Provably Secure Identity-based Threshold Key Escrow from Pairing

Yu Long, Zheng Gong, Kefei Chen, and Shengli Liu
(Corresponding author: Yu Long)

Department of Computer Science and Engineering, Shanghai Jiao Tong University
Postbox 0403391, 800 Dongchuan Road, Shanghai 200240, China
(Email: {longyu, neoyan, kfchen, slliu}@sjtu.edu.cn)
(Received Sept. 13, 2006; revised Feb. 1, 2007; and accepted May 28, 2007)

Abstract

This paper proposes an identity-based threshold key escrow scheme. The scheme is secure against identity-based threshold chosen-plaintext attack. It tolerates the passive adversary to access data of corrupted key escrow agency servers and the active adversary that can modify corrupted servers' keys. The formal proof of security is presented in the random oracle model, assuming the Bilinear Diffie-Hellman problem is computationally hard.
Keywords: Chosen-plaintext attack, identity-based cryptography, pairing based cryptology, threshold key escrow

1 Introduction

During the last decade there has been a large growth in communication over the Internet. There has also been an increased focus on privacy and sending messages encrypted. This however poses a problem for law enforcement agencies that have relied on their ability to make wiretaps and get warrants to solve crime. This has lead to the concept of key escrow [8].

In 2001, D. Boneh and M. Franklin proposed the first practical identity-based encryption (IBE [5]) system from the weil pairing. It provides a public key encryption mechanism where an arbitrary string can be served as the public key. The direct derivation of public keys in identity-based public key cryptography (IB-PKC) eliminates the need for certificates. On the other hand, IB-PKC has an inherent problem of key escrow since a trusted third party named the Private Key Generator (PKG), who uses the master key to generate private keys for every entity. To solve this problem, S.S.Al-Riyami and K.G.Paterson introduced the concept of certificateless public key cryptography (CL-PKC [13]). In [15], a more efficient certificateless public key encryption scheme was proposed. In CL-PKC, the private key of every entity is created by the PKG and each entity unitedly. However, the law enforcement agency is unable to monitor communications in such a scheme.

A way to solve the contradiction is to share the power of monitoring among a set of key escrow agencies (KEAs). Although the PKG generates private keys for all users in the system, it will not be given access to any ciphertexts. To monitor the communications of some entity needs at least a threshold value KEAs' co-operation. We propose such a scheme based on the difficulty of the Bilinear Diffie-Hellman problem, named ID-based threshold key escrow (`IB-ThKE`) scheme. This scheme is provably secure in the ID-based threshold chosen-plaintext attack model.

A related concept of ID-based threshold key escrow is threshold decryption. To our knowledge, other papers that have treated this concept in the context of ID-base cryptography are [1, 7, 11]. However the formal security proof of [7] is based on a weaker model named selective identity (ID) threshold chosen-plaintext attack model. Furthermore, all of these schemes can not resist against active attackers.

2 Preliminaries

2.1 Admissible Bilinear Pairings

Let $\mathbb{G}_1$ be a cyclic additive group and $\mathbb{G}_2$ be a cyclic multiplicative group of the same prime order q. Assume that the discrete logarithm problem in both $\mathbb{G}_1$ and $\mathbb{G}_2$ are hard. An admissible bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ which satisfies the following properties:

- Bilinear: for any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- Non-degenerate: there exists $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.
- Computable: given $P, Q \in \mathbb{G}_1$, there is an efficient algorithm to compute $\hat{e}(P, Q) \in \mathbb{G}_2$.

2.2 Bilinear Diffie-Hellman Problem

Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be an admissible bilinear map. Let P be a generator of $\mathbb{G}_1$, whose order is a large prime q. Let a, b, c be elements of $\mathbb{Z}_q^*$.

Definition 1. *Given (P, aP, bP, cP), compute $\hat{e}(P,P)^{abc}$. An algorithm $\mathcal{A}$ has an advantage ϵ in solving the Bilinear Diffie-Hellman Problem (BDHP) in $< \mathbb{G}_1, \mathbb{G}_2, \hat{e} >$ if $Pr[\mathcal{A}(P, aP, bP, cP) = \hat{e}(P,P)^{abc}] > \epsilon$.*

In general, BDHP is believed to be hard in $< \mathbb{G}_1, \mathbb{G}_2, \hat{e} >$. That means there is no probabilistic algorithm can solve BDHP with a non-negligible advantage ϵ in polynomial time.

2.3 Threshold Security

The idea of (t, n) threshold cryptosystem was proposed in [14]. The formal security model of threshold cryptosystems has been discussed in [9, 16]. In a threshold setting, the latent threat of decryption shares' leaking leads it's very difficult to design a threshold chosen-ciphertext secure scheme.

As [9] pointed out, the adversary of threshold cryptosystems can act passively or actively. The passive adversary accesses only to internal data of some corrupted severs (e.g. getting the private keys of decryption servers in a threshold decryption cryptosystem). And the active adversary can modify behaviors of them (e.g. replacing the keys of decryption servers in a threshold decryption cryptosystem).

3 ID-based (t, n) Threshold Key Escrow Scheme

In this paper, we propose an ID-based (t, n) threshold key escrow scheme (`IB-ThKE`). The system consists of a trusted authority called the Private Key Generator (PKG), n key escrow agency servers (KEAs) and many communication users. The secret key of each user is issued through secret channels by the PKG, and the public key is the unambiguous identity of the user. It means the ciphertexts in the communication phase are outputs of an identity-based encryption algorithm. The plaintext M that is encrypted under an identity is recoverable from at least t of n KEAs. We assume that the PKG has no access to any ciphertext, since the PKG knows the private key of each user.

3.1 Defining IB-ThKE

First, we sketch the characteristics of IB-ThKE.

In every user's view, `IB-ThKE` is similar to traditional identity-based public key cryptosystems. The public key is an arbitrary sting such as an email address or a telephone number, so there is no need of certificates.

Each KEA has a private key chosen by himself and the corresponding public key is given to the PKG. When at least t KEAs want to monitor an user *Alice*'s received ciphertext, the PKG returns partial secret keys and public verification keys of *Alice* to them. Then each KEA can generate a decryption share of this ciphertext with the partial secret key and his private key, after checking the validity of the partial secret key. These shares are sent to a special server called the *combiner*, who starts checking the validity of every share. If more than t shares are valid, the combiner combines them to obtain the plaintext.

3.2 Description of this Scheme

In [5], D. Boneh and M. Franklin constructed a chosen-plaintext secure ID-based encryption scheme named `BasicIdent`. The same idea is used in the communications among users in our scheme. The threshold key escrow based on `BasicIdent` is described as follows:

Setup: Run by the PKG and n KEAs $\Gamma_i (i = 1, 2, \cdots, n)$.

- Given a security parameter k_0, the PKG outputs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order $q (\geq 2^{k_0})$, an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, a generator $P \in \mathbb{G}_1$, a master key $s \in \mathbb{Z}_q^*$. Compute $P_{pub} = sP$ and choose three hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^l$ and $H_3 : \mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$. Note that H_1, H_2 are viewed as random oracles[2] in the security analysis.
- Key escrow agency server $\Gamma_i (i = 1, 2, \cdots, n)$ randomly selects $s_i \in \mathbb{Z}_q^*$ and computes $P_i = s_i P$.

The system public parameters are:

$$cp = \{q, l, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_1, H_2, H_3, P_{pub}, \{P_1, P_2, \cdots, P_n\}\}.$$

KeyGen1: Given an user's identity ID, the PKG returns $d_{ID} = sH_1(ID)$ to this user secretly as his private decryption key.

KeyGen2: Given an user's identity ID and a request for monitoring this user's communication, the PKG chooses a polynomial of degree $t-1$ in $\mathbb{Z}_q$:

$$f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1},$$

where $a_1, \ldots, a_{t-2} \in_R \mathbb{Z}_q$ and $a_{t-1} \in_R \mathbb{Z}_q^*$. For $1 \leq i \leq n$, it computes $P_{ID}^{(i)} = f(i) Q_{ID} + sP_i \in \mathbb{G}_1$ and $V_{ID}^{(i)} = \hat{e}(f(i) Q_{ID}, P)$. $P_{ID}^{(i)}$ are returned to n KEAs secretly and $V_{ID}^{(i)}$ are published.

Encryption: To encrypt a message $M \in \{0,1\}^l$ under the receiver's identity ID, the sender computes $Q_{ID} = H_1(ID)$ and randomly chooses $r \in \mathbb{Z}_q^*$. Then set the ciphertext to be $< U, V > = < rP, M \oplus H_2(\hat{e}(P_{pub}, Q_{ID})^r) >$.

User's Decryption: Let $C =< U, V >$ be a ciphertext with the public key ID. To decrypt C with the the corresponding private key d_{ID}, the receiver computes:

$$V \oplus H_2(\hat{e}(d_{ID}, U)) = M.$$

KEA's Sub-Decryption: Given a ciphertext $C =< U, V >$ and n partial key pairs $(P_{ID}^{(i)}, V_{ID}^{(i)})(i = 1, 2, \cdots, n)$, KEA Γ_i checks the validity of $(P_{ID}^{(i)}, V_{ID}^{(i)})$ and computes his decryption share as follows:

- Each KEA can check the validity of $(P_{ID}^{(i)}, V_{ID}^{(i)})$ by $\hat{e}(P_{ID}^{(i)}, P) = V_{ID}^{(i)} \cdot \hat{e}(P_i, P_{pub})$. And everybody can check $\prod_{i \in T}(V_{ID}^{(i)})^{L_i^T} = \hat{e}(Q_{ID}, P_{pub})$ for any subsets $T \subset \{1, 2, \cdots, n\}$ such that $|T| = t$, where L_i^T denotes the appropriate Lagrange coefficient with respect to the set T.
- If $(P_{ID}^{(i)}, V_{ID}^{(i)})(i = 1, 2, \cdots, n)$ can pass above test, Γ_i computes $k_{ID}^i = \hat{e}(P_{ID}^{(i)} - s_i(P_{pub}), U)$, $R_i = \hat{e}(T_i, P)$, $\tilde{R}_i = \hat{e}(T_i, U)$, $h_i = H_3(V_{ID}^{(i)}, k_{ID}^i, R_i, \tilde{R}_i), \lambda_i = T_i + h_i(P_{ID}^{(i)} - s_i(P_{pub}))$ for random $T_i \in \mathbb{G}_1^*$. Then output the decryption share $\delta_{ID,C}^i = \{i, k_{ID}^i, R_i, \tilde{R}_i, \lambda_i\}$.
- Otherwise, Γ_i returns $\delta_{ID,C}^i = \{i, ID, invalid\}$.

Combination: Given a ciphertext $C =< U, V >$ and a set of decryption shares $\{\delta_{ID,C}^i\}_{i \in T}$ where $|T| = t$, the combiner runs as follows:

- For $i \in T$, compute $h_i = H_3(V_{ID}^{(i)}, k_{ID}^i, R_i, \tilde{R}_i)$. Check if $\hat{e}(\lambda_i, P) = R_i \cdot (V_{ID}^{(i)})^{h_i}$ and $\hat{e}(\lambda_i, U) = \tilde{R}_i \cdot (k_{ID}^i)^{h_i}$.
- If the above test holds, the combiner computes $K = \prod_{i \in T}(k_{ID}^i)^{L_i^T}$ and $M = V \oplus H_2(K)$. Then the ciphertext is decrypted by key escrow agency servers.

KEA's public key updating: In this scheme,we allow KEA $\Gamma_i(i \in \{1, 2, \cdots, n\})$ to renew his private key s_i as follows:

- Γ_i chooses $s_i' \in \mathbb{Z}_q^*$. Compute $P_i' = s_i' P$ and $\Delta_i = s_i' P_{pub}$. Then pass $< i, P_i', \Delta_i >$ to the PKG secretly.
- The PKG checks the validity of P_i' by $\hat{e}(P_i', P_{pub}) = \hat{e}(\Delta_i, P)$. If it holds, the PKG changes P_i to be P_i' publicly and renews $P_{ID}^{(i)}$ in *KeyGen2* accordingly. Else the PKG refuses Γ_i's request.

Note that we use a non-interactive zero knowledge proof (NIZK[1, 4]) to check the validity of decryption shares from every KEA Γ_i. Please refer to appendix A for more details.

4 Security Analysis

4.1 Adversary Types

To give the formal definition of the `IB-ThKE` scheme, we need to define adversaries for it. Since the communication users and KEAs have different views in this scheme, we will distinguish between two adversary types:

1) **IB-ThKE Type 1 Adversary**. Such an adversary $\mathcal{A}_1$ is against the underlying identity-based public key cryptosystem. Since `BasicIdent` is used in `IB-ThKE`, $\mathcal{A}_1$ is defined the same as in [5].
2) **IB-ThKE Type 2 Adversary**. Such an adversary $\mathcal{A}_2$ is to attack KEAs. Since we use a (t, n) threshold scheme, it's reasonable to assume that at most $t-1$ out of n KEAs will be corrupted by $\mathcal{A}_2$. Let $\{\Gamma_i\}_{i \in S, |S|=t-1}$ be the set of corrupted KEAs, the $\mathcal{A}_2$'s actions against `IB-ThKE` are listed below:
   - KEAs' private key extraction queries: $\mathcal{A}_2$ is allowed to make requests for $\Gamma_i(i \in S)$'s private keys s_i.
   - Complete private key extraction queries: $\mathcal{A}_2$ is allowed to query on an identity ID's private decryption key. However, it's unreasonable for $\mathcal{A}_2$ to extract the complete private key of the selected challenge identity ID_{ch}.
   - Partial key queries: Given a ciphertext and an identity ID, $\mathcal{A}_2$ can ask for partial keys $P_{ID}^{(i)}, V_{ID}^{(k)}$, for $i \in S$ and $k \in \{1, 2, \cdots, n\}$.
   - Replace KEAs' public keys: Since Γ_i's public key $P_i = s_i P(i = 1, 2, \cdots, n)$ is not associated with Γ_i's identity, $\mathcal{A}_2$ can choose any $s_i' \in \mathbb{Z}_q^*$ and try to replace P_i by $P_i' = s_i' P$ for corrupted KEAs.

4.2 Security Model for IB-ThKE

In this section, we give the formal security definition and proof of `IB-ThKE` scheme. The full security model is constructed of two distinct parts. One is the interactions between the challenger and $\mathcal{A}_1$. It is selfsame as [5], and the `BasicIdent` scheme has been proved to be chosen-plaintext secure. So we only discuss the other part, which are the interactions between the challenger and $\mathcal{A}_2$.

Definition 2. (IND-IDTH-CPA). *The ID-based (t, n) threshold key escrow scheme is secure against chosen-plaintext attack (denoted by **IND-IDTH-CPA**) if no polynomially bounded adversary has a non-negligible advantage in the following game:*

***Init**: The adversary $\mathcal{A}_2$ chooses a set S of $t-1$ players it wants to corrupt.*

***Setup**: The challenger runs* Setup *algorithm and gives the resulting common parameters to $\mathcal{A}_2$, including the public key P_i of $\Gamma_i(i = 1, 2, \cdots, n)$.*

***KEA's private key extraction queries**: Given S, the challenger generates $t-1$ KEAs' private keys $s_i(i \in S)$. Send (i, s_i) to $\mathcal{A}_2$.*

Phase1: *$\mathcal{A}_2$ chooses $\{ID_1, ID_2, \cdots, ID_m\}$. On an identity $ID \in \{ID_1, ID_2, \cdots, ID_m\}$, $\mathcal{A}_2$ performs a number of queries adaptively:*

- *Complete private key extraction queries: the challenger generates complete decryption key d_{ID}. Send it to $\mathcal{A}_2$.*
- *Partial key queries: the challenger returns $P_{ID}^{(i)}$ for $i \in S$ and $V_{ID}^{(j)}$ for $1 \leq j \leq n$.*
- *Replace KEA's public keys: For $i \in S$, suppose the request is to replace the public key of Γ_i with $< P_i' = s_i' P, \Delta_i = s_i' P_{pub} >$. The challenger accepts $\mathcal{A}_2$'s request. When receiving ID, the challenger returns $P_{ID}^{(i)'}$ associated with P_i' and ID. $V_{ID}^{(i)}$ keeps unmodified.*

Challenge: *$\mathcal{A}_2$ chooses two equal length plaintexts M_0, M_1 and an identity ID_{ch} which it wishes to be challenged on. It's not allowed to choose an identity on which $\mathcal{A}_2$ made a complete private key extraction query during the* Phase1. *The challenger picks a bit $b' \in \{0,1\}$ uniformly and responds a `IB-ThKE` ciphertext $C^* =< U, V >$, such that C^* is the encryption of $M_{b'}$. It sends C^* to $\mathcal{A}_2$.*

Phase2: *$\mathcal{A}_2$ chooses $\{ID_{m+1}, ID_{m+2}, \cdots, ID_k\}$ and performs a number of queries as in* Phase1, *except the complete private key of ID_{ch}.*

Guess: *$\mathcal{A}_2$ outputs a guess $b'' \in \{0,1\}$. $\mathcal{A}_2$ wins if $b'' = b'$.*

The adversary $\mathcal{A}_2$'s advantage is defined to be:

$$Adv(\mathcal{A}_2) = |2Pr[b'' = b'] - 1|.$$

Theorem 1. *Suppose the hash functions H_1, H_2 are random oracles. Then `IB-ThKE` is an **IND-IDTH-CPA** secure scheme assuming the BDH problem is hard in groups generated by* Setup. *Concretely, suppose there is a type2 **IND-IDTH-CPA** adversary $\mathcal{A}_2$ that has advantage ϵ against the `IB-ThKE`. Suppose $\mathcal{A}_2$ makes at most q_E complete private key queries, q_{H_1} hash queries to H_1 and q_{H_2} hash queries to H_2. Then there is an algorithm $\mathcal{C}$ that solves the BDH problem in groups generated by* Setup *with an advantage at least $\epsilon'' = \epsilon(q_{H_1} - q_E)/(q_{H_1})^2 q_{H_2}$.*

Proof. To prove this theorem we first define a related public key threshold decryption scheme called `BasicThIBE`. It is described as follows:

KeyGen: Given a security parameter k_0, the PKG chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order $q \geq 2^{k_0}$, an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, a generator $P \in \mathbb{G}_1$, a secret key $s \in_R \mathbb{Z}_q^*$, $P_{pub} = sP$. Then the PKG chooses a polynomial of degree $t-1$ over $\mathbb{Z}_q^*$:

$$f(x) = s + a_1 x + \ldots + a_{t-1} x^{t-1}.$$

For $i = 1, 2, \cdots, n$, it computes $P_{pub}^{(i)} = f(i)P \in \mathbb{G}_1$ and chooses one cryptographic hash function $H_2 : \mathbb{G}_2 \to \{0,1\}^l$. Pick a random $Q \in \mathbb{G}_1^*$. The public parameters are:

$$cp = \{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_2, P, P_{pub}, Q, \{P_{pub}^{(1)}, P_{pub}^{(2)}, \ldots, P_{pub}^{(n)}\}\}.$$

For $i = 1, 2, \cdots, n$, the PKG delivers $d_i = f(i)Q \in \mathbb{G}_1$ to decryption server i secretly. When receiving d_i, server i can check its validity by $\hat{e}(P_{pub}^{(i)}, Q) = \hat{e}(d_i, P)$ and $\sum_{i \in T} L_i^T (P_{pub}^{(i)}) = P_{pub}$. Where $T \subset \{1, 2, \cdots, n\}$, $|T| = t$ and L_i^T is the Lagrange coefficient with respect to the set T. If the verification fails, he complains to the PKG that issues a new share.

Encrypt: To encrypt a message $m \in \{0,1\}^l$, the sender chooses a random $r \in_R \mathbb{Z}_q^*$. The ciphertext is given by $< U, V >=< rP, m \oplus H_2(\hat{e}(P_{pub}, Q)^r) >$.

Decrypt: When receiving $< U, V >$, decryption server i computes his decryption share $\delta_i = \hat{e}(U, d_i)$ and gives it to the combiner.

Combination: The combiner selects a set $T \subset \{1, 2, \cdots, n\}$ of t acceptable decryption shares δ_i and computes $g = \prod_{i \in T} \delta_i^{L_i^T}$. Then the plaintext can be recovered by $m = V \oplus H_2(g)$. The correctness of this scheme is easy to verify.

The `BasicThIBE` scheme is not an identity-based(non-ID-based) scheme but a traditional public key cryptosystem. The key pair is $< sQ, Q >$.

Definition 3. (IND-TH-CPA). *A non-ID-based threshold decryption scheme is secure against chosen-plaintext attacks (denoted by **IND-TH-CPA**) if no polynomially bounded adversary $\mathcal{B}$ has a non-negligible advantage in the following game:*

Init: *$\mathcal{B}$ corrupts a fixed subset of $t-1$ servers.*

KeyGen: *$\mathcal{B}$'s challenger runs* KeyGen*:*

- *The challenger gives the resulting common parameters cp to $\mathcal{B}$.*
- *The challenger gives $\mathcal{B}$ the private key shares of the corrupted decryption servers. However, the private key shares of uncorrupted decryption servers are kept secret from $\mathcal{B}$.*

Challenge: *$\mathcal{B}$ chooses two equal length plaintexts M_0, M_1 and gives them to the challenger. The challenger responds with $C^* =< U, V >= Encrypt(cp, M_{b'})$ for a random $b' \in \{0,1\}$.*

Guess: *$\mathcal{B}$ outputs a guess $b'' \in \{0,1\}$. $\mathcal{B}$ wins if $b'' = b'$.*

The adversary $\mathcal{B}$'s advantage is defined to be:

$$Adv(\mathcal{B}) = |2Pr[b'' = b'] - 1|.$$

Lemma 1. *If H_1 is a random oracle from $\{0,1\}^*$ to $\mathbb{G}_1^*$. $\mathcal{A}_2$ is an **IND-IDTH-CPA** adversary that has advantage ϵ against IB-ThKE. Suppose $\mathcal{A}_2$ makes q_E complete private key extraction queries and at most $q_{H_1}(q_{H_1} > q_E)$ hash queries to H_1. Then there is an **IND-TH-CPA** adversary $\mathcal{B}$ that has advantage at least $\epsilon' = \epsilon(q_{H_1} - q_E)/(q_{H_1})^2$ against BasicThIBE.*

Proof. $\mathcal{B}$ works by interacting with $\mathcal{A}_2$ in an IND-IDTH-CPA game as follows:

Init: $\mathcal{A}_2$ chooses a fixed set S of $t-1$ KEAs it wants to corrupt. Without loss of generality, assume $\mathcal{A}_2$ chooses $S = \{1, 2, \cdots, t-1\}$.

Setup: Algorithms $\mathcal{B}$ starts by receiving BasicThIBE's public parameters $cp = \{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_2, P, P_{pub}, Q, \{P_{pub}^{(1)}, P_{pub}^{(2)}, \ldots, P_{pub}^{(n)}\}\}$ from his challenger, and gives $\mathcal{A}_2$ the IB-ThKE system parameters $\{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, H_1, H_2, H_3, P_{pub}, \{P_1, P_2, \cdots, P_n\}\}$, where

- $q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}$ are taken from cp.
- H_1 is a random oracle controlled by $\mathcal{B}$. H_2 is described as below. H_3 is a one-way hash function but not need to be a random oracle.
- $P_i (i = 1, 2, \cdots, n)$: $\mathcal{B}$ randomly picks $r_1, r_2, \cdots, r_n \in \mathbb{Z}_q^*$. Keep r_i in secret, and return $P_i = r_i P$ to $\mathcal{A}_2$.

Then $\mathcal{B}$ issues private key share queries to his challenger. $\mathcal{B}$'s challenger returns $\{d_i\}_{i \in S}$ to $\mathcal{B}$.

H_1-queries: $\mathcal{A}_2$ can query the random oracle H_1 at any time. Assume $\mathcal{A}_2$ issues at most q_{H_1} distinct hash queries to H_1, $\mathcal{B}$ begins by choosing an index u uniformly at random with $1 \le u \le q_{H_1}$. Let ID_i be the i-th distinct identity asked by $\mathcal{A}_2$, $\mathcal{B}$ maintains a list L_1 of tuples $< ID_i, b_i, Q_{ID_i} >$, where:

- If $i \neq u$, then $\mathcal{B}$ picks b_i at random from $\mathbb{Z}_q^*$, outputs $Q_{ID_i} = H_1(ID_i) = b_i P$. Add the entry $< ID_i, b_i, Q_{ID_i} >$ to L_1.
- If $i = u$, then $\mathcal{B}$ picks b_u at random from $\mathbb{Z}_q^*$, outputs $Q_{ID_u} = H_1(ID_u) = b_u Q$. Add the entry $< ID_u, b_u, Q_{ID_u} >$ to L_1.

H_2-queries: $\mathcal{A}_2$ can issues H_2 queries at any time. $\mathcal{B}$ forwards them to $\mathcal{B}$'s challenger and returns the answers to $\mathcal{A}_2$.

KEA's private key extraction queries: To answer $\mathcal{A}_2$'s private key extraction queries upon $t-1$ corrupted KEAs, $\mathcal{B}$ returns $r_i (i \in S)$ to $\mathcal{A}_2$.

Phase1: $\mathcal{A}$ issues a number of key extraction queries on ID_i adaptively.

- Complete private key extraction queries: It's reasonable to assume that $\mathcal{A}_2$ asked for $H_1(ID_i)$ before issuing complete private key extraction queries on an identity ID_i. In order to provide complete decryption key of ID_i, $\mathcal{B}$ obtains the corresponding $< ID_i, b_i, Q_{ID_i}, >$ from L_1. If $i \neq u$ then $\mathcal{B}$ returns $d_{ID_i} = b_i P_{pub}$ as the decryption key of ID_i. Else if $i = u$, then $\mathcal{B}$ terminates and outputs "Abort".
- Partial key queries: When $\mathcal{A}_2$ queries on the partial keys of ID_i, $\mathcal{B}$ runs like this:
  - If $i \neq u$, $\mathcal{B}$ randomly chooses a polynomial of degree $t-1$ over $\mathbb{Z}_q^*$: $f_{ID_i}(x) = b_i + \sum_{l=1}^{t-1} c_l x^l, c_l \in_R \mathbb{Z}_q^*$, and defines $F_{ID_i}(j) = f_{ID_i}(j) P_{pub}$ for $1 \le j \le n$. Then $\mathcal{B}$ returns $P_{ID_i}^{(j)} = F_{ID_i}(j) + r_j P_{pub} (j \in S)$ and $V_{ID_i}^{(k)} = \hat{e}(F_{ID_i}(k), P) (1 \le k \le n)$ to $\mathcal{A}_2$.
  - Else if $i = u$. For $j \in S$, $\mathcal{B}$ returns $P_{ID_u}^{(j)} = b_u d_j + r_j P_{pub}$ and $V_{ID_u}^{(j)} = \hat{e}(b_u d_j, P)$. For $k \in \{1, 2, \cdots, n\} - S$, $\mathcal{B}$ returns $V_{ID_u}^{(k)} = \hat{e}(b_u Q, P_{pub}^{(k)})$.

  It's easy to prove that $P_{ID_i}^{(j)}, V_{ID_i}^{(k)}$ $(i \le q_{H_1}, j \in S$ and $k \in \{1, 2, \cdots, n\})$ can pass the validity test of $\mathcal{A}_2$. When $i = u$, we make use of that $\{P_{pub}^{(j)}\}_{1 \le j \le n}$, $\{d_k\}_{k \in S}$ can pass the validity test of $\mathcal{B}$ (see appendix B for more details).
- Replace KEA's public keys: Suppose the request is to replace the public key for $\Gamma_j (j \in S)$ with $P_j' = r_j' P$ after passing $< P_j', \Delta_j >$ to $\mathcal{B}$ (It should be a valid pair: $\hat{e}(P_j', P_{pub}) = \hat{e}(\Delta_j, P)$). $\mathcal{B}$ accepts $\mathcal{A}_2$'s request and computes partial keys upon ID_i as:
  - If $i \neq u$, $P_{ID_i}^{(j)} = F_{ID_i}(j) + \Delta_j$. $F_{ID_i}(j)$ is defined above.
  - If $i = u$, $P_{ID_u}^{(j)} = b_u d_j + \Delta_j$.

Challenge: Adversary $\mathcal{A}_2$ outputs two equal length plaintexts M_0, M_1 and an identity ID_{ch} which it decided to be challenged on. $\mathcal{B}$ responds as follows:

- If $ID_{ch} \neq ID_u$ then $\mathcal{B}$ terminates the game and reports "Abort".
- If $ID_{ch} = ID_u$ then $\mathcal{B}$ forwards M_0, M_1 to its challenger. When it receives the ciphertext $C' =< U^*, V^* >$, $\mathcal{B}$ returns $\mathcal{A}_2$ with the challenge $C^* =< b_u^{-1} U^*, V^* >$. Where C' is the BasicThIBE encryption of $M_{b'}$ for a random $b' \in \{0,1\}$ under the public key Q, and b_u^{-1} is the inverse of b_u mod q.

Phase2: Adversary $\mathcal{A}_2$ makes more queries. $\mathcal{B}$ responds in the same way as in phase1. Except the complete private key query on ID_u.

Guess: Eventually $\mathcal{A}_2$ outputs a guess $b'' \in \{0,1\}$. $\mathcal{B}$ outputs b'' as its guess for b'.

Analysis.

If $\mathcal{B}$ does not abort during the game(denoted this event as $\mathcal{H}$), then $\mathcal{A}_2$'s view is identical to its view in the real attack. Because $\mathcal{B}$'s responses to all hash queries are uniformly and independently distributed as in the real attack, and all responses to $\mathcal{A}_2$'s request can pass validity test unless $\neg\mathcal{H}$. Furthermore, $\hat{e}(d_Q, U^*) = \hat{e}(d_{ID_u}, b_u^{-1}U^*)$. Thus, by the definition of $\mathcal{A}_2$ we have $|2Pr(b'' = b') - 1| = Adv(\mathcal{A}_2) = \epsilon$ when $\mathcal{B}$ never aborts. The advantage of $\mathcal{B}$ is $\epsilon' \geq \epsilon \cdot Pr[\mathcal{H}]$. We name the event that $\mathcal{A}_2$ made a complete private key extraction query on ID_u at some point as $\mathcal{E}_1$ and the event that $\mathcal{A}_2$ chose $ID_{ch} \neq ID_u$ as $\mathcal{E}_2$. Then $Pr[\mathcal{H}] = Pr[\neg\mathcal{E}_1 \bigwedge \neg\mathcal{E}_2] = Pr[\neg\mathcal{E}_1]Pr[\neg\mathcal{E}_2|\neg\mathcal{E}_1] = \frac{1}{q_{H_1}}Pr[\neg\mathcal{E}_1] \geq \frac{1}{q_{H_1}} \cdot \frac{q_{H_1}-q_E}{q_{H_1}}$. So $\epsilon' \geq \frac{\epsilon(q_{H_1}-q_E)}{(q_{H_1})^2}$. This finishes the proof. □

Lemma 2. *Let H_2 be a random oracle from $\mathbb{G}_2$ to $\{0,1\}^l$, and let $\mathcal{B}$ be an **IND-TH-CPA** adversary that has advantage ϵ' against `BasicThIBE`. Suppose $\mathcal{B}$ makes q_{H_2} distinct hash queries to H_2. Then there is an algorithm $\mathcal{C}$ that solves the BDH problem with advantage at least $\epsilon'' = \epsilon'/q_{H_2}$.*

Proof. Algorithm $\mathcal{C}$ is given a random instance $< P, aP, bP, cP >$ of the BDH problem where a, b, c are random in Z_q^*. To compute $D = \hat{e}(P,P)^{abc}$, $\mathcal{C}$ runs $\mathcal{B}$ as follows:

Init: The adversary $\mathcal{B}$ chooses a set S of $t-1$ decryption servers it wants to corrupt. Without loss of generality, assume $\mathcal{B}$ chooses $S = \{1, 2, \cdots, t-1\}$.

KeyGen: Algorithm $\mathcal{C}$ starts by giving $\mathcal{B}$ the `BasicThIBE` system parameters $\{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{Pub}, Q, H_2, \{P_{pub}^{(1)}, P_{pub}^{(2)} \cdots, P_{pub}^{(n)}\}\}$. Here

- $q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P$ are taken from `BasicThIBE`'s public parameters.
- $P_{pub} = cP$, $Q = bP$.
- $P_{pub}^{(i)} (i = 1, 2, \cdots, n)$: pick randomly values c_1, c_2, $\cdots$, $c_{t-1} \in \mathbb{Z}_q^*$, find the appropriate $\lambda_{ij}^{S'}$ coefficients. Then $\mathcal{C}$ computes $P_{pub}^{(i)} = \lambda_{i0}^{S'} P_{pub} + \sum_{j=1}^{t-1} \lambda_{ij}^{S'} c_j P$ $(i = t, t+1, \cdots, n)$, and $P_{pub}^{(j)} = c_j P$ $(j = 1, 2, \cdots, t-1)$. Where $S' = \{0\} \cup S$ and $\lambda_{ij}^{S'}$ denotes a Lagrange coefficient with respect to the set S'.
- H_2 is a random oracle controlled by $\mathcal{C}$.

Private key shares extraction queries: $\mathcal{B}$ issues private key shares extraction query on Q. In order to provide $t-1$ valid secret key shares upon Q, $\mathcal{C}$ returns $d_j = c_j Q$ for $j \in S$.

H_2-queries: $\mathcal{B}$ may issues queries to the random oracle H_2 at any time. In order to simulate H_2, $\mathcal{C}$ maintains an initially empty list L_2 of tuples $< x_i, R_i >$ as below:

- If the query x_i already exists on the L_2 in a tuple $< x_i, R_i >$ then $\mathcal{C}$ responds with $H_2(x_i) = R_i$.
- Otherwise, $\mathcal{C}$ picks a random string $R_i \in \{0,1\}^l$ and adds the tuple $< x_i, R_i >$ to L_2. Respond to $\mathcal{B}$ with $H_2(x_i) = R_i$.

Challenge: $\mathcal{B}$ outputs two equal length plaintexts M_0, M_1. $\mathcal{C}$ chooses a random string $R \in \{0,1\}^l$ and returns $C^* =< U, V >=< aP, R >$ as the challenge ciphertext.

Guess: $\mathcal{B}$ outputs a guess $b'' \in \{0,1\}$. $\mathcal{C}$ ignores b'' and picks a tuple $< x_i, R_i >$ from L_2 list randomly. Then output x_i as the solution to the given instance of the BDH problem.

Analysis.

Let $\mathcal{E}$ be the event that $\mathcal{B}$ issues a query for $H_2(\hat{e}(P,P)^{abc})$ during the simulation above, and $\mathcal{F}$ be the event that $\mathcal{C}$'s outputs is equal to $\hat{e}(P,P)^{abc}$. As explained in [5], if $\mathcal{B}$ never issues a query for $H_2(\hat{e}(P,P)^{abc})$, then the decryption of C^* is uniform in $\mathcal{B}$'s view. That means $Pr[b'' = b'|\neg\mathcal{E}] = 1/2$ in the real attack. We have $Pr[b'' = b'] = Pr[b'' = b'|\neg\mathcal{E}]Pr[\neg\mathcal{E}] + Pr[b'' = b'|\mathcal{E}]Pr[\mathcal{E}] \leq Pr[b'' = b'|\neg\mathcal{E}]Pr[\neg\mathcal{E}] + Pr[\mathcal{E}] = \frac{1}{2} + \frac{1}{2}Pr[\mathcal{E}]$, and $Pr[b'' = b'] \geq Pr[b'' = b'|\neg\mathcal{E}]Pr[\neg\mathcal{E}] = \frac{1}{2} - \frac{1}{2}Pr[\mathcal{E}]$. It follows that in the real attack $\epsilon' = |2Pr[b'' = b'] - 1| \leq Pr[\mathcal{E}]$, where ϵ' is $\mathcal{B}$'s advantage. Then $Pr[\mathcal{F}|\mathcal{E}] = 1/q_{H_2}$, and $Pr[\mathcal{F}|\neg\mathcal{E}] = 1/q$ which is negligible when q is large enough. From $Pr[\mathcal{F}] = Pr[\mathcal{F}|\mathcal{E}]Pr[\mathcal{E}] + Pr[\mathcal{F}|\neg\mathcal{E}]Pr[\neg\mathcal{E}] \geq Pr[\mathcal{F}|\mathcal{E}]Pr[\mathcal{E}]$, $\mathcal{C}$'s probability to solve the instance of BDH problem is at least ϵ'/q_{H_2}. □

Thus, putting all the bounds that have been obtained above, it shows that a type2 IND-IDTH-CPA attacker on `IB-ThKE` scheme with advantage ϵ can be used as a subroutine to construct a BDH-attacker for a given instance of BDH problem with the advantage at least $\epsilon'' = \epsilon(q_{H_1} - q_E)/(q_{H_1})^2 q_{H_2}$. This finishes the proof of Theorem 1. □

5 Further Discussion

5.1 Security Improvements

A stronger attack against `IB-ThKE` is so called chosen-ciphertext attacks (CCA2 [3]), in which $\mathcal{A}_2$ is given a full access to the decryption oracle that runs the *KEA's Sub-Decryption* algorithm to generate decryption shares of the uncorrupted KEAs. However, it's very difficult to design an efficient CCA2 secure ID-based threshold cryptosystem [7, 16].

One possible solution is to introduce publicly checkable encryption [12]. Then every KEA can check the validity of the ciphertext before generating the decryption shares of it. A potential way is adding one additional redundant element to the ciphertext. Like in the construction from [1], using a "validity check" hash function

$H_4 : \mathbb{G}_1 \times \{0,1\}^l \rightarrow \mathbb{G}_1^*$ to convert the ciphertext into $< U, V, W >$, where U, V has the same definitions as in section 3.2 and $W = rH_4(U, V)$. Then the check could be carried out using bilinear pairings. However, since each KEA should check the ciphertext's validity, the additional pairing operations cast a heave burden to each KEA.

An alternative approach is applying the techniques proposed in [16]. That is using non-interactive zero knowledge proof to make the ciphertext publicly checkable without pairing operation. The drawback is the length of the ciphertext will be increased. Another way to solve the problem is to encode the the information necessary for the validity-check into the original chosen-plaintext secure ciphertext, or encode the consistency information in ciphertext element containing the receiver's identity. Till now, this approach is only applied in key encapsulation mechanism but not in any fully-fledged identity-based encryption schemes [10]. How to design an efficient and provably secure publicly checkable encryption scheme will be our future work.

5.2 Practical Extensions

The `IB-ThKE` can readily be changed to a chosen-plaintext secure identity-based threshold decryption scheme. There are only type2 adversaries against such a scheme, and its security reduction is similar to `IB-ThKE`.

Another application of IB-ThKE is the ID-based mediated cryptosystem secure against outside attacks [6, 11], by setting $(t, n) = (2, 2)$.

6 Conclusions

In this paper, we propose an identity-based threshold key escrow scheme `IB-ThKE` to efficiently solve the conflicting between the authorized key escrow and the user's privacy in the identity-based cryptosystem. `IB-ThKE` is proved to be chosen-plaintext secure in the appropriate model, assuming the Bilinear Diffie-Hellman problem is hard. Additionally, we discuss the difficulty in constructing a fully secure scheme, and propose the potential applications of `IB-ThKE`.

Acknowledgements

We would like to thank the anonymous referees for helpful comments. This work is supported by NSFC under the grants 60673077, 60473020, 60573030.

References

[1] J. Baek, and Y. Zheng, "Identity-based threshold decryption," *Proceedings of PKC '04*, LNCS 2947, pp. 262-276, Springer-Verlag, 2004.

[2] M. Bellare, and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," *Proceedings of the First ACM Conference on Computer and Communications Security*, pp. 62-73, 1993.

[3] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Crypto '98*, LNCS 1462, pp. 26-45, Springer-Verlag, 1998.

[4] M. Blum, A. De santis, S. Micali, and G. Persiano, "Non-interactive zero knowledge," *SIAM Journal of Computers*, vol. 6, no. 4, pp. 1084-1118, 1991.

[5] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing," *Advances in Cryptology-Crypto '01*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[6] D. Boneh, X. Ding, and G. Tsudic, "Identity based encryption using mediated RSA," *Proceeding of the 3th Workshop on Information Security Application*, 2002.

[7] Z. C. Chai, Z. F. Cao, and R. X. LU, "ID-based threshold decryption without random oracles and its application in key escrow," *Inforsec 2004*, pp. 119-124, ACM press, 2004.

[8] D. E. Denning, and M. Smid, "Key escrowing today," *IEEE Communications Magazine*, pp. 58-68, 1994.

[9] P. Fouque, and D. Pointcheval, "Threshold cryptosystems secure against chosen-ciphertext attacks," *Asiacrypt '01*, LNCS 2248, pp. 351-368, Springer-Verlag, 2001.

[10] E. Kiltz, "Chosen-ciphertext secure identity-based encryption in the standard model with short ciphertexts," *Cryptology ePrint Archive*, Report 2006/122, 2006. (http://eprint.iacr.org/)

[11] B. Libert, and J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," *PODC '03*, ACM press, pp. 163-171, 2003.

[12] C. Lim, and P. Lee, "Another method for attaining security against adaptively chosen ciphertext attack," *CRYPTO '93*, LNCS 773, pp. 410-434, Springer-Verlag, 1993.

[13] S. S. A. Riyami, K. G. Paterson, "Certificateless public key cryptography," *Asiacrypt '03*, LNCS 2894, pp. 452-473, Springer-Verlag, 2003.

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[15] Y. Shi, J. Li, J. Pan, and J. Shi, "Efficient certificateless public key encryption with pairing," *Proceeding of the The IASTED International Conference on Networks and Communication Systems*, ACTA Press, 2006.

[16] V. Shoup, and R. Gennaro, "Securing threshold cryptosystems against chosen ciphertext attack," *Advances in Cryptology-Eurocrypt '98*, LNCS 1403, Springer, pp. 1-16, 1998.

Yu Long received the B.S. degree in computer science and technology in South West Jiao Tong University, China, and received the M.S. degree in information security engineering department, Shanghai Jiao Tong University, China, and now she is a Doctor Candidate in the same school. Her research interests include information theory and modern cryptography.

Zheng Gong received the B.S. degree in computer science and technology in NanChang University, China, and received the M.S. degree in computer science and technology in South China University of Technology, China. Now he is a Doctor Candidate in Shanghai Jiaotong University, China. His recent research directions are cryptography and provable security.

Ke-fei Chen received his Ph.D degree in Justus Liebig University Giessen, Germany, 1994. His main research areas are classical and modern cryptography, theory and technology of network security, etc. Since 1996, he came to Shanghai Jiao Tong University and become the Professor at the Department of Computer Science and Engineering. Up to now (1996-2005), more than 80 academic papers on cryptology and information security have been published in Journals.

Shengli Liu received her first Ph.D degree in Xidian University in 2000, and received her second Ph.D degree in Eindhoven University of Technology, Holland, 2002. Her research areas are information theory, computer security, and classical cryptography, etc. Since 2002, she came to Shanghai Jiao Tong University and became the adjunct professor at the Department of Computer Science and Engineering.