j = 0, \cdots, n$.
   (h) For each $j \in$ `QUAL`, P_i first chooses t B_{jk_l}'s ($k_l \neq i$, and $l = 0, \ldots, t-1$), and then checks

$$g^{x_{ji}} \stackrel{?}{=} B_{ji} \stackrel{?}{=} \prod_{l=0}^{t-1} (B_{jk_l})^{\lambda_{i,k_l}} \bmod p, \tag{2}$$

   and

$$B_{j0} \stackrel{?}{=} \prod_{l=0}^{t-1} (B_{jk_l})^{\lambda_{0,k_l}} \bmod p, \tag{3}$$

   where λ_{i,k_l}'s are the Lagrange interpolation coefficients. If one of them does not hold, player P_i broadcasts a *complaint* against player P_j.
   (i) Rebuild the set of non-disqualified players `QUAL`.
2. The distributed secret value x is not explicitly computed by any party, but it equals $x = \sum_{i \in \texttt{QUAL}} r_i \bmod q$. Each player P_i sets his share of the secret as $x_i = \sum_{j \in \texttt{QUAL}} x_{ji} \bmod q$. The corresponding public key of this group is $y = \prod_{i \in \texttt{QUAL}} B_{i0} \bmod p$.

Fig. 2. Secure distributed key generation.

3. Upon receiving the private key, the user checks

$$g^{d_{ID}} \stackrel{?}{=} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})} \bmod p. \quad (4)$$

If it does not hold, the user broadcasts a *complaint* against the PKG.

4. The private key of the user is $sk_{ID} = r_{ID} + d_{ID} \bmod q$, such that

$$g^{sk_{ID}} = R_{ID} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})} \bmod p \quad (5)$$

Sign. To sign a message m under the public key ID, the protocol does: (1) choose a random $r \in Z_q^*$, (2) compute $R = g^r \bmod p$ and $\beta = H_2(ID||R_{ID}||R_{PKG}||R||m)$, and (3) set the signature to be $(R_{ID}, R_{PKG}, R, \sigma)$, where $\sigma = r + (r_{ID} + d_{ID})\beta \bmod q$.

Verify. To verify a signature $(R_{ID}, R_{PKG}, R, \sigma)$ for message m, the protocol does: check $g^{\sigma} \stackrel{?}{=} R(R_{ID} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})})^{\beta} \bmod p$.

5.1 Security Analysis of Our ID-based Signature Scheme

To prove the security of our ID-based signature scheme, we make use of the techniques in [PS00], especially the Forking Lemma, which is described as follows. For details, the reader can refer to [PS00].

Lemma 5.1 (The Forking Lemma ([PS00], Theorem 13)). *Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. And $\mathcal{A}$ can ask to the signer with q_s queries, and can ask to the random oracle with q_h queries. Suppose that, $\mathcal{A}$ can produce a valid signature $(m, \sigma_1, h, \sigma_2)$, with probability $\epsilon \geq 10(q_s+1)(q_s+q_h)/2^k$, in time T. If the triple (σ_1, h, σ_2) can be simulated without knowing the private key, with an indistinguishable distribution probability, then there is an another machine which has control over $\mathcal{A}$ and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$, within time $T' \leq 120686 q_h T/\epsilon$.*

Theorem 5.1. *The proposed ID-based signature scheme achieves Girault's trusted level 3.*

Proof. We suppose the PKG wants to impersonate an honest user whose identity information is ID. He can do the following performances:

1. choose a random $r'_{ID} \in Z_q^*$ and compute $R'_{ID} = g^{r'_{ID}} \bmod p$.
2. Perform the `Extract` protocol and `Sign` protocol of the proposed ID-based signature scheme for the message m.
3. Output $(R'_{ID}, R_{PKG}, R', \sigma')$

The signature can be easily verified valid by the `Verify` protocol of the proposed ID-based signature scheme. However, the user can provide a proof to convince that the signature is forged by the PKG. The user first sends R_{ID} and R_{PKG} to the arbiter, and then provides a "knowledge proof" that he knows

$d_{ID} = r_{PKG} + xH_1(ID||R_{ID}||R_{PKG}) \bmod q$: the arbiter randomly chooses a message m' and sends it to the user; the user then perform the Schnorr's signature scheme (i.e., the secret singing key is d_{ID}, and the corresponding public key is $R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})}$). If the result signature can be pass the `Verify` protocol of the Schnorr's signature scheme, the arbiter deduces that the PKG is dishonest because the value d_{ID} is computed by the PKG only (the used signature is the Schnoor's signature scheme, the secret signing key is the PKG's master key, the corresponding public key is the PKG's public key, and the message is $(ID||R_{ID})$.).

As a result, our ID-based signature scheme achieves Girault's trusted level 3. □

Theorem 5.2. *In the random oracle model, our ID-based signature scheme is existentially unforgeable against adaptively chosen message and* ID *attack under the assumption that the Schnorr's signature scheme is secure against the adaptively chosen message attack in the random oracle model.*

Proof. From the simulations of $\texttt{hash}_1$ oracle (see Fig. 3), `Extract` oracle (see Fig. 4), $\texttt{hash}_2$ oracle (see Fig. 5), and `Sign` oracle (see Fig. 6), we can see that the view of these simulations is indistinguishable from a view of an actual random execution of the proposed signature scheme. In other words, the signer can be simulated without knowing the master key x, with an indistinguishable distribution.

Note that, in our scheme, since r is randomly chosen from Z_q^*, hence R is a random number. Furthermore, β is the hash value of $ID||R_{ID}||R_{PKG}||R||m$, and for the same (ID, R_{ID}, R_{PKG}), σ only depends on $ID||R_{ID}||R_{PKG}||R||m$, and β.

Now, we can apply the Forking Lemma, and get two valid signatures $(m, ID, R_{ID}, R_{PKG}, R, \sigma)$ with β and $(m, ID, R_{ID}, R_{PKG}, R, \sigma')$ with β. Then we have

$$\begin{aligned} g^{\sigma}(R_{ID}R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})})^{-\beta} &= g^{\sigma'}(R_{ID}R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})})^{-\beta'} \Rightarrow \\ g^{\sigma-\sigma'} &= (R_{ID}R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})})^{\beta-\beta'} \Rightarrow \\ g^{\frac{\sigma-\sigma'}{\beta-\beta'}} &= R_{ID}R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})} \end{aligned}$$

That is, the user's private key is $\frac{\sigma-\sigma'}{\beta-\beta'}$. As a result, we get a valid forgery on message $m(= ID)$ for the Schnorr's signature scheme, in which the secret signing key and the public key are the master key and the public key of the PKG, respectively. The valid forgery is $(R_{ID}R_{PKG}, \frac{\sigma-\sigma'}{\beta-\beta'})$. Though there is a slight difference between this forgery and the regular signature of the Schnorr's signature scheme ($H_1(m||R_{ID}||R_{PKG})$ VS. $H_1(m||R_{ID}R_{PKG})$), it can be considered as a difference on the construction of hash function which is the same in the random oracle model.

As a result, we finish our proof. □

— For a $\texttt{hash}_1$-query $H_1(ID_i, R_1, R_2)$, such that a record $(ID_i, R_1, R_2, \alpha, \beta)$ appears in list $\mathcal{H}_1$, the answer is β. Otherwise the answer β is defined according to the following rule:

1. Choose a random element $\beta \in Z_q^*$.

The record $(ID_i, R_1, R_2, \perp, \beta)$ is added to $\mathcal{H}_1$.

Fig. 3. Simulation of $\texttt{hash}_1$ oracle

— For an `Extract`-query with (ID_i, R_{ID}), such that a record $(ID_i, R_{ID}, R_2, \alpha, \beta)$ appears in list $\mathcal{H}_1$, the answer is (R_2, α)[a]. Otherwise the answer (R_2, α) is defined according to the following rules:

1. Choose two random elements $\alpha, \beta \in Z_q^*$,
2. Compute R_2, such that $g^{\alpha} = R_2 y^{\beta} \bmod p$.

The record $(ID_i, R_{ID}, R_2, \alpha, \beta)$ is added to $\mathcal{H}_1$.

[a] There is a situation that α is $\perp$, however, such a situation is very rare.

Fig. 4. Simulation of `Extract` oracle

— For a $\texttt{hash}_2$-query with (ID_i, R_1, R_2, R_3, m), such that a record $(ID_i, R_1, R_2, R_3, m, \alpha, \beta)$ appears in list $\mathcal{H}_2$, the answer is β. Otherwise the answer β is defined according to the following rule:

1. Choose a random element $\beta \in Z_q^*$.

The record $(ID_i, R_1, R_2, R_3, m, \perp, \beta)$ is added to $\mathcal{H}_2$.

Fig. 5. Simulation of $\texttt{hash}_2$ oracle

— For a `Sign`-query with (m, ID), we produce the answer $(R_{ID}, R_{PKG}, \sigma)$ is computed as following rules:

Step 1 Choose a random number r_{ID} and compute $R_{ID} = g^{r_{ID}} \bmod p$.
Step 2 Issue the `Extract`-query with (ID_i, R_{ID}) ourself, and get (R_2, α).
Step 3 Compute σ by using (r_{ID}, α) in the `Sign` protocol.

Fig. 6. Simulation of the `Sign` oracle

6 ID-based threshold Signature Scheme

In this section, we extend our ID-based signature scheme to two ID-based threshold signature scheme: Scheme 1 and Scheme 2. Furthermore, we give their security proof in this section.

6.1 Scheme 1

In this subsection, we propose our first ID-based threshold signature scheme, in which there exist a PKG, a group manager and n players, where the group manager knows the private key of this group, and distributes the shares of the private key to n players.

Setup. Identical to that in our ID-based signature scheme.

Thre-Extract. It is an interactive protocol.

1. The group manager first chooses a random $r_{ID} \in Z_q^*$, and computes $R_{ID} = g^{r_{ID}} \bmod p$. At last, the user sends (ID, R_{ID}) to the PKG.
2. Upon receiving (ID, R_{ID}), the PKG does:(1) choose a random $r_{PKG} \in Z_q^*$, (2) compute $R_{PKG} = g^{r_{PKG}} \bmod p$, and (3) set the private key $sk_{ID} \leftarrow (R_{PKG}, d_{ID})$, where $d_{ID} = r_{PKG} + xH_1(ID||R_{ID}||R_{PKG}) \bmod q$. At last, the PKG sends the private key to the group manager.
3. Upon receiving the private key, the group manager checks

$$g^{d_{ID}} \stackrel{?}{=} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})} \bmod p.$$

   If it does hold, the user broadcasts a *complaint* against the PKG.
4. The group manager and n players perform the `Protocol VSS`, the group manager is the dealer, and $(r_{ID}+d_{ID}, R_{ID}R_{PKG}y^{H_1(ID||R_{ID}||R_{PKG})}, p, q, g)$ is the input of the `Protocol VSS`.
5. As a result, $r_{ID} + d_{ID}$ is shared among n players. Let x_i be the share of P_i, and $f(x) = r_{ID} + d_{ID} + \sum_{i=1}^{t-1} a_i x^i \bmod q$ be the used secret-sharing polynomial, $A_i = g^{f(i)} \bmod p$, $(i = 0, \cdots, n)$.

Thre-Sign. Let Φ be a subset of the *non-disqualified* players, and $\lambda_i^{\Phi} = \prod_{k \in \Phi, k \neq i} \frac{0-k}{i-k}$, where $|\Phi| \geq t$. To sign a message m, player P_i in Φ does the following performances.

1. Choose a random r_{P_i}.
2. Compute and broadcast $R_{P_i} = g^{r_{P_i}} \bmod p$.
3. Compute $R_p = \prod_{i \in \Phi} R_{P_i}^{\lambda_i^{\Phi}} \bmod p$.
4. Compute and broadcast $\sigma_{P_i} = r_{P_i} + x_i H_2(R_{ID}||R_{PKG}||R_p||m) \bmod q$.
5. Check

$$g^{\sigma_{P_i}} \stackrel{?}{=} R_{P_i}(A_i)^{H_2(R_{ID}||R_{PKG}||R_p||m)} \bmod p. \quad (6)$$

   If it does not hold, player P_i broadcasts a *complaint* against player P_j.

If all the players in Φ are honest, the signature is $(R_{ID}, R_{PKG}, R_p, \sigma)$, where $\sigma = \sum_{i \in \Phi} \lambda_i^{\Phi} \sigma_{P_i} \bmod q$.

Verify. Identical to that in our ID-based signature scheme.

6.2 Analysis of Scheme 1

Unforgeability. Following the tradition of the security proof for threshold signature schemes, we use the concept of *simulatable adversary view* [GJKR96] to prove unforgeability. The security of a simulatable ID-based threshold signature scheme equals to the security of its underlying ID-based threshold signature scheme.

Definition 6.1. *An ID-based threshold signature scheme $\mathcal{TS}$=(`Setup`, `Thresh-Extract`, `Thresh-Sign`, `Verify`) is simulatable if the following properties hold:*

1. *The protocol `Thresh-Extract` is simulatable. That is, there exists a simulator SIM_1 that, on input the public input , the public output, and the shares of $t-1$ corrupted players of an execution of `Thresh-Extract`, can simulate the view of the adversary on that execution.*
2. *The protocol `Thresh-Sign` is simulatable. That is , there exists a simulator SIM_2 that, the public input, the public output, and the shares of $t-1$ corrupted players of an execution of `Thresh-Sign`, can simulate the view of the adversary on that execution.*

Lemma 6.1. *The scheme 1 is simulatable.*

Proof. A high-level descriptions of the simulators of `Thresh-Extract` and `Thresh-Sign` are given in Fig. 7 and Fig. 8, respectively.

For a polynomial $f(x)$ of degree $t-1$, $f(i)$ can be computed from other t or more $f(j)$'s (we denote this set as Φ), by using $f(i) = \sum_{j\in\Phi} \lambda_{i,j}^{\Phi} f(j)$, where $\lambda_{i,j}^{\Phi}$'s are the Lagrange interpolation coefficients (i.e., $\lambda_{i,j}^{\Phi} = \prod_{k\in\Phi,k\neq j} \frac{i-k}{j-k}$). On the other hand, if we know t or more $g^{f(j)} \bmod p$'s, we can compute $g^{f(i)}$ by using $g^{f(i)} = \prod_{j\in\Phi}(g^{f(j)})^{\lambda_{i,j}^{\Phi}} \bmod p$.

Following these above methods, we construct our simulators. In these two simulators, we assume w.l.o.g. that the adversary corrupted the first $t-1$ players $P_1, \cdots, P_{t-1}$. From the viewpoint of the adversary, the output of these two simulators is indistinguishable from the real execution of `Thresh-Extract` and `Thresh-Sign`. Then we finish this proof. □

Combining Theorem 5.2 and Lemma 6.1, we have the following theorem.

Theorem 6.1. *Our scheme 1 is unforgeable, if the Schnorr's signature scheme is unforgeable.*

Robustness. The following theorem can be easily proven by inspection of the scheme 1.

Theorem 6.2. *Scheme 2 is (t, n) robust, if only $n \geq 2t-1$.*

SIM_1 for scheme 1

input: y, R_{ID}, R_{PKG}, ID, $t-1$ shares x_i's of the corrupted player

1. The SIM_1 computes

$$A_i = (R_{ID}R_{PKG}y^{H_1(ID||R_ID||R_{PKG})})^{\lambda_{i,0}} \prod_{j=1}^{t-1}(g^{x_j})^{\lambda_{i,j}} \bmod p,$$

   for $i = t, \cdots, n$. Notice that the $t-1$ corrupted players can pass the Equation 1.
2. Follow the Step 2 of `Protocol VSS`.

Fig. 7. Simulator for `Thresh-Extract` of scheme 1.

SIM_2 for scheme 1

input: y, R_{ID}, R_{PKG}, ID, $t-1$ shares x_i's of the corrupted player

1. After the $t-1$ corrupted players broadcast $R_{P_i}(i = 1, \cdots, t-1)$, the SIM_2 computes and broadcasts

$$R_{P_i} = R_p^{\lambda_{i,0}} \prod_{j=1}^{t-1} R_{P_j}^{\lambda_{i,j}} \bmod p,$$

   for $i = t, \cdots, n$.
2. The SIM_2 set $f(x)$ as a polynomial of degree $t-1$, such that $f(0) = \sigma$, $f(i) = \sigma_i (i = 1, \cdots, t-1)$. And then SIM_2 computes $\sigma_i = f(i) \bmod q$ for $i = t, \cdots, n$. Notice that σ_i can pass the Equation 6.

Fig. 8. Simulator for `Thresh-Sign` of scheme 1.

6.3 Scheme 2

In this subsection, we propose our second ID-based threshold signature scheme. In this scheme, there still exist a PKG, and n players. However, there is no one knows the private key of these n players.

Setup. Identical to that in our ID-based signature scheme.

Thre-Extract. It is an interactive protocol.

1. The n players perform the `Protocol DKG`. (p, q, g, h) is the input of the `Protocol DKG`, where h is a random number in Z_p, and no one knows the discrete logarithm of h.[2] Let the final shares of player P_i be (x_i, x_i'), and $f_1(x) = \sum_{i=0}^{t-1} a_i x^i \bmod q$ and $f_2(x) = \sum_{i=0}^{t-1} b_i x^i \bmod q$ are the final secret-sharing polynomials, such that $f_1(i) = x_i$ and $f_2(i) = x_i'$, $A_i = g^{f_1(i)} \bmod p$, $(i = 0, \cdots, n)$. At last, set $R_{ID} = A_0$.
2. Upon receiving the tuple (ID, R_{ID}), the PKG does:
   (a) Choose a random $r_{PKG} \in Z_q^*$.
   (b) Compute and broadcast $R_{PKG} = g^{r_{PKG}} \bmod p$.
   (c) Set the private key $sk_{ID} \leftarrow (R_{PKG}, d_{ID})$, where $d_{ID} = r_{PKG} + xH_1(ID||R_{ID}||R_{PKG}) \bmod q$.
   (d) The PKG and n players perform `Protocol VSS`. The input of `Protocol VSS` is $(p, q, g, d_{ID}, R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})})$. As a result, the PKG distributes d_{ID} among n players. Let d_{P_i} be the share of player P_i, $f_3(x) = d_{ID} + \sum_{i=1}^{t-1} c_i x^i \bmod q$ be the secret-sharing polynomial, such that $f_3(i) = d_{P_i}$, $B_i = g^{f_3(i)} \bmod p (i = 0, \cdots, n)$. At last, set $R_{PKG} = B_0$.

Thre-Sign. Let Φ be a subset of the *non-disqualified* players, and $\lambda_i^{\Phi} = \prod_{k \in \Phi, k \neq i} \frac{0-k}{i-k}$, where $|\Phi| \geq t$. To sign a message m, player P_i in Φ does the following performances.

1. Choose a random r_{P_i}.
2. Compute and broadcast $R_{P_i} = g^{r_{P_i}} \bmod p$.
3. Compute $R_p = \prod_{i \in \Phi} R_{P_i}^{\lambda_i^{\Phi}} \bmod p$.
4. Compute and broadcast $\sigma_{P_i} = r_{P_i} + (x_i + d_{P_i}) H_2(R_{ID}||R_{PKG}||R_p||m)$.
5. Check
$$g^{\sigma_{P_i}} \stackrel{?}{=} R_{P_i} (A_j \cdot B_j)^{H_2(R_{ID}||R_{PKG}||R_p||m)} \bmod p. \quad (7)$$
   If it does not hold, player P_i broadcasts a *complaint* against player P_j.

If all the players in Φ are honest, the signature is $(R_{ID}, R_{PKG}, R_p, \sigma)$, where $\sigma = \sum_{i \in \Phi} \lambda_i^{\Phi} \sigma_{P_i} \bmod q$.

Verify. Identical to that in our ID-based signature scheme.

6.4 Analysis of Scheme 2

Unforgeability. Like the security proof of scheme 1, we also use the simulator to prove scheme 2 is unforgeable.

[2] h can be computed by $H(g)$, where $H(\cdot)$ is a cryptographic hash function, $H(\cdot) : Z_p \rightarrow Z_p$.

Lemma 6.2. *Scheme 2 is simulatable.*

Proof. Due to the simulator for `Thresh-Sign` of scheme 2 is the same as that in scheme 1, we only give the simulator of `Thresh-Extract` (see Fig. 9). As mentioned in the proof of Lemma 6.1, the output of the simulator is indistinguishable from that in the real execution of `Thresh-Sign` of scheme 2, from the adversary's viewpoint.

As a result, we finish the proof of this lemma. □

SIM_1 for scheme 2

input: y, R_{ID}, R_{PKG}, ID, $t-1$ shares (x_i, d_{P_i})'s of the corrupted players.

1. The SIM_1 runs Steps 1(a), 1(b), 1(c), 1(e), 1(f) of `Protocol DKG` on the behalf of the uncorrupted players. Let $f_i(x) = \sum_{j=0}^{t-1} a_{ij}x^j (i = 1, \cdots, n)$ be the secret-sharing polynomial dealt by player P_i. Denote by x_i the final share of player P_i.
2. The SIM_1 computes $B_{ij} = g^{f_i(j)} \bmod p$ for $i = 1, \cdots, n$ and $j = 1, \cdots, n$. For player P_n doe the following: Set $B_{n0} = R_{ID} \cdot \prod_{i=1}^{n-1}(B_{i0})^{-1} \bmod p$. Compute $B_{ni} = B_{n0}^{\lambda_{i,0}} \prod_{i=j}^{t-1}(g^{x_j})^{\lambda_{i,j}} \bmod p$, where $\lambda_{i,j}$'s are known coefficients. For each player P_i, $(i = t, \cdots, n)$ broadcasts B_{ij} for $j = 1, \cdots, n$.
3. The SIM_1 runs Steps 1(h), 1(i), 2 of `Protocol DKG`.
4. The SIM_1 runs the SIM_1 for scheme 1 on the input: $(y, R_{ID}, R_{PKG}, ID, t-1$ shares d_{P_i}'s of the corrupted players).

Fig. 9. Simulator for `Thresh-Extract` of scheme 2.

Combining Theorem 5.2 and Lemma 6.2, we have the following theorem.

Theorem 6.3. *Our scheme 2 is unforgeable, if the Schnorr's signature scheme is unforgeable.*

Robustness. The following theorem can be easily proven by inspection of the scheme 2.

Theorem 6.4. *Scheme 2 is (t, n) robust, if only $n \geq 2t - 1$.*

7 Performance Evaluation

In this section, in terms of computational complexity, we show that the efficiency of our ID-based threshold signature schemes is high. The performance evaluation notations are defined as follows:

T_{exp}	time for a modular exponetiation computation
T_{mul}	time for a multiplication computation
T_e	time for a bilinear pairing

Like Chen *et al.*'s scheme, our scheme 1 also has a group manager, while our scheme 2 does not require such a group manager. Furthermore, Chen *et al.*'s scheme is considered as the best scheme among the existing schemes. Hence, we only give the comparison between Chen *et al.*'s scheme and our scheme 1, which is shown in Table 1.

From Table 1, our scheme 1 are more efficient than Chen *et al.*'s scheme, especially in `Thresh-Sign`.

Table 1. The comparison of computational complexity[a]

		Chen *et al.*'s scheme	Our scheme 1
`Setup`	the group manager	$1T_{exp}$	$1T_{exp}$
	the PKG	$1T_{exp}$	$1T_{exp} + 1T_{mul}$
`Thresh-Extract`	the group manager	$nT_{exp} + nT_e$	nT_{exp}
`Thresh-Sign`	Generating a partial signature	$(2+3t)T_{exp} + (3t-2)T_{mul} + 4T_e$	$(t+1)T_{exp} + tT_{mul}$
	Combining partial signatures	$2tT_{exp} + 2(t-1)T_{mul}$	$tT_{exp} + (t-1)T_{mul}$
`Verify`		$2T_{exp} + 2T_{mul} + 6T_e$	$3T_{exp} + 3T_{mul}$

[a] We omit the computational complexity of the data's verification.

8 Conclusion

In this paper, we propose a new ID-based signature scheme without pairings, in which there is on trusted PKG. And then we extend it to be two ID-based threshold signature schemes. In both of these two schemes, there is only one PKG who is not assumed to be trusted. Furthermore, the second scheme does not require trusting any particular party at any time. Our schemes are more efficient than the existing schemes in terms of computation complexity.

References

[BF01] D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. *In Crypto '01*, LNCS 2139, Springer-Verlag, pp. 231-229, 2001.

[BF03] D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.

[BZ04a] J. Baek and Y. Zheng. Identity-Based Threshold Decryption. In *PKC 2004*, LNCS 2947, Springer-Verlag, pp. 262-276, 2004.

[BZ04b] J. Baek and Y. Zheng. Identity-Based Threshold Signature Scheme from the Bilinear Pairings. In *Proceeding of the international Conference on Information and Technology: Coding and Computing (ITCC'04)*, pp. 124-128, 2004.

[CC03] J. Cha and J. H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. *In PKC '03*, LNCS 2567, pp. 18-30, Springer-Verlag, 2003.

[Coc01] C. Cocks. An identiy based encryption scheme based on quadratic residues. *In Eighth IMA International Conference on Cryptography and Coding*, Dec. 2001, Royal Agricultural College, Cirencester, UK.

[CZKK04] X. Chen, F. Zhang, D. M. Konidala, and K. Kim. New ID-Based Threshold Signature Scheme from Bilinear Pairings. In *INDOCRYPT 2004*, LNCS 3348, Springer-Verlag, pp. 371-383, 2004.

[Fel87] P. Feldman. A practical Scheme for Non-interactive Verifiable Secret Sharing. In *Proc. 28th FOCS*, pp. 427-437, 1987.

[Gir91] M. Girault. Self-certified public kyes. In *EUROCRYPT 1991*, LNCS 547, pp. 490-497, 1991.

[GJKR96] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust Threshold DSS Signatures. In *EUROCRYPT 1996*, pp. 354-371, 1996.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. The (in) security of distributed key generation in dlog-based cryptosystems. In *EUROCRYPT 1999*, LNCS 1592, pp. 295-310, 1999.

[Hes02] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. *In SAC '02*, LNCS 2595, pp. 310-324, Springer-Verlag, 2003.

[Pat02] K. G. Paterson. ID-based signatures from pairings on elliptic curves. Avaiable from http://eprint.iacr.org/2002/004.

[Ped91] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Crypt '91*, LNCS 576, pp. 129-140, 1991.

[PS00] D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, (2000) 13, pp. 361-396, 2000.

[Sch91] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, Vol. 4, No. 3, pp. 161-174, 1991.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. *In Crypto '84*, LNCS 196, Springer-Verlag, pp. 47-53, 1984.

[SOK00] R. Sakai, K. Ohgishi and M.Kasahara. Cryptosystems based on pairing. In *SCIS '00*, pp.26-28, 2000.