INFORMATICA, 2006, Vol. 17, No. 4, 519–534

Adaptive Chosen Ciphertext Secure Threshold Key Escrow Scheme from Pairing [*]

Yu LONG, Kefei CHEN

Department of Computer Science and Engineering, Shanghai Jiao Tong University &
National Laboratory of Modern Communications
800 Dongchuan Road, Shanghai 200240, P. R. China
e-mail: longyu@sjtu.edu.cn

Shengli LIU

Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240, P. R. China

Received: January 2006

Abstract. This paper proposes a threshold key escrow scheme from pairing. It tolerates the passive adversary to access any internal data of corrupted key escrow agents and the active adversary that can make corrupted servers to deviate from the protocol. The scheme is secure against threshold adaptive chosen-ciphertext attack. The formal proof of security is presented in the random oracle model, assuming the decision Bilinear Diffie-Hellman problem is computationally hard.

Key words: threshold key escrow, identity-based cryptography, chosen-ciphertext attack, pairing based cryptology.

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 crimes. This has led to the concept of key escrow (Denning, 1994).

In 2001, D. Boneh and M. Franklin proposed a practical Identity-Based Encryption (IBE) (Boneh, 2001) 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. S.S. Al-Riyami and K.G. Paterson introduced

[*] This paper is supported by NSFC under the grants 60303026, 60473020, 60573030 and the Foundation of NLMC under the grant 51436040405JW0304.

the concept of Certificateless Public Key Cryptography (CL-PKC) (Sl-Riyami, 2003) to solve the problem of IB-PKC. In CL-PKC, the private key of every entity is created by the PKG and the entity unitedly. However, the law enforcement agency (LEA) is unable to monitor communications in such a scheme. Another way to solve the problem is to share the power of monitor among a set of key escrow agents (KEAs). A trusted key management center serves as the PKG, to generate private keys for KEAs, while it is given no access to any ciphertext. To monitor the communications of some entity needs at least a threshold value KEAs' co-operation.

Our contribution is to propose such a scheme from ID-based cryptosystems, named ID-based threshold key escrow (`IB-ThKE`) scheme. This scheme is provably secure in threshold adaptive chosen-ciphertext attack model, assuming the decision Bilinear Diffie–Hellman problem is computationally hard.

Other Related Works. In 1989, Y. Desmedt and Y. Frankel presented the first practical threshold scheme (Desmedt, 1989). Afterwards, a lot of threshold cryptosystems are proposed. The threshold decryption was formalized by V.Shoup and R.Gennaro (Shoup, 1998). Many schemes (Libert, 2003; Chai, 2004; Baek, 2004) have been put forward subsequently in the context of ID-based cryptography. However, schemes in (Libert, 2003; Chai, 2004) were only semantically secure, and the scheme in (Baek, 2004) did not analyze the probability of the event that the adversary obtains the decryptions of ill-formed but still *valid-look* ciphertexts. What's more, all of these schemes could not tolerate active adversary that can modify the public verification keys of corrupted servers.

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. Assuming 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 $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = e(P, Q)^{ab}$.
- 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. *Decision Bilinear Diffie–Hellman (BDH) Assumption*

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 $\mathbb{Z}_q^*$. Randomly choose $D \in \mathbb{G}_2$.

Decision Bilinear Diffie–Hellman Problem
Given (P, aP, bP, cP, D), determine whether $D = \hat{e}(P, P)^{abc}$ or not.

An algorithm $\mathcal{A}$ that outputs $b' \in \{0,1\}$ has an advantage ϵ in solving the decision BDH problem in $< \mathbb{G}_1, \mathbb{G}_2, \hat{e} >$ if $|Pr[\mathcal{A}(P, aP, bP, cP, \hat{e}(P,P)^{abc}) = 1] - Pr[\mathcal{A}(P, aP, bP, cP, D) = 1]| > \epsilon$, where D is randomly chosen from $\mathbb{G}_2$. In general, the decision BDH problem is believed to be hard in $< \mathbb{G}_1, \mathbb{G}_2, \hat{e} >$ (Waters, 2005). That means there is no probabilistic algorithm that can solve the decision BDH problem with a non-negligible advantage ϵ within polynomial time.

2.3. *Threshold Security*

The idea of (t, n) threshold secret sharing was proposed in (Shamir, 1979). The formal security model of threshold cryptosystems has been discussed in (Shoup, 1998; Fouque, 2001). In threshold setting, the adversary first corrupts $t-1$ out of n decryption servers and obtains secret key shares held by them. During the course of the chosen-ciphertext attack, the adversary can submit ciphertexts to the uncorrupted decryption servers. So in the threshold chosen-ciphertext attack (IND-TH-CCA) (Bellare, 1998) the adversary sees both the decryptions of chosen ciphertexts and the decryption shares of these ciphertexts. This extra information makes it very difficult to construct an IND-TH-CCA secure threshold cryptosystem.

The paper (Shoup, 1998) proposed two secure threshold cryptosystems against chosen ciphertext attack. In this work, the non-interactive zero knowledge proof of membership was used to make the ciphertext publicly checkable (Blum, 1991; Lim, 1993). Motivated by (Shoup, 1998), we present a threshold key escrow scheme `IB-ThKE`, and prove its security in the sense of threshold adaptive chosen ciphertext attack in the random oracle model (Bellare, 1993).

2.4. *Non-Interactive Proof of Membership*

Similar to (Baek, 2004), a non-interactive zero knowledge proof of membership system named `Proof-Log` can be constructed for the language $L = \{(v, \tilde{v}) \in \mathbb{G}_2 \times \mathbb{G}_2 | \log_g v = \log_{\tilde{g}} \tilde{v}\}$, where $g = \hat{e}(P,P)$ and $\tilde{g} = \hat{e}(P, \tilde{P})$. P and $\tilde{P}$ are generators of $\mathbb{G}_1$. $\mathbb{G}_1, \mathbb{G}_2, \hat{e}$ have the same definitions as in section 2.1.

Given $(P, \tilde{P}, g, \tilde{g})$, a one-way hash function H_5: $\mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$ and $(k, \tilde{k}) \in L$, the Prover wants to convince the Verifier that he indeed knows a secret $S = (\log_g k)P = (\log_{\tilde{g}} \tilde{k})P \in \mathbb{G}_1$ without yielding any "knowledge" of S. The proof system works like this:

- The Prover randomly chooses $T \in \mathbb{G}_1^*$, then computes $\gamma = \hat{e}(T, P)$, $\tilde{\gamma} = \hat{e}(T, \tilde{P})$, $h = H_5(\hat{e}(P, S), \hat{e}(\tilde{P}, S), \hat{e}(T, P), \hat{e}(T, \tilde{P}))$ and $L = T + hS \in \mathbb{G}_1$. Send $\{\gamma, \tilde{\gamma}, L\}$ to the Verifier.
- The Verifier computes $h = H_5(k, \tilde{k}, \gamma, \tilde{\gamma})$ and checks whether $\hat{e}(L, P) = \gamma \cdot k^h$ and $\hat{e}(L, \tilde{P}) = \tilde{\gamma} \cdot \tilde{k}^h$. If both equations hold, then the Verifier returns "*Accept*", else returns "$Reject$".

It's easy to prove that $(k, \tilde{k}) \in L$ if and only if there is an element $S \in \mathbb{G}_1^*$ such that $k = \hat{e}(S, P)$ and $\tilde{k} = \hat{e}(S, \tilde{P})$, and the properties of this protocol can be discussed as in (Baek, 2004).

3. (t, n) Threshold Key Escrow Scheme from Pairing

In the (t, n) threshold key escrow scheme IB-ThKE from pairing, the system is consisted of a trusted authority called the Key Management Center (KMC), a Law Enforcement Agent (LEA) with n Key Escrow Agents (KEAs), and many communication users.

The plaintext M that is encrypted under an identity is recoverable from at least t of n KEAs. We assume that the KMC has no access to any ciphertext, since KMC knows the private key of every user.

3.1. *Defining IB-ThKE*

First, we sketch the characteristics of IB-ThKE.

Similar to IB-PKC, all the users participating in this scheme are connected to KMC by secret channels. The secret key of each user is issued by KMC, and the public key is the unambiguous identity of the user, such as the email address or a telephone number. On the one hand, the prospective of every communication user is identical to the traditional ID-based cryptosystem. On the other hand, to monitor the communication that is encrypted under an identity and a plaintext M, the LEA needs at least t of n KEAs' co-operation.

Every KEA has a private key chosen by himself. And the corresponding public verification key is given to the KMC. When the LEA wants to decrypt a received ciphertext of an user $Alice$, KMC returns the partial secret keys and the public verification keys of $Alice$ to at least t KEAs. Then every KEA can generate a decryption share of this ciphertext, taking as input the ciphertext and the partial secret key and his private key, after checking the validity of partial secret key. These shares are sent to the LEA, who starts checking the validity of every share. If more than t shares are valid, the LEA combines them to obtain the plaintext.

Additionally, each KEA can update his private key. The KMC accepts the KEA's request after verifying its validity, then transmits the new partial secret key to this KEA. This character is attractive in designing a dynamic threshold key escrow scheme. It will be discussed in latter section.

3.2. *Description of our Scheme*

The IB-ThKE consists of the following polynomial-time algorithms.

Setup(k_0): run by KMC and n KEAs Γ_i $(i = 1, 2, ..., n)$.

- Given a security parameter k_0, the KMC outputs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order $q(> 2^{k_0})$, 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 master key $s \in \mathbb{Z}_q^*$. Compute $P_{pub} = sP$ and choose five hash functions H_1: $\{0,1\}^* \to \mathbb{G}_1$, H_2: $\mathbb{G}_2 \to \{0,1\}^l$, H_3: $\mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_1^*$, H_4: $\mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_q^*$ and H_5: $\mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \times \mathbb{G}_2 \to \mathbb{Z}_q^*$. Note that H_1, H_2, H_3, H_4 are viewed as random oracles (Bellare, 1993) in the security analysis.
- Key escrow agent Γ_i $(i = 1, 2, ..., n)$ randomly selects $s_i \in \mathbb{Z}_q^*$, and computes $P_i = s_i P$.

The system public parameters are

$$cp = \big\{q, l, \mathbb{G}_1, \mathbb{G}_2, P, \hat{e}, H_1, H_2, H_3, H_4, H_5, P_{pub}, \{P_1, P_2, ..., P_n\}\big\}.$$

KeyGen1(ID, s, cp): given an user's identity ID, the KMC returns $d_{ID} = sH_1(ID)$ through a secret channel to this user as his complete decryption key.

KeyGen2($ID, s, \{P_i\}_{(i=1,2,...,n)}, cp$): given an user's identity ID and an authorized request for monitoring this user's communication, the KMC chooses a polynomial of degree $t-1$ over $\mathbb{Z}_q^*$:

$$f(x) = s + a_1x + \cdots + a_{t-1}x^{t-1}.$$

For $i = 1, 2, ..., n$, it computes $S_{ID}^{(i)} = f(i)Q_{ID} + sP_i$, $V_{ID}^{(i)} = \hat{e}(f(i)Q_{ID}, P)$. $S_{ID}^{(i)}$ are returned to Γ_i secretly as his partial secret key, and $V_{ID}^{(i)}$ are published.

Encryption(M, ID, cp): To encrypt a message $M \in \{0,1\}^l$ under the receiver's identity ID, the sender chooses $r, t \in \mathbb{Z}_q^*$ uniformly at random and computes $Q_{ID} = H_1(ID)$. Then set the ciphertext to be $(V, U, \bar{U}, e, f)$, where $V = M \oplus H_2(\hat{e}(Q_{ID}, P_{pub})^r), U = rP, W = tP, \bar{P} = H_3(U, V, W), \bar{U} = r\bar{P}, \bar{W} = t\bar{P}, e = H_4(\bar{P}, \bar{U}, \bar{W})$ and $f = t + er$.

User's Decryption(C, d_{ID}, cp): Let $C = (V, U, \bar{U}, e, f)$ be a ciphertext encrypted under an identity ID. To decrypt C with the the corresponding private key d_{ID}, the receiver runs as

- (*Ciphertext validity verification*) Check if $e = H_4(\bar{P}, \bar{U}, \bar{W})$, where $W = fP - eU$, $\bar{P} = H_3(U, V, W)$ and $\bar{W} = f\bar{P} - e\bar{U}$. If C can not pass this test, output "$Invalid \setminus Ciphertext$".
- Else compute $M = V \oplus H_2(\hat{e}(d_{ID}, U))$.

KEA's Sub-Decryption($C, \{S_{ID}^{(i)}, V_{ID}^{(i)}\}_{(i=1,2,...,n)}, cp$): Given a ciphertext $C = (V, U, \bar{U}, e, f)$ and a key pair $(S_{ID}^{(i)}, V_{ID}^{(i)})$ $(i = 1, 2, ..., n)$, the KEA Γ_i checks the validity of $(S_{ID}^{(i)}, V_{ID}^{(i)})$ and computes his decryption share as follows:

- (*Key share verification*) First, Γ_i checks the validity of $(S_{ID}^{(i)}, V_{ID}^{(i)})$ with $\hat{e}(S_{ID}^{(i)}, P) = V_{ID}^{(i)} \cdot \hat{e}(P_i, P_{pub})$. And everybody can check if $\prod_{i\in T}(V_{ID}^{(i)})^{L_i^T} = \hat{e}(Q_{ID}, P_{pub})$ for any subset $T \subset \{1, 2, ..., n\}$ such that $|T| = t$, where L_i^T denotes the appropriate Lagrange coefficient with respect to the set T, $L_i^T = \prod_{j\in T, j\neq i}\frac{j}{j-i}(\text{mod } q)$. If $(S_{ID}^{(i)}, V_{ID}^{(i)})$ can not pass this test, Γ_i outputs (i, "$Invalid \setminus KeyShare$").
- Else Γ_i checks the validity of the ciphertext as in the *User's Decryption*. If it does not hold, output (i, "$Invalid \setminus Ciphertext$").
- Otherwise, both tests succeed. Compute $k_{ID}^i = \hat{e}(S_{ID}^{(i)} - s_i(P_{pub}), U)$, $R_i = \hat{e}(T_i, P)$, $\tilde{R}_i = \hat{e}(T_i, U)$, $h_i = H_5(V_{ID}^{(i)}, k_{ID}^i, R_i, \tilde{R}_i)$, $\lambda_i = T_i + h_i(S_{ID}^{(i)} - s_i(P_{pub}))$ for random $T_i \in \mathbb{G}_1^*$. Then output the decryption share $\delta_{ID,C}^i = \{i, k_{ID}^i, h_i, \lambda_i\}$.

Monitoring($C, \{\delta_{ID,C}^i\}_{i\in T', |T'|>=t}, cp$): given a ciphertext $C = (V, U, \bar{U}, e, f)$ and a set of decryption shares $\{\delta_{ID,C}^i\}_{i\in T'}$, the law enforcement agent (LEA) runs as follows:

- (*Decryption share verification*) For $i \in T'$, check if $h_i = H_5(V_{ID}^{(i)}, k_{ID}^i, R_i, \tilde{R_i})$, where $R_i = \hat{e}(\lambda_i, P)/(V_{ID}^{(i)})^{h_i}$ and $\tilde{R_i} = \hat{e}(\lambda_i, U)/(k_{ID}^i)^{h_i}$. If it fails, discard the decryption share and return (i, "$Invalid\ KEA$").
- Else if the LEA collects t valid decryption shares from Γ_i ($i \in T, T \subseteq T'$, $|T| = t$), it 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 LEA.

KEA's public key updating(P_i', Δ_i, ID): in this scheme, we allow KEA Γ_i ($i \in \{1, 2, ..., n\}$) 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 transmit $< i, P_i', \Delta_i >$ to the KMC secretly.
- The KMC checks the validity of P_i' by $\hat{e}(P_i', P_{pub}) = \hat{e}(\Delta_i, P)$. If it holds, KMC changes P_i to P_i' publicly and renews $S_{ID}^{(i)}$ in *KeyGen*2 accordingly. Else KMC refuses Γ_i's request.

Note that each KEA uses the non-interactive zero knowledge protocol *Proof-Log* to make its decryption share checkable.

4. Security Analysis

4.1. *Adversary Types*

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

IB-ThKE Type1 Adversary. The general adversary $\mathcal{A}_1$ against the underlying identity based cryptosystem is called the *type*1 adversary. $\mathcal{A}_1$ operates in several phases which were formally defined in (Boneh, 2001). $\mathcal{A}_1$ can adaptively issue hash queries, decryption queries and complete decryption key extraction queries.

IB-ThKE Type2 Adversary. The adversary that can corrupt KEAs is called the *type*2 adversary. 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 $\mathcal{A}_2$. Assume $\{\Gamma_i\}_{i \in S}(|S| = t - 1)$ be the set of corrupted KEAs. $\mathcal{A}_2$ can learn the secret information of corrupted KEAs, get all broadcasting messages and decryption shares of uncorrupted KEAs. Furthermore, the $\mathcal{A}_2$ can make the corrupted KEAs to deviate from the protocol in an unrestricted fashion. The actions that $\mathcal{A}_2$ against the IB-ThKE are listed below:

- *KEA's private key and partial secret key extraction queries*. For $i \in S$, $\mathcal{A}_2$ is allowed to make request for Γ_i's private key s_i, and $\mathcal{A}_2$ can ask for partial decryption key $S_{ID}^{(i)}$ for a given identity ID.
- *Complete decryption key extraction queries*. $\mathcal{A}_2$ is allowed to query on an identity ID's complete decryption key. However, it is not reasonable for $\mathcal{A}_2$ to extract the complete decryption key of the selected challenge identity ID_{ch}.
- *Decryption queries*. $\mathcal{A}_2$ is allowed to query on chosen ciphertexts, to get the plaintexts and decryption shares from uncorrupted KEAs.

- *Update KEA's public key*. Since Γ_i's public key $P_i = s_i P$ $(i = 1, 2, \ldots, n)$ is not associated with Γ_i's identity, $\mathcal{A}_2$ can choose $s'_i \in \mathbb{Z}_q^*$ and try to replace P_i by $P'_i = s'_i P$ for the corrupted KEAs.

4.2. *Security Model for* `IB-ThKE`

In this section, we give the formal security definition of `IB-ThKE` scheme. There are two distinct ways to define the threshold adaptive chosen ciphertext attacks against the `IB-ThKE` scheme, depending on whether the adversary acts as $\mathcal{A}_2$ or as $\mathcal{A}_1$. Explicitly, if `IB-ThKE` can resist against $\mathcal{A}_2$, it is secure against $\mathcal{A}_1$. So we only discuss the former.

DEFINITION 4.1 (IND-IDTH-CCA). The (t, n) threshold key escrow scheme from ID-based cryptosystem is secure against adaptive chosen ciphertext attacks (denoted by IND-IDTH-CCA) 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 public parameters to $\mathcal{A}_2$, including the public key P_i of Γ_i $(i = 1, 2, ..., n)$.

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 $\mathcal{A}_2$.

Key extraction queries1. On an identity ID, $\mathcal{A}_2$ performs a number of queries adaptively:

- *Complete decryption key extraction queries*. The challenger generates complete decryption key d_{ID}. Send it to $\mathcal{A}_2$.
- *Verification key and the corrupted KEA's private key share queries*. The challenger returns $V_{ID}^{(i)}$ for $i \in \{1, 2, ..., n\}$ and $S_{ID}^{(j)}$ for $j \in S$.
- *Update KEA's public key*. For $i \in S$, suppose the request is to update the public key of Γ_i with $< P'_i = s'_i P, \Delta_i = s'_i P_{pub} >$. After receiving $< ID, P'_i, \Delta_i >$, the challenger accepts $\mathcal{A}_2$'s request, and returns ${S_{ID}^{(i)}}'$ associated with P'_i and ID.

Decryption queries1. $\mathcal{A}_2$ arbitrarily feeds the challenger ciphertexts, then obtains plaintexts and decryption shares of uncorrupted KEAs.

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 $\mathcal{A}_2$ has made a complete decryption key extraction query, during the key extraction queries1. The challenger picks a bit $b' \in \{0, 1\}$ uniformly and sets the challenge ciphertext to be $C^* = Encryption(M_{b'}, ID_{ch}, cp)$. Return C^* to $\mathcal{A}_2$.

Key extraction queries2. $\mathcal{A}_2$ issues more key extraction queries as in key extraction queries1, except the complete decryption key of ID_{ch}.

Decryption queries2. $\mathcal{A}_2$ continues to interact with the challenger by feeding it with ciphertexts $C \neq C^*$.

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

Such an adversary $\mathcal{A}_2$ is called an IND-IDTH-CCA adversary. $\mathcal{A}_2$'s advantage is defined to be

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

Theorem 4.1. *Let H_1, H_2, H_3, H_4 be random oracles. Then* `IB-ThKE` *is an IND-IDTH-CCA secure scheme assuming the decision BDH problem is hard in groups generated by* Setup. *Concretely, suppose there is a type2 adversary $\mathcal{A}_2$ that has an advantages ϵ against the* `IB-ThKE`*. Suppose $\mathcal{A}_2$ makes at most q_E complete decryption key extraction queries. Then there is an algorithm $\mathcal{C}$ that solves the decision BDH problem in groups generated by* Setup *with the advantage at least $\epsilon'' = \epsilon/2e(1 + q_E)$.*

Proof. The proof is by reduction. First, we define a related non-identity based threshold scheme called `BasicThIBE`, and show how the adaptive chosen ciphertext attack of `IB-ThKE` can be reduced to the chosen plaintext attack of `BasicThIBE`, in the random oracle model. Then we prove `BasicThIBE`'s semantic security.

`BasicThIBE` works as follows:

KeyGen(k_0): Given a security parameter k_0, a trusted third party $\mathcal{T}$ chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order $q > 2^{k_0}$, 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 $\mathcal{T}$ 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 $i = 1, 2, ..., n$, it computes $P_{pub}^{(i)} = f(i)P \in \mathbb{G}_1$ and chooses one cryptographic hash function H_2: $\mathbb{G}_2 \to \{0, 1\}^l$. Randomly pick $Q \in \mathbb{G}_1^*$. The public parameters are

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

For $i = 1, 2, \cdots, n$, $\mathcal{T}$ 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 $T \subset \{1, 2, ..., n\}$, $|T| = t$, and L_i^T is the Lagrange coefficient with respect to the set T. If the validity test fails, he complains to $\mathcal{T}$ that issues a new share.

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

Decrypt(C, d_i): when receiving (U, V), decryption server i computes his decryption share $\delta_i = \hat{e}(U, d_i)$ and gives it to a special server called the *combiner*.

Recombine($\{\delta_i\}_{i\in T,|T|=t}$): the *combiner* selects a set $T \subset \{1, 2, \cdots, n\}$ of t 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 can be viewed as a general public key cryptosystem. The key pair is $(d_q, P_Q) = (sQ, Q)$.

DEFINITION 4.2 (IND-TH-CPA). A non-identity 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 public parameters to $\mathcal{B}$.
- The challenger gives $\mathcal{B}$ the private key shares $\{d_i\}_{i \in S}$ of the corrupted decryption servers. However, the private key shares of the 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 $C^* = (U, V) = \mathit{Encrypt}(M_{b'}, cp)$ for a random $b' \in \{0,1\}$.

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

$\mathcal{B}$ is called an IND-TH-CPA adversary (Libert, 2003). $\mathcal{B}$'s advantage is defined to be

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

Lemma 4.1. *If H_1, H_2, H_3, H_4 are random oracles. Let $\mathcal{A}_2$ be an IND-IDTH-CCA adversary that has an advantage ϵ against* `IB-ThKE`*. Suppose $\mathcal{A}_2$ makes q_E complete 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 an advantage at least $\epsilon' = \epsilon/e(1 + q_E)$ against the* `BasicThIBE`*.*

Proof. $\mathcal{B}$ works by interacting with $\mathcal{A}_2$ in an IND-IDTH-CCA 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 $S = \{1, 2, ..., t-1\}$.

Setup. Algorithms $\mathcal{B}$ starts by receiving the `BasicThIBE`'s public parameters $cp = \{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_2, P, \{P_{pub}^{(1)}, P_{pub}^{(2)}, \ldots, P_{pub}^{(n)}\}, P_{pub}, Q\}$ from his challenger, and gives $\mathcal{A}_2$ the `IB-ThKE` system parameters $\{q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, H_1, H_2, H_3, H_4, H_5, P_{pub}, \{P_1, P_2, ..., P_n\}\}$, where

- $q, l, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}$ are taken from cp;
- H_1, H_2, H_3, H_4 are random oracles controlled by $\mathcal{B}$. H_5 is a one-way hash function;
- randomly pick $m_1, m_2, ..., m_n \in \mathbb{Z}_q^*$. Keep m_i in secret, return $P_i = m_i P$ to $\mathcal{A}_2$.

Then $\mathcal{B}$ issues partial key 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 H_1 at any time. Let ID_i be the i-th distinct identity asked by $\mathcal{A}$, $\mathcal{B}$ flips a $coin_i(coin_i \in \{0,1\}, Pr[coin_i = 0] = \delta)$ and maintains a list L_1 of tuples $< coin_i, ID_i, b_i, Q_{ID_i} >$, where:

- if $coin_i = 0$, then $\mathcal{B}$ picks b_i at random from $\mathbb{Z}_q^*$. Output $Q_{ID_i} = H_1(ID_i) = b_i P$, add the tuple $< coin_i = 0, ID_i, b_i, Q_{ID_i} >$ to L_1;

- if $coin_i = 1$, then $\mathcal{B}$ picks b_i at random from $\mathbb{Z}_q^*$. Output $Q_{ID_i} = H_1(ID_i) = b_iQ$, add the tuple $< coin_i = 1, ID_i, b_i, Q_{ID_i} >$ to L_1.

***H_2*-queries**. $\mathcal{A}_2$ can issue H_2 queries at any time. $\mathcal{B}$ forwards it to $\mathcal{B}$'s challenger and returns the answer to $\mathcal{A}_2$.

***H_3*-queries**. When $\mathcal{A}_2$ queries H_3 at a distinct point (U_i, V_i, W_i), $\mathcal{B}$ defines $\bar{P}_i$ at that point by choosing $T_i \in \mathbb{Z}_q^*$ uniquely, and maintains an initially empty list L_3 of tuples $< T_i, (U_i, V_i, W_i) >$. Return $\bar{P}_i = T_iP_{pub}$.

***H_4*-queries**. When $\mathcal{A}_2$ queries H_4 at a distinct point $(\bar{P}_i, \bar{U}_i, \bar{W}_i)$, $\mathcal{B}$ chooses $e_i \in \mathbb{Z}_q^*$ uniquely at random as the answer and maintains an initially empty list L_4 of tuples $< e_i, (\bar{P}_i, \bar{U}_i, \bar{W}_i) >$.

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

Key extraction queries1. $\mathcal{A}_2$ issues a number of key extraction queries on ID_i adaptively. It's reasonable to assume that $\mathcal{A}_2$ has asked about $H_1(ID_i)$ before issuing complete decryption key extraction queries on an identity ID_i. Let $< coin_i, ID_i, b_i, Q_{ID_i} >$ be the corresponding tuple on the L_1 list.

- *Complete private key extraction queries.*
  - If $coin_i = 0$, $\mathcal{B}$ outputs the decryption key of ID_i as $d_{ID_i} = b_iP_{pub}$.
  - Else if $coin_i = 1$, $\mathcal{B}$ terminates the game and outputs "*Abort*".
- *Validity key and the corrupted KEA's private key share queries.*
  - If $coin_i = 0$, $\mathcal{B}$ randomly chooses a polynomial of degree $t-1$ over $\mathbb{G}_2^*$: $f_{ID_i}(x) = b_i + \sum_{j=1}^{t-1} a_jx^j$. Then compute $S_{ID_i}^{(k)} = f_{ID_i}(k)P_{pub} + m_kP_{pub}, V_{ID_i}^{(k)} = \hat{e}(f_{ID_i}(k)P_{pub}, P)$ for $k = 1, 2, \ldots, n$. Return $V_{ID_i}^{(k)}$ and $S_{ID_i}^{(j)}(j \in S)$ to $\mathcal{A}_2$. Then add $< ID_i, \{S_{ID_i}^{(j)}\}_{(j=t,t+1,\ldots,n)} >$ to the list L_{ks}.
  - Else if $coin_i = 1$, $\mathcal{B}$ returns $S_{ID_i}^{(j)} = b_id_j + m_jP_{pub}$ for $j \in S$, $V_{ID_i}^{(k)} = \hat{e}(b_iQ, P_{pub}^{(k)})$ for $k = 1, 2, ..., n$.

  It's easy to prove that $S_{ID_i}^{(j)}, V_{ID_i}^{(k)}$ ($j \in S$ and $k \in \{1, 2, ..., n\}$) can pass the validity test of key shares. (When $coin_i = 1$, we make use of the fact that $\{d_l\}_{l \in S}, \{P_{pub}^{(k)}\}_{(k=1,2,\ldots,n)}$ can pass the validity test of $\mathcal{B}$).
- *Update KEA's public key.* 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, i.e. $\hat{e}(P_j', P_{pub}) = \hat{e}(\Delta_j, P)$). $\mathcal{B}$ accepts $\mathcal{A}_2$'s request and computes the partial keys upon ID_i as
  - if $coin_i = 0$, $S_{ID_i}^{(j)} = f_{ID_i}(j)P_{pub} + \Delta_j$,
  - if $coin_i = 1$, $S_{ID_i}^{(j)} = b_id_j + \Delta_j$.

  The public verification keys $V_{ID_i}^{(j)}$ keeps invariably.

Decryption queries1. Given a ciphertext $C_i = (V_i, U_i, \bar{U}_i, e_i, f_i)$ that is encrypted under ID_j and M_i, $\mathcal{B}$ can simulate the decryption oracle and the uncorrupted KEAs via L_3, L_4 and L_{ks}. It responds to decryption queries as follows.

- First, $\mathcal{B}$ computes $W_i = f_iP - e_iU_i$ and searches the L_3 list for a tuple

$< T_i, (U_i, V_i, W_i) >$ containing (U_i, V_i, W_i). If it is nonexistent, $\mathcal{B}$ returns "$InvalidCiphertext$".

- Else $\mathcal{B}$ searches L_4 for a tuple $< e_i, (\bar{P}_i, \bar{U}_i, \bar{W}_i) >$, where $\bar{P}_i = T_i P_{pub}, \bar{W}_i = f_i \bar{P}_i - e_i \bar{U}_i$. If $\mathcal{B}$ fails, return "$InvalidCiphertext$".
- Else, M_i and $\delta^l_{ID_j, C_i} (l = t, t+1, ..., n)$ can be computed as follows:
  - if $coin_j = 0$, when $\mathcal{A}_2$ queries $\mathcal{B}$ at C_i, $\mathcal{B}$ performs the following:
    - since $\hat{e}(d_{ID_j}, U_i) = \hat{e}(b_j P_{pub}, U_i)$, output $M_i = V_i \oplus H_2(\hat{e}(b_j P_{pub}, U_i))$;
    - with $S^{(l)}_{ID_j}$ and $k^l_{ID_j} = \hat{e}(S^{(l)}_{ID_j} - m_l P_{pub}, U_i)$, $\mathcal{B}$ can readily run *Proof-Log* to output the decryption share $\delta^l_{ID_j, C_i} = \{l, k^l_{ID_j}, h_l, \lambda_l\}$.
  - if $coin_j = 1$, although $\mathcal{B}$ cannot get r_i from $U_i = r_i P$, he can assume $\bar{U}_i = r_i \bar{P}_i$ and simulate the decryption of C_i as:
    - since $\hat{e}(d_{ID_j}, U_i) = \hat{e}(b_j sQ, r_i P) = \hat{e}(b_j Q, r_i(T_i P_{pub}))^{\frac{1}{T_i}}$, output $M_i = V_i \oplus H_2(\hat{e}(b_j Q, \frac{1}{T_i} \bar{U}_i))$;
    - with $d_1, d_2, ..., d_{t-1}$, $k^l_{ID_j}$ can be computed as $k^l_{ID_j} = \hat{e}(S^{(l)}_{ID_j} - m_l P_{pub}, U_i) = \hat{e}(L^{S'}_{l0} Q, \bar{U}_i)^{\frac{b_j}{T_i}} \cdot \prod_{k=1}^{t-1} \hat{e}(L^{S'}_{lk} d_j, U_i)^{b_j}$. Where $L^{S'}_{lm} = \prod_{j \in S', j \neq m} \frac{l-j}{m-j} (\mathrm{mod}\ q)$ is the Lagrange coefficient with respect to $S' = \{0\} \cup S$, for $m = 0, 1, ..., t-1$. Run *Proof-Log*, and return $\delta^l_{ID_j, C_i} = \{l, k^l_{ID_j}, h_l, \lambda_l\}$.

Challenge. Adversary $\mathcal{A}_2$ issues 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 $coin_{ch} = 0$ then $\mathcal{B}$ terminates the game and reports "*Abort*".
- If $coin_{ch} = 1$ then $\mathcal{B}$ forwards (M_0, M_1) to its challenger. When it receives the `BasicThIBE` ciphertext $C' = (U', V')$, $\mathcal{B}$ simply chooses $e^*, f^*, l^* \in \mathbb{Z}_q^*$ and sets $V^* = V', U^* = b_{ch}^{-1} U', \bar{P}^* = l^* P, \bar{U}^* = l^* U^*, W^* = f^* P - e^* U^*, \bar{W}^* = f^* \bar{P}^* - e^* \bar{U}^*$. Then $\mathcal{B}$ backpatches and defines the challenge ciphertext $C^* = (V^*, U^*, \bar{U}^*, e^*, f^*)$. Where C^* is the `IB-ThKE` encryption of $M_{b'}$ for a random $b' \in \{0, 1\}$ under the public key ID_{ch} as required, and b_{ch}^{-1} is the inverse of b_{ch} mod q.

Key extraction queries2. Adversary $\mathcal{A}_2$ makes more queries. $\mathcal{B}$ responds in the same way as in key extraction queries1, except the complete decryption key of ID_{ch}.

Decryption queries2. $\mathcal{A}_2$ issues more decryption queries, $\mathcal{B}$ runs in the same way it did in decryption queries1. The only restriction here is that the target ciphertext C^* is not allowed to be queried.

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

Analysis.

- Suppose $\mathcal{B}$ is given a ciphertext $C \neq C^*$, where $C =< (V, U, \bar{U}, e, f), W, \bar{W} >$. If C can pass the *ciphertext validity verification*, and $<V, U, W> \neq <V^*, U^*, W^*>$. Then $\mathcal{A}_2$ must has queried H_3 at the point $< U, V, W >$. So $\mathcal{B}$ has $\bar{P} = H_3(U, V, W) = tP_{pub}$ and $\mathcal{B}$ can exactly decrypt C as described above.

  Else if C can pass the ciphertext validity test while $< V, U, W > = < V^*, U^*, W^* >$ and $< \bar{U}, e, f > \neq < \bar{U}^*, e^*, f^* >$, then $\bar{P} = \bar{P}^*$, $U = rP$ and $\bar{U} = r' \bar{P}$ with

$r \neq r'$ (if $r' = r$, then $t = t'$ and $C = C^*$). Set $W = tP$ and $\bar{W} = t'\bar{P}$. Because $\mathcal{B}$ accepts C, we have $f = t + er = t' + er'$. So, since $r - r' \neq 0$ and H_4 is a random oracle controlled by $\mathcal{B}$, this happens with probability at most $\frac{1}{q}$. It's negligible when q is large enough.

Note that if we additionally check whether $\hat{e}(P, \bar{U}) = \hat{e}(\bar{P}, U)$ in the *ciphertext validity verification*, making use of the decision Diffie–Hellman problem is polynomially solvable in $< \mathbb{G}_1, \mathbb{G}_2, \hat{e} >$, then the latter case can be prevented readily. However, we conceal it for the efficiency of validity test. As shown above, it does not reduce the security of our scheme.

- If $\mathcal{B}$ does not abort during the game, 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 $\mathcal{B}$ aborts in the game. Furthermore, $\hat{e}(d_Q, U') = \hat{e}(d_{ID_{ch}}, U^*)$. Thus, by the definition of $\mathcal{A}_2$, we have $|2Pr(b'' = b') - 1| = Adv(\mathcal{A}_2) = \epsilon$. Let $\mathcal{H}$ denote the event that $\mathcal{B}$ does not abort in the game, then the advantage of $\mathcal{B}$ is $\epsilon' > \epsilon \cdot Pr[\mathcal{H}]$. We name the event that $\mathcal{A}_2$ made a complete private key extraction queries on ID_i with $coin_i = 1$ at some points as $\mathcal{E}_1$, and the event that $\mathcal{A}_2$ chose ID_{ch} with $coin_{ch} = 0$ as $\mathcal{E}_2$. If $Pr[coin = 0] = \delta$, then $Pr[\mathcal{H}] = Pr[\neg\mathcal{E}_1 \bigwedge \neg\mathcal{E}_2] = \delta^{q_E}(1 - \delta)$. This value is maximized when $\delta_{opt} = 1 - 1/(q_E + 1)$ (Boneh, 2001). Using δ_{opt}, $Pr[\mathcal{H}]$ is at least $1/e(1 + q_E)$. This shows that $\mathcal{B}$'s advantage is at least $\epsilon' = \epsilon/e(1 + q_E)$. This finishes the proof.

Lemma 4.2. *Let H_2 be a public one way hash functions from $\mathbb{G}_2$ to $\{0,1\}^n$, and let $\mathcal{B}$ be an IND-TH-CPA adversary that has the advantage ϵ' against the* `BasicThIBE`*. Suppose $\mathcal{B}$ makes q_{H_2} hash queries to H_2. Then there is an algorithm $\mathcal{C}$ that solves the decision BDH problem with the advantage at least $\epsilon'' = \epsilon'/2$.*

Proof. Algorithm $\mathcal{C}$ is given a random instance (P, aP, bP, cP, D) of the decision BDH problem where a, b, c are random in Z_q^*. To determine whether $D = \hat{e}(P, P)^{abc}$ or not, $\mathcal{C}$ 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 $S = \{1, 2, ..., t - 1\}$.

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

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

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

Challenge. $\mathcal{B}$ outputs two equal length plaintexts (M_0, M_1). $\mathcal{C}$ chooses a random bit $b' \in \{0,1\}$ and a random $R \in \{0,1\}^l$. Set $H_2(D) = R$, return $C' = (U,V) = (aP, M_{b'} \oplus R)$ as the challenge ciphertext.

Guess. $\mathcal{B}$ outputs a guess $b'' \in \{0,1\}$. If $b'' = b'$, then $\mathcal{C}$ outputs 1 meaning $D = \hat{e}(P,P)^{abc}$. Otherwise, return 0 meaning $D \neq \hat{e}(P,P)^{abc}$.

Analysis. If $D = \hat{e}(P,P)^{abc}$, then $\mathcal{B}$'s view is identical to its view in the real game, and $Pr[b'' = b'] = \frac{1}{2} \pm \frac{\epsilon'}{2}$ by the definition of $\mathcal{B}$. Else if $D \neq \hat{e}(P,P)^{abc}$ then D is uniform and independent in $\mathbb{G}_1$, and the challenge ciphertext C' is independent of b', that is $Pr[b'' = b'] = \frac{1}{2}$. Therefore, we have $|Pr[\mathcal{B}(P, aP, bP, cP, \hat{e}(P,P)^{abc}) = 1] - Pr[\mathcal{B}(P, aP, bP, cP, D) = 1]| = |(\frac{1}{2} \pm \frac{\epsilon'}{2}) - \frac{1}{2}| = \epsilon'/2$.

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

5. Further Discussion

The `IB-ThKE` can readily be converted into a fully secure identity-based threshold decryption scheme. There are only type2 adversaries in the threshold decryption scheme, so its security reduction is similar to `IB-ThKE`. In such a scheme, the ciphertexts are publicly checkable, without pairing computation.

Another application of `IB-ThKE` is a dynamic threshold key escrow scheme (Sun, 1994). Intuitively, each KEA only needs to keep his private key s_i in secret, while the partial secret keys $S_{ID}^{(i)}$ can be transmitted through public channels, without betraying any information of the complete decryption key. Then the KMC can update the master key s, or add/remove any KEA without changing KEAs' private information. That is, the KEAs and KMC may communicate via broadcast channels after the key generation process has taken place, and the secret channel is demanded only when a KEA wants to update his private key. Obviously, this scheme is very practical.

6. Conclusions

In this paper, we propose a threshold key escrow scheme that can resist against active attack, and we show its security against adaptive chosen ciphertext secure in the appropriate model, assuming that the decision Bilinear Diffie–Hellman problem is hard. At last, we illustrate the applications of `IB-ThKE`.

References

Al-Riyami, S.S., K.G. Paterson (2003). Certificateless public key cryptography. *Asiacrypt 2003*. *LNCS 2894*, Springer. pp. 452–473.

Baek, J., and Y. Zheng (2004). Identity-based threshold decryption. In *Proceedings of PKC'04*. *LNCS 2947*, Springer. pp. 262–276.

Bellare, M., and P. Rogaway (1993). Random oracles are practical – A paradigm for designing efficient protocols. In *Proceedings of the First ACM Conference on Computer and Communications Security*. pp. 62–73.

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

Blum, M., A.D. Santis, S. Micali and G. Persiano (1991). Non-interactive zero knowledge. *SIAM J. Comput.*, **6**(4), 1084–1118.

Boneh, D., and M. Franklin (2001). Identity-based encryption from the weil pairing. *Advances in Cryptology-Crypto 2001*. *LNCS 2139*, Springer. pp. 213–229.

Chai, Z.C., Z.F. Cao and R.X. Lu (2004). ID-based threshold decryption without random oracles and its application in key escrow. In *Inforsec 2004*. ACM press. pp. 119–124.

Denning, D. E., and M. Smid (1994). Key escrowing today. *IEEE Communications Magazine*, 58–68.

Desmedt, Y., and Y. Frankel (1989). Threshold cryptosystems. *Proceeding of CRYPTO'89*. *LNCS 435*. Springer. pp. 307–315.

Fouque, P., and D. Pointcheval (2001). Threshold cryptosystems secure against chosen-ciphertext attacks. *Proceedings of Asiacrypt 2001*. *LNCS 2248*, Springer. pp. 351–368.

Libert, B., and J. Quisquater (2003). Efficient revocation and threshold pairing based cryptosystems. In *PODC 2003*. ACM press. pp. 163–171.

Lim, C., and P. Lee (1993). Another method for attaining security against adaptively chosen ciphertext attack. *Proceedins of Crypto'93*, *LNCS 773*. Springer. pp. 410–434.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, **22**(11), 612–613.

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

Sun, H.M., and S.P. Shieh (1994). Construction of dynamic threshold schemes. *Electronics Letters*, **30**(24), 2023–2025.

Waters, B. (2005). Efficient identity-based encryption without random oracles. *Advances in Cryptology-Eurocrypt 2005*, *LNCS 3494*. Springer. pp. 114–127.

Y. Long received the BS degree in computer science and technology in South West Jiao Tong University, China, and received the MS degree in Information Security Engineering Department, Shanghai Jiao Tong University. Now she is a doctor candidate in the same school. Her research interests include information theory and modern cryptography etc.

K.F. Chen received his PhD 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 became the professor at the Department of Computer Science and Engineering. Up to now (1996–2006), more than 80 academic papers on cryptology and information security have been published in journals.

S.L. Liu received her first PhD degree in Xidian University in 2000, and received her second PhD 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. Up to now (1999–2006), more than 20 high quality papers on information security have been published in journals.

Adaptyviai parinkta šifruoto teksto slapto slenkstinio rakto sąlyginio indėlio schema

Yu LONG, Kefei CHEN, Shengli LIU

Straipsnyje siūloma šifruoto teksto slapto slenkstinio rakto sąlyginio indėlio schema. Ji toleruoja pasyvų priešininką, siekiantį prieiti prie vidinių duomenų, esant pažeistiems agentams, ir aktyvų priešininką, kuris gali pažeisti serverį, kad jis nukryptų nuo protokolo. Formalus saugumo įrodymas pateiktas randomizuotam oraklo modeliui, skaitant, kad bitiesinė Diffie–Hellmano problema yra skaičiuojamuoju požiūriu sunki.