Efficient identity-based threshold signature scheme from bilinear pairings in the standard model ☆

Wei Gao[a,b], Guilin Wang[c], Xueli Wang[d], Kefei Chen[b]

[a] *School of Mathematics and Information, Ludong University, Yantai 264025, P.R.China*
[b] *Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, P.R.China*
[c] *School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK*
[d] *School of Mathematics, South China Normal University, Guangzhou 510631, P.R.China*

Abstract

We propose a new identity-based threshold signature (IBTHS) scheme from bilinear pairings enjoying the following advantages in efficiency, security and functionality. The round-complexity of the threshold signing protocol is optimal since each party pays no other communication cost except broadcasting one single message. The computational complexity of the threshold signing procedure is considerably low since there appears no other time-consuming pairing except two pairings for verifying each signature shares. The communication channel requirement of the threshold signing procedure is the lowest since the broadcast channel among signers is enough. It is proved secure with optimal resilience in the standard model. It is the private key associated with an identity rather than a master key of the Public Key Generator (PKG) that is shared among signature generation servers. All these excellent properties are due to our new basic technique by which the private key in the bilinear group is indirectly shared through simply sharing an element in the finite field.

Keywords:
Provable security, Identity-based signature, Threshold signature, Standard model, Bilinear pairing

☆This work is partially supported by National Natural Science Foundation of China (No. 10771078, No. 60973135) and the Ludong University Research Program (No. LY20072701).

Email addresses: sdgaowei@gmail.com (Wei Gao), G.Wang@cs.bham.ac.uk (Guilin Wang), wangxuyuyan@gmail.com (Xueli Wang), kfchen@sjtu.edu.cn (Kefei Chen)

1. Introduction

In 1987, Desmedt introduced the concept of threshold signatures [1]. In a (t, n) threshold signature scheme, a secret key (and equivalently, the signing power) is distributed to a group of n players in a way that any subset of t players can cooperatively produce a signature on behalf of the group, while up to $t - 1$ players cannot. A threshold signature scheme is a very important cryptographic primitive because of its twofold application: to increase the availability of the signing agency and at the same time to increase the protection against forgery by making it harder for the adversary to learn the secret signature key. Since Desmedt's work, in the so-called threshold cryptography field, many threshold signature schemes based on different assumptions, such as [2, 3, 4, 5, 6, 7], have been constructed.

In 1984, Shamir [8] asked for identity-based (ID-based) encryption and signature schemes to simplify key management procedures in certificate-based public key setting. Since then, in the so-called identity-based cryptography filed, many ID-based cryptographic schemes, such as the results in [9, 10], have been proposed. Bilinear pairing [9] is the most popular tool to construct identity-based cryptographic primitives. The ID-based public key setting can be an alternative for certificate-based public key setting, especially when efficient key management and moderate security are required.

In 2004, as the combination of the above two interesting concepts, the notion of identity-based threshold signature (IBTHS) was proposed by Baek and Zheng [11]. They defined the security of the IBTHS scheme and presented a concrete implementation from bilinear pairings. Different from Boneh and Franklin's threshold approach [9] that distributes the master key of the PKG into a number of other PKGs (called the Distributed PKGs) to perform threshold decryption or threshold signature generation, the important feature of the Baek-Zheng scheme is that a private key associated with an identity rather than a master key of the PKG is shared among signature generation servers. In other words, it is the holder of the ID-based private key rather than PKG that distributes the key shares. Then several IBTHS schemes from bilinear pairings have been proposed [12, 13, 14, 15]. Now we simply review these results. In [11], Baek and Zheng generalized the basic tools in finite field for threshold cryptography including Sharmir sharing, Feldman sharing, Pedersen sharing and distributed key generation protocol to the paring-based cryptography field. Depending on these basic tools, Baek and Zheng's IBTHS scheme corresponding to Hess's identity-based signature scheme is naturally obtained as in [4]. Directly applying the Baek and Zheng's methodology to various identity-based signature, a line of works [12, 13, 14] were proposed. Although the threshold signing protocols of [12, 13] avoid the distributed key generation sub-protocol which consumes most of the time for Baek and Zheng's scheme, their robustness is weaker than that of the Baek-Zheng's scheme in some sense as stated in [15, 14]. Based on the identity-based signature scheme secure without random oracles [16, 17], Sun et al. constructed a non-interactive identity-based threshold signature scheme without random oracles [14]. Since the IBTHS schemes of [11, 12, 13, 14] all heavily depend on the secret sharing tools based on pairings whose computation is very time-consuming [18], their computation efficiency is not very satisfying in practice. To solve this problem, Gao et al. [15] proposed a new identity-based signature scheme and then constructed the corresponding IBTHS scheme which enjoys the computation efficiency similar to the non-identity-based threshold signature scheme in [6].

In this paper, we propose a new identity-based threshold signature (IBTHS) scheme from bilinear pairings. It obtains the following satisfying properties in efficiency, security and functionality due to several new techniques of ours. (1) The round-complexity of the threshold signing protocol is optimal. Namely, during the signing procedure, broadcasting the signature share is the only communication cost for each player. (2) The computational complexity is optimal in the sense that only two bilinear pairings are involved for verifying each signature shares during the threshold signing procedure. (3) The communication channel is optimal. Namely, only the broadcast channel among signers is enough during the threshold signing procedure. (4) It is proved (t, n)secure (unforgeable and robust) in the standard model, for $n \geq 2t-1$, and so its resilience is optimal. (5) It is the private key associated with an identity rather than a master key of the Public Key Generator that is shared among signature generation servers. To the best of our knowledge, there is no identity-based threshold signature scheme which enjoys all the above advantages among all the schemes in [11, 12, 13, 14, 15].

The rest of the paper is organized as follows. We first review the basic property of pairings, the computational assumption on which our scheme is (indirectly) based, and the security notion of IBTHS schemes is described in Section 2. We then present our IBTHS scheme in Section 3, prove it secure in Section 4 and compare it with other IBTHS schemes in Section 5. Section

6 concludes this paper.

2. Preliminaries

2.1. Bilinear Pairing and Complexity Assumption

This section briefly reviews the definition of bilinear pairings and the relative complexity assumption.

Definition 1. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p and let g be a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be an bilinear pairing and the group $\mathbb{G}$ is called a bilinear group, if the following three conditions hold:*

(1) e is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p^*$;

(2) e is non-degenerate, i.e. $e(g, g) \neq 1$;

(3) e is efficiently computable.

The security of our signature scheme will be reduced to the hardness of the computational Diffie-Hellman (CDH) problem in the group in which the signature is constructed. We brief review the definition of the CDH problem.

Definition 2. *Given a group $\mathbb{G}$ of prime order p with generator g and elements $g^a, g^b \in \mathbb{G}$ where a, b are selected uniformly at random from $\mathbb{Z}_p^*$, the CDH problem in $\mathbb{G}$ is to compute g^{ab}.*

2.2. Security Definitions

As in [11], we review the syntax and security definitions of identity-based signature schemes and identity-based threshold signature schemes below.

Definition 3. *(identity-based signature scheme). An identity-based signatures scheme $\mathcal{IDS}$ is a tuple of the following four algorithms (***Setup***,* **Extract***,* **Sign***,* **Verify***):*

- *System initialization algorithm* **Setup** (k). *Given a security parameter k, PKG generates the system common parameters $params$ and the master secret key x. $params$ are made public, while x is kept secret.*

- *Private key extraction algorithm* **Extract** $(params, \mathfrak{u})$. *Given identity $\mathfrak{u}$ and master key x, PKG computes the private key $d_{\mathfrak{u}}$ and sends it to the corresponding entity secretly.*

- *Signing algorithm* **Sign** $(params, d_{\mathfrak{u}}, m)$. *Given the common parameter params, the private key* $d_{\mathfrak{u}}$ *associated with* $\mathfrak{u}$ *and a message* m*, this algorithm generates the signature* σ *of* $\mathfrak{u}$ *on* m.

- *Verification algorithm* **Verify** $(params, \mathfrak{u}, m, \sigma)$. *Given an identity* $\mathfrak{u}$*, a message* m *and a signature* σ*, this algorithm checks the validity of* σ *to output 1 (valid) or 0 (invalid).*

We now review the unforgeability notion for the identity-based signature scheme against chosen message attack, which we denote by UF-IDS.

Definition 4. *Let* A^{IDS} *be an attacker assumed to be a probabilistic Turing machine. Consider the following game* G^{IDS} *in which* A^{IDS} *interacts with the challenger* C^{IDS}.

Phase 1. *The challenger runs the* **Setup** *algorithm and gives* A^{IDS} *the resulting common parameter params.*

Phase 2. A^{IDS} *issues a number of private key extraction queries, each of which consists of* $\mathfrak{u}$*. On receiving* $\mathfrak{u}$*, the challenger runs the private key extraction algorithm to get the private key* $d_{\mathfrak{u}}$ *gives it to* A^{IDS}*. In addition to the key extraction queries,* A^{IDS} *issues a number of signature generation queries, each of which consists of* $(\mathfrak{u}, m)$*. The challenger first runs the private key extraction algorithm to obtain a corresponding private key and then runs the signature generation algorithm and gives a resulting signature* σ *to* A^{IDS}.

Phase 3. A^{IDS} *outputs* $(\mathfrak{u}^*, \tilde{m}, \tilde{\sigma})$*, where* $\tilde{\sigma}$ *is a valid signature of the identity* $\mathfrak{u}^*$ *on the message* $\tilde{m}$*. A restriction here is that* A^{IDS} *must not make a private key extraction query for* $\mathfrak{u}^*$ *or a signature generation query for* $(\mathfrak{u}^*, \tilde{m})$.

We denote A^{IDS}*'s success by*

$$Succ^{UF-IDS}_{IDS,A^{IDS}}(k) = Pr[\mathbf{Verify}(params, \mathfrak{u}^*, \tilde{m}, \tilde{\sigma}) = 1]$$

We denote by $Succ^{UF-IDS}_{IDS,A^{IDS}}(t, q_e, q_s)$ *the maximum of the attacker* A^{IDS}*'s success over all attackers* A^{IDS} *having running time* t *and making at most* q_e *key extraction queries and* q_s *signature generation queries. The ID-based signature scheme is said to be* (t, q_e, q_s, ϵ) *UF-IDS secure if*

$$Succ^{UF-IDS}_{IDS,A^{IDS}}(t, q_e, q_s) < \epsilon.$$

Definition 5. *(identity-based threshold signature (IBTHS) scheme). Let* $\mathcal{IDS}$ *=(***Setup***,* **Extract***,* **Sign***,* **Verify***) be an identity-based signature scheme.*

A (t, n) IBTHS scheme $\mathcal{IDTS}$ for $\mathcal{IDS}$ involves the PKG, a group of n users (signing entities) under the same identity, and the verifier, and is defined by the pair of probabilistic polynomial time algorithms (**KeyDis**, **ThrSig**)*.*

- *Key distribution algorithm* **KeyDis** $(params, d_{\mathfrak{u}}, n, t)$*. Given a private key $d_{\mathfrak{u}}$ associated with an identity $\mathfrak{u}$, this algorithm generates n shares $\{d_{\mathfrak{u},i}\}_{i=1}^{n}$ and provides them to the signing players $\{\Gamma_i\}_{i=1}^{n}$ respectively. It also generates a set of public verification keys that can be used to check the validity of each private key share and to verify partial signatures in the future.*

- *Threshold signing algorithm* **ThrSig** $(params, d_{\mathfrak{u},i}, m)$*. Assume that each server Γ_i is given the common parameter params, a share $d_{\mathfrak{u},i}$ of the private key $d_{\mathfrak{u}}$ associated with $\mathfrak{u}$ and a message m. n signature generation servers jointly generate a signature σ for the message m.*

Security requirement of IBTHS schemes includes both unforgeability and robustness [11]. We first briefly review the notion of unforgeability against chosen message and identity attack, or UF-IBTHS-CMA for short.

Definition 6. *Let A^{IDTS} be an attacker assumed to be a probabilistic Turing machine. Consider the following game G^{IDTS} in which A^{IDTS} interacts with the challenger C^{IDTS}.*

Phase 1. *The challenger runs the* **Setup** *algorithm and gives A^{IDTS} the resulting common parameter params.*

Phase 2. *A^{IDTS} corrupts $t-1$ signature generation servers(The attacker is assumed to be static).*

Phase 3. *A^{IDTS} issues a number of private key extraction queries, each of which consists of $\mathfrak{u}$. On receiving $\mathfrak{u}$, the Challenger runs the key extraction algorithm taking $\mathfrak{u}$ as input and obtains a corresponding private key $d_{\mathfrak{u}}$. The challenger gives $\mathfrak{u}$ to A^{IDTS}.*

Phase 4. *A^{IDTS} submits a target identity $\mathfrak{u}^*$. On receiving $\mathfrak{u}^*$, the challenger runs the key extraction algorithm taking $\mathfrak{u}^*$ as input and obtains a corresponding private key $d_{\mathfrak{u}^*}$. Subsequently, it runs the private key distribution algorithm taking $d_{\mathfrak{u}^*}$ as input to share it among n signature generation servers. We denote the key shares by $d_{\mathfrak{u}^*,i}$ for $i = 1, \ldots, n$. The challenger gives $d_{\mathfrak{u}^*,i}$ for $i = 1, \ldots, t-1$, (private keys for the corrupted servers) to A^{IDTS}.*

Phase 5. *A^{IDTS} issues a number of signature generation queries, each of which consists of a message denoted by m. On receiving m, the challenger,* *on behalf of the uncorrupted servers, runs the signature generation algorithm taking $d_{\mathfrak{u}^*,i}$ for $i = t, \ldots, n$ and m as input, and responds to A^{IDTS} with σ output by the signature generation algorithm . Note that in this phase, A^{IDTS} is allowed to issue private key extraction queries (identities) except for $\mathfrak{u}^*$. Note also that A^{IDTS} is allowed to see partial signature broadcast during the execution.*

Phase 6. *A^{IDTS} outputs $(\mathfrak{u}^*, \tilde{m}, \tilde{\sigma})$, where $\tilde{\sigma}$ is a valid signature of the identity $\mathfrak{u}^*$ on the message $\tilde{m}$. A restriction here is that A^{IDTS} must not make a private key extraction query for $\mathfrak{u}^*$ and it must not make a signature generation query for $\tilde{m}$.*

We denote A^{IDTS}'s success by

$$Succ^{UF-IDTHS}_{IDTHS,A^{IDTS}}(k) = Pr[\mathbf{Verify}(params, \mathfrak{u}^*, \tilde{m}, \tilde{\sigma}) = 1]$$

We denote by $Succ^{UF-IDTHS}_{IDTHS,A^{IDTS}}(t, q_e, q_s)$ the maximum of the attacker A^{IDTS}'s success over all attackers A^{IDTS} having running time t_2 and making at most q_e key extraction queries and q_s signature generation queries. The ID-based threshold signature scheme is said to be (t, q_e, q_s, ϵ) UF-IDTHS secure if

$$Succ^{UF-IDTHS}_{IDTHS,A^{IDTS}}(t, q_e, q_s) < \epsilon.$$

Definition 7. *A (t, n) ID-based threshold signature scheme is said to be robust if it computes a correct output even in the presence of a malicious attacker that makes the corrupted signature generation servers deviate from the normal execution.*

3. Construction

Let $\mathbb{G}$ be a group of prime order p for which there exists an efficiently computable bilinear map into $\mathbb{G}_T$. Additionally, let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ denote the bilinear map and g be the corresponding generator. The size of the group is determined by the security parameter. In the following all identities and messages will be assumed to be bit strings of length n_u and n_m respectively. To construct a more flexible scheme which allows identities and messages of arbitrary lengths, collision-resistant hash functions $H_u : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_m : \{0,1\}^* \to \{0,1\}^{n_m}$, can be defined and used to create identities and messages of the desired length.

The identity-based signature scheme $\mathcal{IDTS}$=(**Setup**, **Extract**, **Sign**, **Verify**) underlying our IBTHS scheme $\mathcal{IDTS}$ = (**KeyDis**, **ThrSig**) is the one due to Paterson and Schuldt [16]. All these involved algorithms are as follows.

Setup. The system parameters are generated as follows. A secret $\alpha \in \mathbb{Z}_p$ is chosen at random. We choose a random generator, $g \in \mathbb{G}$, and set the value $g_1 = g^\alpha$ and choose g_2 randomly in $\mathbb{G}$. Additionally, the authority chooses a random value $u', m' \in \mathbb{G}$ and vectors $U = (u_i), M = (m_i)$ of length n_u and n_m, respectively, whose elements are chosen at random from $\mathbb{G}$. The published public parameters are $params = (\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u',\ m', U, M)$. The master secret is g_2^α.

Extract. Let $\mathfrak{u}$ be a string of n_u bits representing an identity, $\mathfrak{u}[i]$ denote the ith bit of $\mathfrak{u}$, and $\mathcal{U} \subset \{1, \cdots, n_u\}$ be the set of all i for which $\mathfrak{u}[i] = 1$. A private key for identity $\mathfrak{u}$ is generated as follows. First, a random $r_u \in \mathbb{Z}_p$ is chosen. Then the private key $d_\mathfrak{u}$ is constructed as:

$$d_\mathfrak{u} = (g_2^\alpha (u' \prod_{i \in \mathcal{U}} u_i)^{r_u}, g^{r_u}).$$

Note that a user can easily re-randomize his private key after he has received it from the master entity.

Sign. Let $\mathfrak{u}$ be the bit string of length n_u representing a signing identity and let $\mathfrak{m}$ be a bit string representing a message. As in the **Extract** algorithm, let $\mathcal{U}$ be the set of indicies i such that $\mathfrak{u}[i] = 1$, and likewise, let $\mathcal{M} \subset \{1, \cdots, n_m\}$ be the set of indicies j such that $\mathfrak{m}[j] = 1$, where $\mathfrak{m}[j]$ is the jth bit of $\mathfrak{m}$. With the private key $d_\mathfrak{u} = (g_2^\alpha (u' \prod_{i \in \mathcal{U}} u_i)^{r_u}, g^{r_u})$, A signature of $\mathfrak{u}$ on $\mathfrak{m}$ is constructed by randomly picking $r_m \leftarrow \mathbb{Z}_p$ and computing

$$\sigma = (g_2^\alpha (u' \prod_{i \in \mathcal{U}} u_i)^{r_u} (m' \prod_{i \in \mathcal{M}} m_i)^{r_m}, g^{r_u}, g^{r_m}) \in \mathbb{G}^3$$

Verify. Given a purported signature $\sigma = (\sigma[1], \sigma[2], \sigma[3]) \in \mathbb{G}_3$ of an identity $\mathfrak{u}$ on a message $\mathfrak{m}$, a verifier accepts σ if the following equality holds:

$$e(\sigma[1], g) = e(g_2, g_1) e(u' \prod_{i \in \mathcal{U}} u_i, \sigma[2]) e(m' \prod_{i \in \mathcal{M}} m_i, \sigma[3]).$$

KeyDis. Given a private key

$$d_\mathfrak{u} = (g_2^\alpha (u' \prod_{i \in \mathcal{U}} u_i)^{r_u}, g^{r_u}),$$

the number of servers n and a threshold parameter t, this algorithm distributes $d_\mathfrak{u}$ to n servers as follows.

(1). First, it picks $a_0, a_1, \ldots, a_{t-1} \in \mathbb{Z}_p$, constructs the polynomial over $\mathbb{Z}_p$

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}$$

and sets $r'_u = a_0$.

(2). Second, it computes the public parameter Y for all n parties:

$$\begin{aligned} Y &= \left(\frac{g_2^\alpha (u' \prod_{i\in\mathcal{U}} u_i)^{r_u}}{(u' \prod_{i\in\mathcal{U}} u_i)^{r'_u}}, g^{r_u}\right) \\ &= (g_2^\alpha (u' \prod_{i\in\mathcal{U}} u_i)^{r_u - r'_u}, g^{r_u}). \end{aligned}$$

(3). Third, for each server Γ_k, it computes the shared key $d_{\mathfrak{u},k}$ and the verification key y_k:

$$\begin{aligned} d_{\mathfrak{u},k} &= f(k), \\ y_k &= e(u' \prod_{i\in\mathcal{U}} u_i, g)^{f(k)}. \end{aligned}$$

(4). Last, it secretly sends the distributed private key $d_{\mathfrak{u},k}$ to each server Γ_k, $1 \le k \le n$, and publishes $Y, y_1, y_2, \ldots, y_n$.

Remark. In the above algorithm, we share the private key in $\mathbb{G}$ by an indirect but very simple way. Roughly speaking, the above method uses the Shamir sharing scheme to simply share the secret $r'_u \in \mathbb{Z}_p$. The private key $d_\mathfrak{u} \in \mathbb{G}$ can be reconstructed immediately after $r'_u \in \mathbb{Z}_p$ is reconstructed, since

$$d_\mathfrak{u} = (Y[1] e(u' \prod_{i\in\mathcal{U}} u_i, g)^{r'_u}, Y[2]) \ .$$

In contrast, the method in [11] involves much complicated pairing computation due to direct sharing $d_\mathfrak{u} \in \mathbb{G}$ by the sharing method customized for the bilinear group. What's more, due to this initial difference in key distribution and the following full dependence on bilinear map in the threshold signing protocol, the method in [11] becomes much more consuming in terms of computation and communication than that of ours.

ThrSig. Given the message $\mathfrak{m}$, the n sharers $\{\Gamma_k\}_{k=1}^n$ generate the signature of the identity $\mathfrak{u}$ by the following protocol.

(1). With its shared key $d_{\mathfrak{u},k} = f(k)$, each sharer Γ_k randomly selects $r_k \in \mathbb{Z}_p$, computes and broadcasts the signature share

$$\sigma_k = ((u' \textstyle\prod_{i \in \mathcal{U}} u_i)^{f(k)}(m' \displaystyle\prod_{i \in \mathcal{M}} m_i)^{r_k}, g^{r_k}).$$

With $\Gamma'_k s$ verification key $y_k = e(u' \prod\limits_{i \in \mathcal{U}} u_i, g)^{f(k)}$, the validity of the signature share $\sigma_k = (\sigma_k[1], \sigma_k[2])$ due to player Γ_k can be publicly verified by checking

$$e(\sigma_k[1], g) = y_k \cdot e(m' \prod_{i \in \mathcal{M}} m_i, \sigma_k[2]).$$

(2). Each party locally reconstructs the full signature as follows. He first collects t valid signature shares using the above verification equation. Suppose that Φ is the set of indices of t honest players who generated valid signature shares. Given the signature shares $\{\sigma_k\}_{k \in \Phi}$ and the public parameter Y:

$$\begin{aligned} \{\sigma_k\}_{k \in \Phi} &= \{(\sigma_k[1], \sigma_k[2])\}_{k \in \Phi}, \\ Y &= (Y[1], Y[2]) \\ &= (g_2^{\alpha}(u' \prod_{i \in \mathcal{U}} u_i)^{r_u - r'_u}, g^{r_u}), \end{aligned}$$

the signature $\sigma = (\sigma[1], \sigma[2], \sigma[3])$ is computed as follows.

$$\begin{aligned} \sigma[1] &= Y[1] \prod_{k \in \Phi} \sigma_k[1]^{l_{\Phi,k}}, \\ \sigma[2] &= Y[2], \\ \sigma[3] &= \prod_{k \in \Phi} \sigma_k[2]^{l_{\Phi,k}}, \end{aligned}$$

where the Lagrange coefficient $l_{\Phi,k} = \prod\limits_{\substack{j \in \Phi, \\ j \neq k}} \frac{-j}{k-j}$. Since

$$\begin{aligned} \sigma[1] &= Y[1] \prod_{k \in \Phi} \sigma_k[1]^{l_{\Phi,k}} \\ &= Y[1](u' \prod_{i \in \mathcal{U}} u_i)^{\sum\limits_{k \in \Phi} f(k) l_{\Phi,k}} (m' \prod_{i \in \mathcal{M}} m_i)^{\sum\limits_{k \in \Phi} r_k l_{\Phi,k}} \\ &= g_2^{\alpha}(u' \prod_{i \in \mathcal{U}} u_i)^{r_u} (m' \prod_{i \in \mathcal{M}} m_i)^{\sum\limits_{k \in \Phi} r_k l_{\Phi,k}}, \\ \sigma[2] &= Y_2 = g^{r_u}, \\ \sigma[3] &= \prod_{k \in \Phi} \sigma_k[2]^{l_{\Phi,k}} = (m' \prod_{i \in \mathcal{U}} m_i)^{\sum\limits_{k \in \Phi} r_k l_{\Phi,k}}, \end{aligned}$$

it is obvious that $\sigma = (\sigma[1], \sigma[2], \sigma[3])$ is a valid signature. In other words, the correctness property of our scheme is satisfied.

4. Security Proof

Theorem 1. *(Robustness) The (t, n) IBTHS scheme ours is robust in the presence of up to $t-1$ malicious servers if $n \geq 2t-1$.*

Proof. First consider the **KeyDis** algorithm. It is observed that t honest players are required for later operation and at most $t-1$ players can be corrupted, which requires $n \geq 2t-1$. Furthermore, the misbehaving players cannot affect the functionality of the **KeyDis** protocol, and consequently the status of all the uncorrupted players. Therefore the **KeyDis** protocol is robust if $n \geq 2t-1$.

In the **ThrSig** phase, suppose up to $t-1$ players are corrupted and the number of honest players is at least $n-(t-1) \geq t$. Each of the corrupted players can either be halted or issue invalid partial signature. In the first case, the corrupted player does not produce any malicious data hence cannot affect the protocol execution of honest players. In the second case, if the partial signature is invalid, then by definition, it can be detected hence excluded from the final signature. Therefore, due to the correctness property, all the honest (at least t) players can still generate a valid signature in the presence of up to $t-1$ corrupted players, and our scheme is robust if $n \geq 2t-1$. The proof completes. □

To reduce the unforgeability of our IBTHS scheme to that of the underlying identity-signature scheme, for the identity-based signature scheme and the corresponding security model in Section 2, we consider a slightly modified forger model which requires that for any two signatures $\sigma_1 = (\sigma_1[1], \sigma_1[2], \sigma_1[3])$ and $\sigma_2 = (\sigma_2[1], \sigma_2[2], \sigma_2[3])$ of the same identity on different messages from the challenger, the equation $\sigma_1[2] = \sigma_2[2]$ should hold. In other words, although there are many identity-based private key corresponding to one identity in the Paterson-Schuldt scheme, the challenger is required to fix a single one and to use it to generate the signature of this identity on different messages. In this customized security model, it is obvious that the security results and the proof procedure of the identity-based signature scheme in [16] remains the same as before.

Below, we prove that if the Paterson-Schuldt signature is unforgeable, then our IBTHS scheme is also unforgeable.

Theorem 2. *The (t, n) IBTHS scheme of ours is $(t_2, q_e, q_s, \epsilon)$ UF-IDS secure, assuming the Paterson-Schuldt identity-based signature scheme is $(t_1, q_e, q_s, \epsilon)$ UF-IDTHS secure where*

$$t_1 = t_2 + q_s(t+1)(n-t+1)T_e$$

and T_e is the time for an exponentiation in $\mathbb{G}$.

Proof. Let A^{IDTS} be the attacker that defeats the UF-IDTHS security of our IBTHS scheme. In the following, we will construct an attacker A_1 that defeats the UF-IDS security of the Paterson-Schuldt signature scheme in the game G^{IDS} with his challenger C^{IDS} as in Definition 4, by simulating the challenger C^{IDTS} in the game G^{IDTS} with A^{IDTS} as in Definition 6.

Here note that as discussed above, A^{IDS} requires that for any two signatures $\sigma_1 = (\sigma_1[1], \sigma_1[2], \sigma_1[3])$ and $\sigma_2 = (\sigma_2[1], \sigma_2[2], \sigma_2[3])$ of the same identity on different messages from the challenger C^{IDS}, the equation $\sigma_1[2] = \sigma_2[2]$ should hold.

Phase 1. A^{IDS} obtains the public parameter *param* from its own challenger C^{IDS} and then passes it to the attacker A^{IDTS}.

Phase 2. A^{IDTS} corrupts $t-1$ signature generation servers $\Gamma_1, \Gamma_2, \ldots, \Gamma_{t-1}$ and controls their behavior. The other servers $\Gamma_t, \Gamma_{t+1}, \ldots, \Gamma_n$ is simulated by A^{IDS}. (That is, the attacker is assumed to be static).

Phase 3. A^{IDTS} issues a number of private key extraction queries, each of which consists of $\mathfrak{u}$. For the query $\mathfrak{u}$, A^{IDS} obtains the answer $d_{\mathfrak{u}}$ from its own challenger C^{IDS} and then passes it to A^{IDTS}.

Phase 4. A^{IDTS} submits a target identity $\mathfrak{u}^*$. First, A^{IDS} asks its challenger C^{IDS} for a signature of $\mathfrak{u}^*$ on one randomly chosen message $\mathfrak{m}^*$, for which $\mathcal{M}^* \subset \{1, \cdots, n_m\}$ be the set of indicies j such that $\mathfrak{m}^*[j] = 1$. We will see this signature as

$$\begin{aligned}\sigma^* &= (\sigma^*[1], \sigma^*[2], \sigma^*[3]) \\ &= (g_2^{\alpha}(u' \prod_{i \in \mathcal{U}^*} u_i)^{r_u}(m' \prod_{i \in \mathcal{M}^*} m_i)^{r_{m^*}}, g^{r_u}, g^{r_{m^*}}).\end{aligned}$$

Second, A^{IDS} randomly picks $R \in \mathbb{G}$ and then sets

$$\begin{aligned} Y &= (\frac{\sigma^*[1]}{R}, \sigma^*[2]), \\ y_0 &= \frac{e(R, g)}{e(m' \prod_{i \in \mathcal{M}^*} m_i, \sigma^*[3])}. \end{aligned}$$

If we see R as $(u' \prod_{i\in\mathcal{U}^*} u_i)^{r'_u}(m' \prod_{i\in\mathcal{M}^*} m_i)^{r_{m^*}}$, then we have

$$\begin{aligned}
&Y = (g_2^{\alpha}(u' \prod_{i\in\mathcal{U}^*} u_i)^{r_u - r'_u}, g^{r_u}),\\
&y_0 = \frac{e(R,g)}{e(m' \prod_{i\in\mathcal{M}^*} m_i, g^{r_{m^*}})} = e(u' \prod_{i\in\mathcal{U}} u_i, g)^{r'_u}.
\end{aligned}$$

Third, for $1 \le k \le t-1$, A^{IDS} randomly picks elements $x_k \in \mathbb{Z}_p$ as the key share of Γ_k, sets the verification key $y_k = e(u' \prod_{i\in\mathcal{U}} u_i, g)^{x_k}$, and sends x_k, y_k to Γ_k. Let $f(x) \in \mathbb{Z}_p[x]$ be the degree $(t-1)$ polynomial implicitly defined to satisfy

$$f(0) = r'_u, f(k) = x_k, \text{ for } 1 \le k \le t-1.$$

Fourth, for $t \le k \le n$, A^{IDS} computes the Lagrange coefficients $l_{0,k}, l_{1,k}, \ldots, l_{t-1,k}$ such that $f(k) = \sum_{j=0}^{t-1} l_{j,k} f(j)$, and sets $y_k = y_0^{l_{0,k}} y_1^{l_{1,k}} \ldots y_{t-1}^{l_{t-1,k}}$, where $l_{j,k} = \prod_{s\ne j, 0\le s\le t-1} \frac{k-s}{j-s}$. At last, A^{IDS} publishes $Y, y_1, y_2, \ldots, y_n$.

Phase 5. A^{IDTS} issues a number of signature generation queries, each of which consists of a message denoted by $\mathfrak{m}$. First, A^{IDS} gets the signature

$$\sigma = (g_2^{\alpha}(u' \prod_{i\in\mathcal{U}^*} u_i)^{r_u}(m' \prod_{i\in\mathcal{M}} m_i)^{r_m}, g^{r_u}, g^{r_m})$$

of the identity $\mathfrak{u}^*$ on the message $\mathfrak{m}$. Second, given

$$Y = (Y[1], Y[2]) = (g_2^{\alpha}(u' \prod_{i\in\mathcal{U}^*} u_i)^{r_u - r'_u}, g^{r_u}),$$

for $t \le k \le n$, A^{IDS} computes

$$\begin{aligned}
\sigma_k[1] &= (\frac{g_2^{\alpha}(u' \prod_{i\in\mathcal{U}^*} u_i)^{r_u}(m' \prod_{i\in\mathcal{M}} m_i)^{r_m}}{Y[1]})^{l_{0,k}} \prod_{1\le j\le t-1} (u' \prod_{i\in\mathcal{U}^*} u_i)^{l_{j,k} f(j)}\\
&= (u' \prod_{i\in\mathcal{U}^*} u_i)^{f(k)}(m' \prod_{i\in\mathcal{M}} m_i)^{r_m l_{0,k}},\\
\sigma_k[2] &= g^{r_m l_{0,k}}.
\end{aligned}$$

Then A^{IDS} broadcasts signature shares $\sigma_k = (\sigma_k[1], \sigma_k[2])$, for $t \le k \le n$.

Phase 6. At last, when A^{IDTS} outputs $(\mathfrak{u}^*, \widetilde{\mathfrak{m}}, \widetilde{\sigma})$, A^{IDS} passes them to its challenger C^{IDS}.

Since we often consider the very large q_s, the time complexity of the algorithm A^{IDS} is dominated by the exponentiations performed in **Phase 5**, in addition to the time t_1 of A^{IDTS}. There are $q_s(t+1)(n-t+1)$ exponentiations in **Phase 5**. So the time complexity of A^{IDS} is $t_2 + q_s(t+1)(n-t+1)T_e$. Additionally, it is obvious that the success probability of A^{IDTS} is the same to that of A^{IDS}. Thus, the theorem follows. □

In [16], the authors proved the security of the above scheme as the following theorem:

Theorem 3. *Theorem 1 The Paterson-Schuldt identity-based signature scheme is (t, q_e, q_s, t) unforgeable against adaptive chosen identity and message attack in the standard model, assuming that the CDH problem in $\mathbb{G}$ is (t', ϵ') intractable, where*

$$\epsilon' = \frac{\epsilon}{16(q_e+q_s)q_s(n_u+1)(n_m+1)}$$
$$t' = t + O\{[q_e n_u + q_s(n_u + n_m)]\rho + (q_e + q_s)\tau\},$$

where ρ is the time for a multiplication in $\mathbb{G}_1$ and τ for an exponentiation.

Combining Theorems 1, 2, 3, we now obtain the following theorem on the unforgeability of our IBTHS scheme:

Theorem 4. *(Unforgeability) In the standard model, our proposed IBTHS scheme is UF-IBTHS secure against an adversary who corrupts up to $t \leq (n+1)/2$ players, if the CDH problem is intractable in the underlying pairing friendly group $\mathbb{G}$.*

It can be seen from Theorem 1,4 that an adversary can corrupt up to $t-1$ of the n players in the network, for any value of $t-1 < n/2$. This is the optimal achievable threshold or resilience for solutions that provide both secrecy and robustness.

5. Comparison

Our IBTHS scheme enjoys the following desirable property in terms of functions. First, in our scheme, any user which holds the private key associated with an identity, including the identity himself and the PKG, can distribute the the private key (Property 1). Second, our scheme is more robust in the sense that a valid signature can always be generated only if more than $t-1$ parties among all n parties are honest (Property 2), while the threshold signing process [12, 13] will aborts even if there is among all n parties only one party who honestly plays its role except refusing to present the valid signature share at the last step. Third, the threshold signature scheme of ours is proved secure in the standard model (Property 3) .

In addition to the desirable function property, our scheme also enjoys excellent efficiency property. First, the round complexity (Property 4) and the communication channel condition (Property 5) is optimal , since in the threshold signing process, the only requirements for each party is to compute and broadcast his signature share. Second, the time-consuming bilinear pairing is performed only 2 times for each signature share verification (Property 6).

Now we compare our scheme with the other ones. Baek and Zheng's scheme only enjoys Property 1, 2, but involves the Distributed Key Generation Protocol Based on the Bilinear Map sub-protocol which is very expensive in terms of time and communication. Chen et al.'s scheme only enjoys Property 1, 5. Yu et al.'s scheme only enjoys Property 5. Sun et al.'s scheme in [14] enjoys Property 1, 2, 3, 4, but involves many more pairings in the key distribution and the threshold signing process. Gao et al.'s scheme enjoys property Property 1,2,4,5,6.

6. Conclusion

We proposed a new identity-based threshold signature (IBTHS) scheme from bilinear pairings. The threshold signing protocol is optimal in terms of communication complexity and communication channel requirement. It involves no other time-consuming pairing except two pairings for verifying each signature share. It is proved secure with optimal resilience in the standard model. It is the private associated with an identity rather than a master key of the Public Key Generator (PKG) that is shared among signature generation servers. All these excellent properties are due to our new basic technique by which the private key in the bilinear group is indirectly shared through simply sharing an element in the finite field.

References

[1] Desmedt Y. Society and group oriented cryptography: a new concept, In *A Conference on the Theory and Applications of Cryptographic Techniques*, Santa Barbara, USA, 1987, pp. 120-127.

[2] Desmedt Y. Threshold cryptography, *European Transactions on Telecommunications*, 1994, 5 (4): 449–457.

[3] Shoup V. Practical threshold signatures, In *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques*, Bruges, Belgium, 2000, pp.207-220.

[4] Gennaro R, Jarecki S, Krawczyk H, Rabin T. *Robust threshold DSS signatures, Information and Computation*, 2001, 164 (1): 54–84.

[5] Fouque P A, Stern J. Fully distributed threshold RSA under standard assumptions, In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, Gold Coast, Australia, 2001, pp. 310-330.

[6] Boldyreva A. Efficient threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme, In *Proceedings of PKC 2003*, Miami, USA, 2003, pp. 31-46.

[7] Yu J, Kong F, Cheng X, et al. Forward-Secure Multisignature, Threshold Signature and Blind Signature Schemes. *Journal of Networks*, 2010, 5 (6), pp.634-641.

[8] Shamir V. Identity-based cryptosystems and signature schemes, In *Proceedings of Crypto 1984*, Santa Barbara, USA, 1984, pp. 47-53.

[9] Boneh D, Franklin M. Identity-Based encryption from the Weil pairing, In it Crypto 2001, Santa Barbara, USA, 2001, 213-229.

[10] Bellare M, Namprempre C and Neven G. Security proofs for identity-based identification and signature schemes, In *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, Switzerland, 2004, pp. 268-286.

[11] Baek J, Zheng Y. Identity-based threshold signature scheme from the bilinear pairings, In *Proceedings of the International Conference on Information Technology: Coding and Computing*, Las Vegas, USA, 2004, pp. 124-128.

[12] Chen X, Zhang F, Konidala D M, Kim K. New ID-based threshold signature scheme from bilinear pairing, In *IndoCrypt 2004*, Chennai (Madras), India, 2004, pp.371-383.

[13] Yu Y, Yang B, Sun Y. Identity-based threshold signature and mediated proxy signature schemes. *Journal of China Universities of Posts and Telecommunications*, 2007, 14 (2): 69–74.

[14] Sun X, Li J, Yang S, Chen G. Non-interactive identity-based threshold signature scheme without random oracles. *Journal of Zhejiang University-Science*, 2008, 9 (6): 727-736.

[15] Gao W, Wang G, Wang X, Yang Z. One-round ID-based threshold signature scheme from bilinear pairings, *Informatica*, 2009, 20 (4): 461–476.

[16] Paterson K G, Schuldt J C N. Efficient identity-based signatures secure in the standard model, In *Proceedings of Austrliasian Conference on Information Security and Privacy*, Melbourne, Australia, 2006, pp. 207-222.

[17] Canetti R, Goldreich O, Halevi S. The Random Oracle Methodology, Revisited, In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, Dallas, USA, 1998, pp. 209–218.

[18] Barreto P S L M, Lynn B, and Scott M. Efficient implementation of pairing-based cryptosystems, *Journal of Cryptology*, 2004, 17 (4): 321–334.