Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Non-interactive identity-based threshold signature scheme without random oracles*

Xun SUN†[1], Jian-hua LI[1,2], Shu-tang YANG[1,2], Gong-liang CHEN[2]

([1]*Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

([2]*School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

†E-mail: xun.sun.cn@gmail.com

Received Oct. 15, 2007; revision accepted Mar. 7, 2008

Abstract: A (*t*, *n*) threshold signature scheme distributes the secret key and hence the signing ability to *n* players in a way that any set of *t*+1 or more honest players can collaborate to sign, while any set of *t* players cannot. In this paper we propose an identity-based threshold signature (IBTHS) scheme from bilinear pairings. The signing phase of our scheme is non-interactive, meaning that the signing players do not need to talk to each other. We prove our scheme secure (i.e., unforgeable and robust) in the standard model (i.e., without random oracles). No earlier proposed IBTHS scheme achieved even one of the features of being non-interactive (in the signing phase) and secure in the standard model.

Key words: Bilinear pairings, Identity-based threshold signature (IBTHS), Standard model
doi:10.1631/jzus.A0720028 **Document code:** A **CLC number:** TP311

INTRODUCTION

Desmedt (1987) introduced the concept of threshold signatures. 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*+1 players can cooperatively produce a signature on behalf of the group, while up to *t* players cannot. The goal of threshold signature schemes is twofold (Gennaro *et al*., 2001): to increase the availability of signing agency by removing the single point of failure of the signer and to enhance the security against forgery by making it harder for the adversary to learn the secret key. Since Desmedt's work, researchers have taken considerable efforts to build threshold signature schemes that are both secure and efficient (Desmedt, 1994; Shoup, 2000; Gennaro *et al*., 2001; Fouque and Stern, 2001; Boldyreva, 2002; Almansa *et al*., 2006; Chu and Tzeng, 2007; Wang *et al*., 2007).

Distributed key generation (DKG) is a main component of threshold signature schemes. It allows a set of *n* players to jointly generate a pair of public and private keys according to the predefined distribution without any single entity ever knowing the secret key. This shared private key can be later used by the share holders to generate signatures, without ever being reconstructed in a single location. DKG protocols are useful not only for generating private keys, but also for collaboratively generating random numbers during threshold signing. See (Pedersen, 1991; Gennaro *et al*., 1999; Gennaro *et al*., 2003) for a series of work on DKG protocols.

Security notions of threshold signatures include unforgeability and robustness (Gennaro *et al*., 2001). The former notion extends the existential unforgeability of digital signatures under adaptively chosen message attacks due to (Goldwasser *et al.*, 1988). It requires that the adversary be unable to fake a valid signature on behalf of the group even after corrupting up to *t* players and seeing the signing process on several selected messages. The latter notion requires that the scheme be functional even if up to *t* players

* Project (No. 2005AA145110) supported by the Hi-Tech Research and Development Program (863) of China

are corrupted.

The notion of identity-based threshold signature (IBTHS) was proposed by Baek and Zheng (2004). They defined the security of the IBTHS scheme, investigated the relationship between the unforgeability of the IBTHS scheme and that of the underlying non-threshold identity-based signature (IBS) scheme, and presented a concrete implementation from bilinear pairings. After (Baek and Zheng, 2004), several IBTHS schemes have been proposed, e.g., from bilinear pairings (Chen *et al*., 2004; Cheng *et al*., 2005), GQ (Guillou-Quisquater) signature (Chu and Tzeng, 2007) and Schnorr's signature scheme (Shao *et al*., 2006).

Most of the threshold signature schemes (either in the identity-based setting or not) are proved secure in the random oracle model (Bellare and Rogaway, 1993), where hash functions are modeled by ideal oracle functions. However, it has been shown that security proof in the random oracle model does not necessarily lead to a secure protocol when the random oracles are replaced with concrete hash functions (Canetti *et al*., 1998). Hence it is desirable to devise threshold signature schemes provably secure in the standard model (i.e., without random oracles). Wang *et al.*(2005) proposed the first threshold signature scheme secure without relying on random oracles under the q-SDH (strong Diffie-Hellman) assumption. Very recently, Li *et al.*(2007) proposed a scheme also in the standard model but based on the computational Diffie-Hellman (CDH) assumption. Additionally their scheme enjoys the non-interaction property, meaning that the players need not talk to each other during the threshold signing operation. This property is desirable not only in theory, but also in real-world applications, since the entity who is responsible for outputting the final signature does not need to wait for all the signers to finish—it is sufficient to collect t+1 valid partial signatures. This means that the bottleneck incurred by the low-capability signers can be (at least partially) removed.

However, the situation for IBTHS schemes is not so good. Actually it is fair to say that devising an IBTHS scheme which achieves even one of the features of being non-interactive (in the signing phase) and secure in the standard model is still an open problem, after the work of (Baek and Zheng, 2004; Chen *et al*., 2004; Cheng *et al*., 2005; Shao *et al*., 2006; Chu and Tzeng, 2007).

We close this open problem (two open problems actually) in this paper. We propose an IBTHS scheme from bilinear pairings that is proven secure in the standard model. The scheme is based on Paterson and Schuldt (2006)'s IBS scheme. The signature generation phase of our protocol is non-interactive between the signers, while signature generation in all previous IBTHS schemes is interactive. The scheme proves to be unforgeable and robust.

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 Paterson-Schuldt IBS scheme in Section 2. Security notion of IBTHS schemes is described in Section 3. We then present our IBTHS scheme in Section 4 with formal security analysis. Section 5 presents a variant with multiple private key generators (PKGs) to remove the need for any entity to know any secret key. Section 6 concludes this paper.

PRELIMINARIES

Bilinear pairing

Let G_1 and G_2 be groups of prime order q, where we write G_1 additively and G_2 multiplicatively. Let P be a generator of G_1. A bilinear pairing is an efficiently computable map e: $G_1 \times G_1 \rightarrow G_2$, with the following properties:

(1) Bilinearity: $\forall a, b \in \mathbb{Z}_q$, $e(aP, bP)=e(P, P)^{ab}$.

(2) Non-degeneracy: $e(P, P)\neq 1$.

Modified Weil Pairing (Boneh and Franklin, 2001; 2003) and Tate Pairing (Barreto *et al*., 2002; Galbraith *et al*., 2002; Hu *et al*., 2005) over elliptic curves are working examples of bilinear pairings. For their application to cryptosystems, we refer to a survey on pairing-based cryptographic protocols due to (Dutta *et al*., 2004).

Computational Diffie-Hellman assumption

Let G_1, q and P be as described above. Given P, aP, bP where a, b are selected uniformly at random from $\mathbb{Z}_q^*$ by a challenger, the CDH problem is to find the abP.

Definition 1 (CDH assumption) The (ε, t)-CDH assumption holds in group G_1 if there is no algorithm which runs in time at most t and solves the CDH problem with probability at least ε.

Paterson-Schuldt IBS scheme

Paterson and Schuldt (2006) presented the first direct construction of IBS scheme in the standard model (hereafter referred to as the PS-IBS scheme), whose security is based on the CDH problem in the underlying group. We now briefly review this scheme as the building block of our IBTHS scheme. Here $x\in_R X$ means that x is chosen uniformly at random from X.

1. Setup. The PKG chooses groups G_1 and G_2 as described in the subsection "Bilinear pairing" with a bilinear pairing e and a generator P of G_1, then selects $x\in_R\mathbb{Z}_q^*$ and $P_2\in_R G_1$, computes $P_1=xP$. PKG also randomly selects U', M' and two vectors $\boldsymbol{U}=(U_i)$, $\boldsymbol{M}=(M_j)$ of length n_u and n_m, respectively, from G_1, where n_u and n_m are parameters unrelated to q. In the following, the identities will be represented as bit strings of length n_u, while the messages will be n_m-bit strings. PKG now publishes public parameters $\boldsymbol{params}=(G_1, G_2, q, e, P, P_1, P_2, U', \boldsymbol{U}, M', \boldsymbol{M})$, and keeps xP_2 secret as the master secret key.

2. Extraction. Given identity $u\in\{0, 1\}^{n_u}$, let $\mathcal{U}\subseteq\{1, 2, ..., n_u\}$ be the set of indices i such that the ith bit of u is 1, define function $F(u)=U'+\sum_{i\in\mathcal{U}}U_i$. PKG selects $r_u\in_R\mathbb{Z}_q^*$, then the user's private key sk_u is the tuple

$$(d_0, d_1)=(xP_2+r_uF(u), r_uP).$$

The user verifies it by evaluating $e(d_0, P)\stackrel{?}{=}e(P_2,P_1)e(F(u),d_1)$, and can re-randomize it too.

3. Signature. Let $sk_u=(d_0, d_1)$ be the user u's private key. Given a message $m\in\{0, 1\}^{n_m}$, let $\mathcal{M}\subseteq\{1, 2, ..., n_m\}$ be the set of indices j such that the jth bit of m is 1. Define function $H(m)=M'+\sum_{j\in\mathcal{M}}M_j$. Pick $r_u\in_R\mathbb{Z}_q^*$, and compute

$$V=d_0+r_mH(m),\ R_u=d_1,\ R_m=r_mP.$$

The signature is $\sigma=(V, R_u, R_m)\in G_1^3$.

4. Verification. Given a purported signature $\sigma=(V, R_u, R_m)$ on identity u and message m, evaluate

$$e(V,P)\stackrel{?}{=}e(P_2,P_1)e(F(u),R_u)e(H(m),R_m),$$

and output "valid" if the equation holds or "invalid" otherwise.

In (Paterson and Schuldt, 2006), the authors proved the security of the above scheme as the following theorem:

Theorem 1 The PS-IBS scheme is $(\varepsilon, t, q_e, q_s)$-unforgeable against adaptive chosen identity and message attack in the standard model, assuming that the CDH problem in G_1 is (ε', t')-intractable, where

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

where ρ is the time for a multiplication in G_1 and τ for an exponentiation.

DEFINITION OF IDENTITY-BASED THRESHOLD SIGNATURES

Syntax

Introduced by Baek and Zheng (2004), a (t, n) IBTHS scheme involves the PKG, a group of n users (signing entities) under the same identity, and the verifier, and is defined by the following probabilistic polynomial time (PPT) algorithms:

1. System initialization algorithm (Setup). Given a security parameter $k\in\mathbb{N}$, PKG generates the system public parameters ***params*** and the master secret key x. ***params*** are made public, while x is kept secret. ***params*** are implicit input to all the following algorithms.

2. User key extraction algorithm (Extraction). Given identity u and master key x, PKG computes the private key sk_u and sends it to the corresponding entity secretly.

3. Private key distribution algorithm (KD). Given a private key sk_u associated with an identity u, this algorithm generates n shares of sk_u and provides each one to one of the signing players $\mathcal{P}_1,\mathcal{P}_2,...,\mathcal{P}_n$ under identity u. 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.

4. Threshold signing algorithm (Thresh-Sign). Given l ($\geq t+1$) shares of the private key sk_u associated with u and a message m, l signature-generation serv-

ers jointly generate a signature σ on m.

5. Signature verification algorithm (Verification). Given an identity u, a message m and a signature σ, this algorithm checks the validity of σ to output 1 (valid) or 0 (invalid).

Security notions

Security requirement of IBTHS schemes includes both unforgeability and robustness (Baek and Zheng, 2004). We first briefly review the notion of unforgeability against chosen message and identity attack, or UF-IBTHS-CMA for short.

Definition 2 (UF-IBTHS-CMA) An IBTHS scheme is (t, n)-unforgeable if no PPT adversary who corrupts at most t players, and is given the private keys of k identities adaptively chosen, and the view of Thresh-Sign on input l (identity, message) tuples $(u_1, m_1), (u_2, m_2), \cdots, (u_l, m_l)$ adaptively chosen, can produce a signature on a message m^* under identity u^* with non-negligible probability, such that

(1) The adversary does not know the private key corresponding to u^*, under which it corrupts up to t players;

(2) $(u^*, m^*) \notin \{(u_1, m_1), (u_2, m_2), \cdots, (u_l, m_l)\}$.

To study this notion, we need the notion of simulatability of IBTHS schemes, which is first given in (Baek and Zheng, 2004) as an extension of the simulatability notion due to (Gennaro *et al*., 2001). In the following definitions, the simulators can be seen as abstractions of real adversaries, thus have as input all knowledge that the real adversaries have, including the private key shares of the corrupted players.

Definition 3 (Simulatability of IBTHS) An IBTHS scheme is simulatable if the following conditions hold:

(1) KD is simulatable. There exists a simulator SIM_{KD} that, given the system parameters ***params*** and an identity u, can simulate the view of the attacker on an execution of KD.

(2) Thresh-Sign is simulatable. There exists a simulator SIM_{TS} that, given the system parameters ***params***, an identity u, a message m, a signature σ on (u, m), t shares of sk_u of the corrupted players, and the public output of KD, can simulate the view of the attacker on an execution of Thresh-Sign.

The following theorem relates the unforgeability of a simulatable IBTHS scheme to that of the underlying non-threshold IBS scheme:

Theorem 2 (Baek and Zheng, 2004) If an IBTHS scheme is simulatable and the underlying IBS scheme is existentially unforgeable against adaptive chosen message and identity attack (UF-IBS-CMA secure), then the IBTHS scheme is UF-IBTHS-CMA secure.

Another security property of IBTHS schemes is robustness:

Definition 4 (Robustness of IBTHS) An IBTHS scheme is (h, c, n)-robust if in a group of n layers, even in the presence of an adversary who halts h players and corrupts maliciously c players, both KD and Thresh-Sign complete successfully. It may also be written as (t, n)-robust if we do not distinguish between halted and malicious players.

PROPOSED IDENTITY-BASED THRESHOLD SIGNATURE SCHEME

We now describe our IBTHS scheme based on the PS-IBS scheme, in the model of Section 3.

1. Setup. The same as the Setup algorithm of the PS-IBS scheme.

2. Extraction. The same as the Extraction algorithm of the PS-IBS scheme.

3. KD. To distribute the private key $sk_u=(d_0, d_1)$ of identity u to the group of n players, $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_n$, this protocol proceeds as follows:

(1) The clerk in possession of sk_u performs the following actions:

Step 1: Set $A_0=d_0$, choose $A_i \in_{\mathrm{R}} G_1$ for $1 \le i \le t$ and set $K(x) = A_0 + xA_1 + x^2A_2 + \cdots + x^tA_t$. Also set $B_0=d_1$, choose $B_i \in_{\mathrm{R}} G_1$ for $1 \le i \le t$ and set $L(x) = B_0 + xB_1 + x^2B_2 + \cdots + x^tB_t$.

Step 2: Compute the partial signing key $d_i^{\{0\}} = K(i) \in G_1$ and $d_i^{\{1\}} = L(i) \in G_1$ for i=1, 2, ..., n. Compute $\alpha_i = e(P, A_i) \in G_2$ and $\beta_i = e(F(u), B_i) \in G_2$ for i=0, 1, ..., t.

Step 3: Send $(d_i^{\{0\}}, d_i^{\{1\}})$ to the player $\mathcal{P}_i$ and broadcast (α_i, β_i) for i=0, 1, ..., t.

(2) Each player $\mathcal{P}_i$ first evaluates

$$\alpha_0 \stackrel{?}{=} e(P_2, P_1)\beta_0 \tag{1}$$

to make sure that the clerk is distributing a valid private key. $\mathcal{P}_i$ then verifies the share received by

evaluating

$$e(d_i^{\{0\}},P)\stackrel{?}{=}\prod_{j=0}^{t}\alpha_j^{i^j} \text{ and } e(F(u),d_i^{\{1\}})\stackrel{?}{=}\prod_{j=0}^{t}\beta_j^{i^j}. \quad (2)$$

Note that $\mathcal{P}_i$ cannot re-randomize his share $(d_i^{\{0\}},d_i^{\{1\}})$ of sk_u after receiving it from the clerk, however, this inflexibility (compared with the PS-IBS scheme) does not hurt the security of our scheme; as a matter of fact, security proof of the PS-IBS scheme in (Paterson and Schuldt, 2006) does not need the re-randomization either.

Remark 1 Here we are assuming that the clerk is honest. In the model of Section 3, the clerk, as well as the PKG who can be the same entity as the clerk, is a single point of failure and the scheme becomes insecure if the clerk is corrupted. On the other hand, the trust is not complete as the players do check whether the shares distributed by the clerk are valid.

4. Thresh-Sign. Let $\Phi\subseteq\{1, 2, ..., n\}$ be the set of presented members in the signing task, such that $|\Phi|\geq t+1$. On input a message m, this protocol proceeds as follows to create a signature on m:

(1) Each member i in Φ performs the following steps:

Step 1: Select $r_i\in_R\mathbb{Z}_q^*$, compute $V^{\{i\}}=d_i^{\{0\}}+r_iH(m)$, $R_u^{\{i\}}=d_i^{\{1\}}$, and $R_m^{\{i\}}=r_iP$.

Step 2: Broadcast (or simply send to a designated clerk) $\sigma_i=(V^{\{i\}},R_u^{\{i\}},R_m^{\{i\}})$.

(2) On receipt of each σ_i, the clerk verifies it by evaluating

$$\begin{cases} e(V^{\{i\}},P)=\prod_{j=0}^{t}\alpha_j^{i^j}e(H(m),R_m^{\{i\}}), \\ e(R_u^{\{i\}},F(u))=\prod_{j=0}^{t}\beta_j^{i^j}. \end{cases} \quad (3)$$

σ_i is valid if both equations hold. Let $QUAL\subseteq\Phi$ be the set of indices i such that σ_i is valid. If $|QUAL|\geq t+1$, the clerk computes

$$\begin{cases} V=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}V^{\{i\}}, \\ R_u=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}R_u^{\{i\}}, \\ R_m=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}R_m^{\{i\}}, \end{cases}$$

where $\lambda_{0,i}^{QUAL}=\prod_{j\in QUAL,j\neq i}j/(j-i)\ (\mathrm{mod}\ q)$ is the Lagrange interpolation coefficient. The final signature is then $\sigma=(V, R_u, R_m)$.

5. Verification. The verification is identical to the Verification algorithm of the PS-IBS scheme.

Remark 2 Note that in the Thresh-Sign phase, the clerk does not need to wait for all the players in Φ to complete but only needs to collect $t+1$ valid σ_i in order to output a full signature.

Correctness

If the protocol is executed as specified, since $V=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}V^{\{i\}}$, $R_u=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}R_u^{\{i\}}$ and $R_m=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}R_m^{\{i\}}$, first note that R_u constructed in this way is equal to d_1. It then follows that

$$\begin{aligned} e(V,P)&=\prod_{i\in QUAL}e(V^{\{i\}},P)^{\lambda_{0,i}^{QUAL}} \\ &=\prod_{i\in QUAL}\left(e(d_i^{\{0\}},P)e(H(m),R_m^{\{i\}})\right)^{\lambda_{0,i}^{QUAL}} \\ &=\prod_{i\in QUAL}\left(e(d_i^{\{0\}},P)\right)^{\lambda_{0,i}^{QUAL}}\prod_{i\in QUAL}\left(e(H(m),R_m^{\{i\}})\right)^{\lambda_{0,i}^{QUAL}} \\ &=e(d_0,P)e\left(H(m),R_m\right) \\ &=e(P_2,P_1)e\left(F(u),R_u\right)e\left(H(m),R_m\right). \end{aligned}$$

Therefore the scheme is correct.

Security proof

Theorem 3 (Robustness) The proposed scheme is (t, n)-robust if $n\geq 2t+1$.

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

In the Thresh-Sign phase, suppose up to t players are corrupted and the number of honest players is at least $n-t\geq t+1$. 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 de-

tected by Eq.(3) hence excluded from the final signature. Therefore, due to the correctness property showed in the previous subsection "Correctness", all the honest (at least $n-t\geq t+1$) players can still generate a valid signature in the presence of up to t corrupted players, and our scheme is robust if $n\geq 2t+1$.

We now proceed to prove the unforgeability of our scheme. First observe that the underlying IBS scheme is just the PS-IBS scheme as described in Section 2.

Lemma 1 The KD algorithm is simulatable.

Proof Given the identity u, we construct a simulator SIM_{KD} for the KD algorithm. Without loss of generality, we assume that the players 1, 2, ..., t are corrupted by the adversary, therefore the secret shares $(d_i^{\{0\}}, d_i^{\{1\}})$ for $i=1, 2, ..., t$ are available to SIM_{KD}. SIM_{KD} selects $\alpha_0 \in_{\mathrm{R}} G_2$. Since $e(d_i^{\{0\}}, P)=\prod_{j=0}^{t}\alpha_j^{i^j}$, α_j $(1\leq j\leq t)$ can be computed.

SIM_{KD} then computes $\beta_0=\alpha_0/e(P_2,P_1)$. Since $e(d_i^{\{1\}}, F(u))=\prod_{j=0}^{t}\beta_j^{i^j}$, β_j $(1\leq j\leq t)$ can be computed.

Lemma 2 The Thresh-Sign algorithm is simulatable.

Proof Given the identity u, message m, the corresponding signature $\sigma=(V, R_u, R_m)\in G_1^3$, and t private key shares $(d_i^{\{0\}}, d_i^{\{1\}})$ for $i=1, 2, ..., t$, we construct a simulator SIM_{TS} for the Thresh-Sign algorithm as follows. SIM_{TS} selects $r_1, r_2, ..., r_t \in_{\mathrm{R}} \mathbb{Z}_q^*$ and computes the following partial signatures (on behalf of the corrupted players):

$$\sigma_i=(V^{\{i\}}, R_u^{\{i\}}, R_m^{\{i\}})=(d_i^{\{0\}}+r_iH(m), d_i^{\{1\}}, r_iP),\quad i=1,2,\cdots,t.$$

These partial signatures are valid since they are generated with valid private key shares.

Now, SIM_{TS} defines $W(x)$ as a polynomial function of degree t such that $W(0)=V$ and $W(i)=V^{\{i\}}$, for $i=1, 2, ..., t$. Then, it can compute $V^{\{i\}}=W(i)$ for $t+1\leq i\leq n$. Similarly SIM_{TS} can compute $R_u^{\{i\}}$, $R_m^{\{i\}}$ for $t+1\leq i\leq n$. It is easy to verify that the view provided by SIM_{TS} is indistinguishable from a real execution of the Thresh-Sign algorithm. Therefore the Thresh-Sign algorithm is simulatable.

Combining Theorems 1 and 2, Lemmas 1 and 2, 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-CMA 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 G_1.

Comparison

We now compare our IBTHS scheme with previously proposed schemes BZ04 (Baek and Zheng, 2004), CZKK04 (Chen *et al*., 2004), CLW05 (Cheng *et al*., 2005), SCW06 (Shao *et al*., 2006), and CT07 (Chu and Tzeng, 2007) in terms of size of final signature, dependence on the random oracle heuristic, number of communication rounds for signing and the underlying computational assumption. The result is summarized in Table 1. Here DLP is short for "discrete logarithm problem". We note that the number of communication rounds of each scheme is computed in the absence of faults. When there are corrupted players issuing inconsistent shares, the numbers of communication rounds for the other schemes will increase, while our scheme is still non-interactive.

VARIANT WITH MULTIPLE PKGS

In our scheme proposed in Section 4, the users need to fully depend on and trust the PKG. Another single point of failure is the clerk who distributes the user private key to the group of n signers. This section aims to remove these single points of failure.

Table 1 Comparison of our scheme to previous schemes

Parameter	BZ04	CZKK04	CLW05	SCW06	CT07	Our scheme
Signature size	$G_1\times\mathbb{Z}_q^*$	$G_1^3\times G_2^3$	G_1^2	$\mathbb{Z}_p^4$	$\mathbb{Z}_n^2$	G_1^3
Random oracle	Yes	Yes	Yes	Yes	Yes	No
Interactive signing	Yes	Yes	Yes	Yes	Yes	No
Number of communication rounds	3	2	2	2	4	1
Computational assumption	CDH	CDH	CDH	DLP	RSA	CDH

To eliminate the dependence on a single PKG in identity-based cryptosystems, Boneh and Franklin (2001; 2003) proposed to distribute the master secret key to n' PKGs in a (t', n') threshold fashion therefore any t'+1 honest PKGs can cooperatively perform the private key extraction for a user. Simply applying this multi-PKG approach to our scheme would lead to either of the following two scenarios:

(1) The PKGs collectively generate a private key sk_u for u and send it to a clerk, who then distributes sk_u to the group of n players.

(2) Each PKG sends his extracted partial private key sk_i for u to a player $\mathcal{P}_i$, then $\mathcal{P}_i$ uses sk_i for signature generation.

In the first case, the key distributing clerk is still a single point of failure. In the latter case, it requires that $n'=n$ and $t'=t$, therefore the scheme does not scale well.

Desmedt and Lange (2006) proposed another approach based on the secret redistribution techniques (Desmedt and Jajodia, 1997). Their proposal eliminates the need for any party to ever know any secret key and allows n' and t' to be chosen independently of n and t.

Here we describe how Desmedt and Lange's proposal can be applied to our scheme. Assume that there are n' PKGs, denoted by $PKG_1, PKG_2, \cdots, PKG_{n'}$, and n signing players, denoted by $\mathcal{P}_1,\mathcal{P}_2,...,\mathcal{P}_n$. In this new setting, up to t' PKGs can be corrupted by the adversary, therefore the PKGs are not trusted. On the other hand, the private key extraction and distribution (Extraction) algorithm needs to be recoverable in the presence of faults. Therefore the new scheme does not work in the model of Section 3. There is no clerk who holds sk_u and distributes it among the signer group. Instead, the PKGs perform the following collaboratively and secretly:

1. Setup. On input G_1, G_2, P, q, the n' PKGs run the GJKR-DKG Protocol of (Gennaro *et al.*, 1999) to generate the master secret key x and corresponding public value $P_1=xP$, where each PKG_i holds a share $x_i\in\mathbb{Z}_q^*$ of x such that $(x_1, x_2 \cdots, x_{n'}) \xrightarrow{(t',n')} x$, the partial public keys $Q_i=x_iP$ of PKG_i are computed from publicly available information. Other public parameters P_2, U', $\boldsymbol{U}$, M', $\boldsymbol{M}$ are selected uniformly at random such that no one knows the discrete logarithm of any element to P. We may assume a trusted authority (the same one who generates the system parameters G_1, G_2, P, q) is available, or apply the distributed method of Section 4.2 of (Gennaro *et al.*, 1999) to do this.

2. Extraction. On input identity u, the PKGs collectively generate sk_u without anyone ever knowing it, and simultaneously distribute it to the group of n players as follows:

Step 1: Each PKG_i computes a partial private key for u as $sk_i=(x_iP_2 + r_iF(u),\ r_iP)$ where $r_i\in\mathbb{Z}_q^*$, then runs the KD algorithm presented in Section 4 to share sk_i among the n players in a (t, n)-threshold fashion.

Step 2: Each player $\mathcal{P}_j$ verifies the share received from PKG_i, $sk_{ij}=(d_{ij}^{\{0\}}, d_{ij}^{\{1\}})$, based on PKG_i's public data $\alpha_0^{\{i\}}, \alpha_1^{\{i\}}, \cdots, \alpha_t^{\{i\}}$ and $\beta_0^{\{i\}}, \beta_1^{\{i\}}, \cdots, \beta_t^{\{i\}}$, as in the verification operations of the KD algorithm. $\mathcal{P}_j$ broadcasts a FATAL message against PKG_i if Step 1 fails, and a message COMPLAINT against PKG_i if Step 2 fails.

Step 3: If t+1 or more FATAL messages against PKG_i are received, PKG_i is marked as "disqualified". If t+1 or more COMPLAINT messages against PKG_i are received, PKG_i is set as "disqualified". Otherwise, PKG_i publishes all the values $(d_{ij}^{\{0\}}, d_{ij}^{\{1\}})$ that $\mathcal{P}_j$ complained about. Then, Step 2 is again performed for the published values by every player. If the values do not pass Step 2, PKG_i is set to be "disqualified". Note that there are still no consequences for $\mathcal{P}_j$ at this stage since $\mathcal{P}_j$ does not need to compute anything for later use yet.

Step 4: If at least t'+1 participating PKGs are not "disqualified", denote them by $QUAL$ and the Extraction phase is successful, otherwise it fails. In the first case, each player $\mathcal{P}_j$ computes his share of the secret key

$$sk_j = \left(\sum_{i\in QUAL} \lambda_{0,i}^{QUAL} d_{ij}^{\{0\}},\ \sum_{i\in QUAL} \lambda_{0,i}^{QUAL} d_{ij}^{\{1\}} \right) = (d_j^{\{0\}}, d_j^{\{1\}}),$$

where $\lambda_{0,i}^{QUAL} = \prod_{j\in QUAL, j\neq i} j/(j-i) \pmod q$ are the Lagrange interpolation coefficients corresponding to the (t', n') threshold scheme.

Anyone can then compute $\alpha_k = \prod_{i \in QUAL} \alpha_k^{\{i\}\lambda_{0,i}^{QUAL}}$ and $\beta_k = \prod_{i \in QUAL} \beta_k^{\{i\}\lambda_{0,i}^{QUAL}}$ for k=0, 1, ..., t. α_k and β_k are used to verify partial signatures in the Thresh-Sign phase.

3. Thresh-Sign. The same as the Thresh-Sign algorithm of the scheme of Section 4.

4. Verification. The same as the Verification algorithm of the scheme of Section 4.

Correctness of this scheme is easily verified.

Security model

To argue about the security of this scheme, we first extend the formal model of Section 3 to capture multiple PKGs. We name this kind of signatures as IBTHS with multiple PKGs (IBTHSmPKG).

1. Syntax. A (t', n'; t, n)-IBTHSmPKG scheme is a tuple of PPT algorithms (Setup, Extraction, Thresh-Sign, Verification). Setup is run by the group of n' PKGs to collectively generate the system parameters and each PKG's share of master secret key. Extraction is run by the PKGs on input an identity u to issue each signing player one share of the private key corresponding to u. Thresh-Sign and Verification bear the same functionalities as in the model of Section 3.

2. Security notions. Similar to that of IBTHS schemes, security notions of IBTHSmPKG schemes include also unforgeability and robustness. Here we only consider the unforgeability property since the robustness property is more straightforward to define and study.

Let $\mathcal{A}$ be a PPT attacker against a (t', n'; t, n)-IBTHSmPKG scheme. Consider the following game in which $\mathcal{A}$ interacts with the challenger $\mathcal{C}$:

Step 1: $\mathcal{A}$ corrupts t' PKGs and t signature generation players.

Step 2: $\mathcal{C}$ runs the Setup phase of the scheme on behalf of the honest PKGs, by interaction with the corrupted PKGs controlled by $\mathcal{A}$.

Step 3: $\mathcal{A}$ issues a number of private key extraction queries, each of which consists of identity u. To respond to such a query, $\mathcal{C}$ runs the key extraction algorithm "Extraction" of the IBTHSmPKG scheme taking u as the input. As a result, the corrupted signing players get their respective secret key shares.

Step 4: $\mathcal{A}$ issues a number of signature generation queries, each of which consists of an identity u and a message m. To respond, $\mathcal{C}$ first runs the key extraction algorithm "Extraction" of the IBTHSmPKG scheme taking u as the input, then on behalf of the uncorrupted signing players, runs the Thresh-Sign algorithm and responds to $\mathcal{A}$ with the resulting signature. Note that in this phase $\mathcal{A}$ is also allowed to issue private key extraction queries.

Step 5: Finally $\mathcal{A}$ outputs (u^*, m^*, σ^*). $\mathcal{A}$ succeeds if the following conditions hold: (1) σ^* is a valid signature on message m^* and identity u^*; (2) $\mathcal{A}$ has not made a signature generation query for m^* and u^*.

$\mathcal{A}$'s advantage in the above game is defined as

$$Adv_{\mathcal{A}} = Pr[\mathcal{A} \text{ succeeds}],$$

where the probability is taken over all coin tosses made by $\mathcal{C}$ and $\mathcal{A}$.

Definition 5 (UF-IBTHSmPKG-CMA) A (t', n'; t, n)-IBTHSmPKG scheme is unforgeable if no PPT adversary can have non-negligible advantage in winning the above game.

To study the unforgeability property, we define the simulatability of IBTHSmPKG schemes similar to IBTHS schemes.

Definition 6 (Simulatability of IBTHSmPKG) An IBTHSmPKG scheme is simulatable if the following conditions hold:

(1) Setup is simulatable: There exists a simulator SIM_{Setup} that can simulate the view of the attacker on an execution of Setup.

(2) Extraction is simulatable: There exists a simulator SIM_{Extract} that, given the system parameters ***params*** and an identity u, can simulate the view of the attacker on an execution of Extraction.

(3) Thresh-Sign is simulatable: There exists a simulator SIM_{TS} that, given the system parameters ***params***, an identity u, a message m, a signature σ on (u, m), and the public output of Extraction, can simulate the view of the attacker on an execution of Thresh-Sign.

Now we can derive the following theorem on the unforgeability of a simulatable IBTHSmPKG scheme. The proof is by analogy with that of Theorem 1 in (Baek and Zheng, 2004).

Theorem 5 If an IBTHSmPKG scheme is simulatable and the underlying IBS scheme is existentially unforgeable against adaptive chosen message and

identity attack (UF-IBS-CMA secure), the IBTHSmPKG scheme is UF-IBTHSmPKG-CMA secure.

Security result

Lemma 3 Our IBTHSmPKG scheme is simulatable.

Proof We show the following conditions.

1. Assume that the public parameters P_2, U', $\boldsymbol{U}$, M', $\boldsymbol{M}$ of the scheme are selected by a trusted authority (the same one who generates the system parameters G_1, G_2, P, q), then the Setup algorithm of our scheme is essentially the DKG protocol of (Gennaro *et al.*, 1999), therefore it is simulatable.

2. The Extraction algorithm of our scheme is simulatable. Actually, we can construct a simulator SIM_{Extract} for this algorithm. Let the corrupted PKGs be indexed by 1, 2, …, t' and the corrupted signing players be indexed by 1, 2, …, t. On input an identity u, SIM_{Extract} works as follows:

Step 1: Perform Step 1 of the Extraction algorithm on behalf of all the corrupted PKGs. As a result, $sk_{ij}=(d_{ij}^{\{0\}},d_{ij}^{\{1\}})$ for i=1, 2, …, t' and j=1, 2, …, n are available to the simulator.

Step 2: Define $QUAL$ as {1, 2, …, t'+1}. For each j=1, 2, …, t, parse the corrupted signing player $\mathcal{P}_j$'s secret key share as $sk_j=(d_j^{\{0\}},d_j^{\{1\}})$. Since $d_j^{\{0\}}=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}d_{ij}^{\{0\}}$ and $d_j^{\{1\}}=\sum_{i\in QUAL}\lambda_{0,i}^{QUAL}d_{ij}^{\{1\}}$, $d_{ij}^{\{0\}}$ and $d_{ij}^{\{1\}}$ for $i=t'+1$ can be computed. By defining $QUAL$ differently, the values of $d_{ij}^{\{0\}}$ and $d_{ij}^{\{1\}}$ for $i=t'+2, t'+3, \ldots, n'$ can be computed.

Step 3: For each $i=t'+1, t'+2, \ldots, n'$, take u and $sk_{ij}=(d_{ij}^{\{0\}},d_{ij}^{\{1\}})$ for j=1, 2, …, t as the input, run the simulator SIM_{KD} for the KD algorithm constructed in Lemma 1.

3. The Thresh-Sign algorithm of our scheme is identical to that of the scheme in Section 4, therefore it is simulatable.

Therefore our IBTHSmPKG scheme is simulatable.

Combining Theorems 1 and 5, and Lemma 3, we now obtain the following theorem on the unforgeability of our IBTHSmPKG scheme:

Theorem 6 (Unforgeability) In the standard model, our proposed IBTHSmPKG scheme is UF-IBTHSmPKG-CMA secure, if the CDH problem is intractable in the underlying pairing friendly group G_1.

Other extensions

1. More efficient setup. Using the GJKR-DKG protocol (Gennaro *et al.*, 1999) for distributed initialization in the Setup phase ensures that the scheme is simulatable in terms of Definition 6. Hence its security can be related to that of the underlying PS-IBS scheme. It has been shown in (Gennaro *et al.*, 2003) that the more efficient Pedersen-DKG protocol (Pedersen, 1991) can be used as a replacement of GJKR-DKG for the threshold DSS scheme, with the security of the resulting threshold DSS scheme directly reduced to the discrete logarithm problem. We can apply this methodology to our scheme to improve efficiency of the Setup phase. Then, in the security proof, to embed the CDH problem instance (P, aP, bP) into the challenge public key, the simulator (challenger) sets $P_2=bP$, picks an honest PKG, say $PKG_{n'}$, and sets its partial public key as $Q_{n'}=aP$. To simulate a real challenger for the adversary $\mathcal{A}$, the simulator works similarly to that of (Gennaro *et al.*, 2003). When $\mathcal{A}$ finally outputs a forgery σ^* on identity u^* and message m^*, the simulator generates partial signatures σ_i on behalf of the uncorrupted PKGs, PKG_i on u^* and m^* for $i=t'+1, t'+2, \ldots, n'-1$ (recall that, in an IBS scheme the PKG can sign on behalf of any user on any message using its master secret key). Then the simulator computes the partial signature $\sigma_{n'}$ corresponding to $PKG_{n'}$, and applies the process of (Paterson and Schuldt, 2006) to solve the CDH problem.

2. Adding adaptive security. The above scheme, as well as many other threshold cryptosystems, works in the static adversary model, where the adversary decides and fixes the subset of the corrupted parties at the beginning of the protocol run. To achieve adaptive security which captures more realistic attacks, we substitute the adaptively secure DKG protocol of (Canetti *et al.*, 1999) for the GJKR-DKG protocol. Intuitively, our Extraction and Thresh-Sign algorithms are both adaptively secure since during the execution of both algorithms, there is no interaction among either the participating PKGs or the participating signing players, therefore it does not matter whether an entity is corrupted earlier or later.

CONCLUSION

In this paper we presented an identity-based threshold signature scheme from bilinear pairings, which proves to be secure (robust and unforgeable) in the standard model. Signature generation in our scheme is very efficient since it requires no interaction between the signing players. We also showed how to modify this scheme to work with multiple PKGs to remove the single point of failure on a single PKG or a private key distributing clerk.

References

Almansa, J.F., Damgård, I., Nielsen, J.B., 2006. Simplified threshold RSA with adaptive and proactive security. *LNCS*, **4004**:593-611. [doi:10.1007/11761679_35]

Baek, J., Zheng, Y., 2004. Identity-based Threshold Signature Scheme from the Bilinear Pairings. Proc. Int. Conf. on Information Technology: Coding and Computing. IEEE Computer Society, p.124-128. [doi:10.1109/ITCC.2004.1286437]

Barreto, P., Kim, H., Lynn, B., Scott, M., 2002. Efficient algorithms for pairing-based cryptosystems. *LNCS*, **2442**:354-368. [doi:10.1007/3-540-45708-9_23]

Bellare, M., Rogaway, P., 1993. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. Proc. First Annual Conf. on Computer and Communications Security. ACM Press, p.62-73.

Boldyreva, A., 2002. Efficient threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. *LNCS*, **2567**:31-46. [doi:10.1007/3-540-36288-6_3]

Boneh, D., Franklin, M., 2001. Identity-based encryption from the Weil pairing. *LNCS*, **2139**:213-229. [doi:10.1007/3-540-44647-8_13]

Boneh, D., Franklin, M., 2003. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, **32**(3):586-615. [doi:10.1137/S0097539701398521]

Canetti, R., Goldreich, O., Halevi, S., 1998. The Random Oracle Methodology, Revisited. Proc. 30th ACM Annual Symp. on Theory of Computing. ACM Press, p.209-218.

Canetti, R., Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T., 1999. Adaptive security for threshold cryptosystems. *LNCS*, **1666**:98-116.

Chen, X., Zhang, F., Konidala, D.M., Kim, K., 2004. New ID-based threshold signature scheme from bilinear pairing. *LNCS*, **3348**:371-383.

Cheng, X., Liu, J., Wang, X., 2005. An Identity-Based Signature and its Threshold Version. Proc. 19th Int. Conf. on Advanced Information Networking and Applications, p.973-977.

Chu, C.K., Tzeng, W.G., 2007. Optimal resilient threshold GQ signatures. *Inf. Sci.*, **177**:1834-1851. [doi:10.1016/j.ins.2006.11.001]

Desmedt, Y., 1987. Society and group oriented cryptography: a new concept. *LNCS*, **293**:120-127.

Desmedt, Y., 1994. Threshold cryptography. *Eur. Trans. on Telecommun.*, **5**(4).

Desmedt, Y., Jajodia, S., 1997. Redistributing Secret Shares to New Access Structures and its Applications. Technical Report ISSE-TR-97-01, George Mason University.

Desmedt, Y., Lange, T., 2006. Pairing based threshold cryptography improving on Libert-Quisquater and Baek-Zheng. *LNCS*, **4107**:154-159. [doi:10.1007/11889663_12]

Dutta, R., Barua, R., Sarkar, P., 2004. Pairing-Based Cryptographic Protocols: A Survey. Cryptology ePrint Archive.

Fouque, P.A., Stern, J., 2001. Fully distributed threshold RSA under standard assumptions. *LNCS*, **2248**:310-330. [doi:10.1007/3-540-45682-1_19]

Galbraith, S.D., Harrison, K., Soldera, D., 2002. Implementing the Tate pairing. *LNCS*, **2369**:324-337. [doi:10.1007/3-540-45455-1_26]

Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T., 1999. The (in)security of distributed key generation in Dlog-based cryptosystems. *LNCS*, **1592**:295-310.

Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T., 2001. Robust threshold DSS signatures. *Inf. & Comput.*, **164**(1):54-84. [doi:10.1006/inco.2000.2881]

Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T., 2003. Secure applications of Pedersen's distributed key generation protocol. *LNCS*, **2612**:373-390. [doi:10.1007/3-540-36563-X_26]

Goldwasser, S., Micali, S., Rivest, R.L., 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, **17**(2):281-308. [doi:10.1137/0217017]

Hu, L., Dong, J.W., Pei, D.Y., 2005. Implementation of cryptosystems based on Tate pairing. *J. Computer Sci. & Technol.*, **20**(2):264-269. [doi:10.1007/s11390-005-0264-1]

Li, J., Yuen, T.H., Kim, K., 2007. Practical threshold signatures without random oracles. *LNCS*, **4784**:198-207. [doi:10.1007/978-3-540-75670-5_14]

Paterson, K.G., Schuldt, J.C.N., 2006. Efficient identity-based signatures secure in the standard model. *LNCS*, **4058**:207-222. [doi:10.1007/11780656_18]

Pedersen, T., 1991. A threshold cryptosystem without a trusted party. *LNCS*, **547**:522-526.

Shao, J., Cao, Z., Wang, L., 2006. Efficient ID-Based Threshold Signature Schemes Without Pairings. Http://eprint.iacr.org/2006/308

Shoup, V., 2000. Practical threshold signatures. *LNCS*, **1807**:207-220.

Wang, H., Zhang, Y., Feng, D., 2005. Short threshold signature schemes without random oracles. *LNCS*, **3797**:297-310. [doi:10.1007/11596219_24]

Wang, L., Cao, Z., Li, X., Qian, H., 2007. Simulatability and security of certificateless threshold signatures. *Inf. Sci.*, **177**(6):1382-1394. [doi:10.1016/j.ins.2006.08.008]