一种新的基于身份选择密文安全的门限解密方案

龙 宇[1),2)] 陈克非[1),2)] 洪 璇[1)]

[1)](上海交通大学计算机科学与工程系密码与信息安全实验室 上海 200240)

[2)](现代通信国家实验室 成都 610041)

摘 要 该文提出了一种具有完备安全性的、基于身份的门限解密方案 IB-ThDec. 方案的安全性可以规约到四元双线性 Diffie-Hellman 判定问题上. 我们在随机预言模型下给出了方案的安全性证明. 此外,我们指出 IB-ThDec 方案可以应用到无证书体制和基于身份的动态门限解密体制中.

关键词 门限解密;基于身份的密码学;自适应选择密文攻击;基于配对的密码学

中图法分类号 TP309

A New Chosen Ciphertext Secure ID-Based Threshold Decryption Scheme from Pairing

LONG Yu[1),2)] CHEN Ke-Fei[1),2)] HONG Xuan[1)]

[1)](*Cryptography and Information Security Laboratory, Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai* 200240)

[2)](*National Laboratory of Modern Communications, Chengdu* 610041)

Abstract This paper proposes a fully secure identity-based threshold decryption scheme IB-ThDec and reduces its security to the quadruple Decision Bilinear Diffie-Hellman problem. The formal security proof of this scheme is provided in the random oracle model. Additionally, the authors show that IB-ThDec can be used to certificateless public key cryptosystem and the identity-based dynamic threshold decryption system.

Keywords threshold decryption; identity-based cryptography; adaptive chosen-ciphertext attack; pairing based cryptology

1 Introduction

With the development of networks, distributed cryptography has received a lot of attention in modern cryptographic research. Threshold cryptosystems provide security to distributed applications, and can avoid single point of failure in a network system. In 2001, Boneh and Franklin proposed a practical identity (ID)-based encryption scheme from the Weil pairing[1]. 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 needs for certificates.

Combining IB-PKC with threshold cryptosystems, some "ID-based threshold decryption" schemes have been constructed. In such schemes, an entity's public key is derived directly from its identity. A trusted third party called the Private Key Generator (PKG) uses the master key to generate private keys for all entities. The power of de-

收稿日期:2006-04-04;**修改稿收到日期**:2006-06-02. **本课题得到国家自然科学基金**(60303026,60473020,60573030)**和** NLMC **基金**(51436040405JW0304)**资助**.**龙 宇**,**女**,1980 **年生**,**博士研究生**,**目前主要从事信息论与现代密码学的研究**. E-mail: longyu@sjtu.edu.cn. **陈克非**,**男**,1959 **年生**,**博士**,**教授**,**博士生导师**,**研究兴趣包括经典与现代密码学、网络安全的理论与技术等**. **洪 璇**,**女**,1982 **年生**,**博士研究生**,**研究兴趣为签名与可证安全**.

cryption is shared among a set of decryption servers, each of which holds a piece of private key associated with an identity or a piece of the master key. When given a ciphertext, which is the output of an ID-based encryption algorithm under an identity, a quorum of servers can act together to decrypt it.

Our contribution is to present and analyze such a scheme based on the difficulty of the quadruple Decision Bilinear Diffie-Hellman problem, named ID-based threshold decryption (IB-ThDec) scheme. This scheme is provably adaptive chosen-ciphertext secure, in the random oracle model. At last, we show IB-ThDec implies a fully secure certificateless public key decryption scheme, and an ID-based dynamic threshold decryption system resisting against adaptive chosen-ciphertext attack in the selective identity model.

Other related works: To our knowledge, other papers that have treated threshold decryption in the context of ID-based cryptography are [2～5]. However schemes in [2,4,5] were only chosen-plaintext secure, and the formal security proof in [3] was in a weaker model named selective-ID model[2]. Baek and Zheng added a tag after the ciphertext to convert it into a fully secure system[3]. Their main idea was the validity of the ciphertext became publicly checkable. However, in [3], security reduction didn't touch on the probability of the event that the adversary obtain the decryption of ill-formed (but valid-look) ciphertext. What's more, all of these schemes could not tolerate active adversary that can modify the public verification information of corrupted decryption servers.

2 Preliminaries

2.1 Admissible bilinear pairings

Let G_1 be a cyclic additive group and G_2 be a cyclic multiplicative group of the same prime order q. Assuming that the discrete logarithm problem in both G_1 and G_2 are hard, an admissible bilinear pairing is a map $e: G_1 \times G_1 \rightarrow G_2$ which satisfies the following properties:

(1) Bilinear: For any $P,Q \in G_1$ and $a,b \in Z_q^*$, $e(aP,bQ)=e(p,Q)^{ab}$.

(2) Non-degenerate: There exists $P,Q \in G_1$ such that $e(P,Q) \neq 1$.

(3) Computable: Given $P,Q \in G_1$, there is an efficient algorithm to compute $e(P,Q) \in G_2$.

2.2 Quadruple Bilinear Diffie-Hellman Problem (4-BDHP)

We now give the description of the quadruple problem in bilinear group systems.

Definition 1 (4-BDH problem). Let $e: G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear map. Let P be a generator of G_1, whose order is a large prime q. Let a,b,c,d be elements of Z_q^*. Given $(P,aP,bP,cP,abP,bcP,acP,dP)$, output $D=e(P,P)^{abcd}$.

Definition 2 (4-Decision BDH problem). Let $e: G_1 \times G_1 \rightarrow G_2$ be an admissible bilinear map. Let P be a generator of G_1, whose order is a large prime q. Let a,b,c,d be elements of Z_q^*, and randomly choose $D \in G_2^*$. Given $(P,aP,bP,cP,abP,bcP,acP,dP)$, determine whether $D=e(P,P)^{abcd}$.

An algorithm A that outputs $b' \in \{0,1\}$ has an advantage ε in solving the 4-Decision BDH (4-DBDH) problem in $\langle G_1, G_2, e \rangle$ if $|Pr[A(P,aP,bP,cP,dP,abP,acP,bcP,e(P,P)^{abcd}=1]-Pr[A(P,aP,bP,cP,dP,abP,acP,bcP,D)=1]| > \varepsilon$, where D is randomly chosen from G_2. Virtually, the 4-DBDH problem can be viewed as the combination of three general Decision Bilinear Diffie-Hellman problems. Thus it's hard in $\langle G_1, G_2, e \rangle$. That means there is no probabilistic algorithm that can solve the 4-DBDH problem with a non-negligible advantage ε within polynomial time.

2.3 Threshold Security

The idea of (t,n) threshold cryptosystem was proposed in [6,7]. The formal security model of threshold cryptosystems has been discussed in [8,9]. 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-CCA2[10]) the adversary sees both the decryptions of chosen ciphertexts and the decryption shares of these ciphertexts. This extra information makes it's very difficult to construct an IND-TH-CCA2 secure threshold cryptosystem.

In [9], two secure threshold cryptosystems against chosen ciphertext attack are proposed. In this work, non-interactive zero knowledge proof of membership was used to make the ciphertext publicly checkable[11,12]. Motivated by [9], we present an ID-based threshold decryption scheme IB-ThDec, and prove its security in the sense of threshold adaptive chosen-ciphertext attack in the random oracle model[13].

2.4 Non-interactive proof of membership

Similar to [3], a non-interactive zero knowledge proof of membership system named Proof-Log can be constructed for the language $L=\{(v,\tilde{v}) \in G_2 \times G_2 \mid \log_g v = \log_{\tilde{g}} \tilde{v}\}$, where $g=e(P,P)$ and $\tilde{g}=$

$(P,\widetilde{P})$. P and $\widetilde{P}$ are generators of G_1. G_1, G_2, e have the same definitions as in section 2.1.

Given $(P,\widetilde{P},g,\tilde{g})$, a one-way hash function $H_5: G_2\times G_2\times G_2\times G_2\to Z_q^*$ and $(k,\tilde{k})\in L$, the Prover wants to convince the Verifier that he indeed knows a secret $S=(\log_g k)P=(\log_{\tilde{g}}\tilde{k})P\in G_1$ without yielding any "knowledge" of S. The proof system works like this:

(1) The Prover randomly chooses $T\in G_1^*$, then computes $\gamma=e(T,P)$, $\tilde{\gamma}=e(T,\widetilde{P})$, $h=H_5(e(P,S), e(\widetilde{P},S), e(T,P), e(T,\widetilde{P}))$ and $L=T+hS\in G_1$. Send $\{\gamma,\tilde{\gamma},L\}$ to the Verifier.

(2) The Verifier computes $h=H_5(k,\tilde{k},\gamma,\tilde{\gamma})$ and checks whether $e(L,P)=\gamma\cdot k^h$ and $e(L,\widetilde{P})=\tilde{\gamma}\cdot\tilde{k}^h$. 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 $S\in G_1^*$ such that $k=e(S,P)$ and $\tilde{k}=e(S,\widetilde{P})$, and the properties of this protocol can be discussed as in [3].

2.5 Modified basic CL-PKE scheme using binding technique

In [14], the concept of certificateless public key encryption (CL-PKE) is proposed. And the binding technique is used to reduce the degree of trust that users need to have in the Key Generation Center (KGC). We slightly modify their scheme to construct a basic certificateless scheme (BasicPub-CL-PKE). Following algorithms are used to define the BasicPub-CL-PKE.

Setup. Run by the KGC. It takes as input a security parameter k_0, to output G_1, G_2, e, q. Choose a generator $P\in G_1$, a master key $s\in Z_q^*$, and publish $P_{\text{pub}}=sP$. Choose one hash function $H_2: G_2\to\{0,1\}^l$, where l is the bit-length of plaintexts. Then the system public parameters are $params=\{G_1, G_2, e, l, q, P, P_{\text{pub}}, H_2\}$.

Set-Secret-Value-and-Public-Key. Takes as inputs $params$ and a random $Q_A\in G_1^*$. Choose random $x_A\in Z_q^*$ as an entity A's secret value, and output $P_A=(X_A, Y_A)=(x_AP, x_AP_{\text{pub}})$ as A's public key.

Public-Partial-Key-Extract. Takes as input Q_A, x_A, s. It outputs the public partial key pair $(D_A, D_A')=(sQ_A, x_AQ_A)$.

Set-Private-Key. Takes as inputs A's secret value x_A and the public partial key D_A. It outputs A's private key $S_A=x_A(D_A)\in G_1^*$.

BasicPuB-Encrypt. To encrypt a message M with Q_A, P_A, first check whether $e(X_A, P_{\text{pub}})=e(Y_A, P)$. If not, output $\perp$ and abort. Else choose random $r\in Z_q^*$, and output $C=(U,V)=(rP, M\oplus H_2(e(Q_A, Y_A)^r))$.

Decrypt. Suppose the ciphertext $C=(U,V)=(rP, M\oplus H_2(e(Q_A, Y_A)^r))$. To decrypt it using the private key S_A, compute and output $V\oplus H_2(e(S_A, U))$.

Note that the modified basic CL-PKE scheme is introduced to simplify the security analysis in section 4.3. In the security proof of the BasicPub-CL-PKE, Q_A can be looked as the fixed identity of only one user. Thus, Q_A can be added to the public parameters.

3 ID-based (t,n) threshold decryption scheme from pairing

In this section, we describe a pair of ID-based (t,n) threshold decryption scheme. The first scheme is only a basic scheme which is analogous to the BasicPub-CL-PKE and is only semantic secure against chosen-plaintext attack. And then the full scheme is presented, which is provably fully secure in the random oracle model.

3.1 The basic ID-based threshold decryption scheme

The basic ID-based threshold decryption scheme Basic-IB-ThDec works as follows:

Setup. Run by the PKG.

(1) Given a security parameter k_0, the PKG chooses $G_1, G_2, q(>2^{k_0}), e: G_1\times G_1\to G_2$, a generator $P\in G_1$, the master key $s\in Z_q^*$, and publishes $P_{\text{pub}}=sP$. Choose two cryptography hash functions $H_1: \{0,1\}^*\to G_1^*$, $H_2: G_2\to\{0,1\}^l$.

(2) Choose $a\in Z_q^*$ and output $\langle X,Y\rangle=\langle aP, aP_{\text{pub}}\rangle$. Then the public parameters are: $cp=\{q,l, G_1, G_2, e, H_1, H_2, P, P_{\text{pub}}, X, Y\}$.

Share-Key-Gen. Given an identity ID, the PKG chooses a polynomial of degree $t-1$ over Z_q^*: $f(x)=s+a_1'x+\cdots+a_{t-1}'x^{t-1}$. It computes and publishes $P_{ID}=sQ_{ID}=sH_1(ID)$, $P_{ID}^{(i)}=f(i)Q_{ID}\in G_1$ for $1\le i\le n$. Then deliver $d_{ID,i}=af(i)Q_{ID}\in G_1$ to the i-th decryption server Γ_i secretly. When receiving $d_{ID,i}$, Γ_i can check its validity by $e(P_{ID}^{(i)}, X)=e(d_{ID,i}, Y)$ and $\sum_{i\in T}L_i^T(P_{ID}^{(i)})=P_{ID}$, where $T\subset\{1,2,\cdots,n\}$, $|T|=t$ and $L_i^T=\prod_{j\in T, j\neq i} j/(j-i)$ is the Lagrange coefficient with respect to the set T. If the validity test fails, he complains to the PKG that issues a new share. Note that P_{ID}'s validity is publicly checkable by $e(P_{ID}, X)=e(Q_{ID}, Y)$.

Basic-IB-Encrypt. To encrypt a message $M\in\{0,1\}^l$ under ID, the sender computes $Q_{ID}=H_1(ID)$, and then chooses a random $r\in Z_q^*$. The ciphertext is given by $(U,V)=(rP, M\oplus H_2(e(Q_{ID}, Y)^r))$.

Decrypt. When receiving (U,V), decryption

server Γ_i computes his decryption share $\delta_{i,ID,C}=e(U,d_{ID,i})$ and gives it to a special server called the *combiner*.

Recombine. The combiner selects a set $T\subset\{1,2,\cdots,n\}$ of t decryption shares $\delta_{i,ID,C}$ and computes $g=\prod_{i\in T}\delta_{i,ID,C}^{L_i^T}$. Then the plaintext can be recovered by $M=V\oplus H_2(g)$.

This basic scheme is presented to help the understanding and security proof of the full scheme. Note that the PKG can use one polynomial to compute and distribute key shares of different identities, without degradation of system security.

3.2 The full ID-based threshold decryption scheme

First, we sketch the characteristics of our full ID-based threshold decryption scheme IB-ThDec.

In the ID-based (t,n) threshold decryption scheme IB-ThDec from pairing, the system consists of a trusted Private Key Generator (PKG), n decryption servers $\Gamma_i(1\leq i\leq n)$, and many communication users. Similar to IB-PKC, the public keys are the unambiguous identity of the users, such as the email address or a telephone number. The plaintext M that is encrypted under an identity is recoverable from at least t of n decryption servers. Every decryption server has a private key chosen by itself. And the corresponding public verification key is given to the PKG. When an entity wants to decrypt a received ciphertext from Alice, PKG returns the partial secret keys and the public verification keys of Alice to at least t Γ_is. Then each Γ_i can generate a decryption share of this ciphertext, taking as input the ciphertext and the partial secret key and his private key. These shares are sent to a combiner, who starts checking the validity of every share. If more than t shares are valid, the combiner combines them to obtain the plaintext.

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

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

Setup. Run by the PKG and n decryption servers $\Gamma_i(1\leq i\leq n)$.

(1) Given a security parameter k_0, the PKG outputs two groups G_1 and G_2 of the same prime order $q(>2^{k_0})$, an admissible bilinear map $e: G_1\times G_1\to G_2$, a generator $P\in G_1$, a master key $s\in Z_q^*$. Compute $P_{\text{pub}}=sP$ and choose five hash functions $H_1:\{0,1\}^*\to G_1$, $H_2:G_2\to\{0,1\}^l$, $H_3:G_1\times G_1\times G_1\to Z_q^*$, $H_4:G_1\times G_1\times G_1\to Z_q^*$ and $H_5:G_2\times G_2\times G_2\times G_2\to Z_q^*$. Note that H_1, H_2, H_3, H_4 are viewed as random oracles in the security analysis[13]. Then randomly choose $a\in Z_q^*$, publish $\langle X,Y\rangle=\langle aP,aP_{\text{pub}}\rangle$.

(2) Γ_i randomly selects $s_i\in Z_q^*$, and computes $P_i=s_iP$. Keep s_i as Γ_i's private key.

The system public parameters are:

$$cp=\{q,l,G_1,G_2,P,P_{\text{pub}},e,\{H_j\}_{1\leq j\leq 5},\{P_i\}_{1\leq i\leq n},X,Y\}.$$

Key-Gen-1. Given an user's identity ID, the PKG returns $P_{ID}=sH_1(ID)$ publicly. The user can check its validity with $e(P_{ID},X)=e(H_1(ID),Y)$.

Key-Gen-2. Given ID, the PKG chooses a polynomial of degree $t-1$ over Z_q^*:

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

For $i=1,2,\cdots,n$, it computes a key share $(S_{ID}^{(i)},V_{ID}^{(i)})=(af(i)Q_{ID}+sP_i,e(f(i)Q_{ID},X))$. $S_{ID}^{(i)}$ are returned to Γ_i secretly, while $V_{ID}^{(i)}$ are transmitted through public channels.

Full-IB-Encryption. To encrypt a message $M\in\{0,1\}^l$ under the receiver's identity ID, the sender chooses $r,t'\in Z_q^*$ uniformly at random and computes $Q_{ID}=H_1(ID)$. Then set the ciphertext to be $(V,U,\bar{U},e',f)$, where $V=M\oplus H_2(e(Q_{ID},Y)^r)$, $U=rP$, $W=t'P$, $\bar{P}=H_3(U,V,W)$, $\bar{U}=r\bar{P}$, $\bar{W}=t'\bar{P}$, $e'=H_4(\bar{P},\bar{U},\bar{W})$ and $f=t'+e'r$.

Ciphertext-Validity-Test. Let $C=(V,U,\bar{U},e',f)$ be a ciphertext encrypted under an identity ID. Then C's validity is publicly checkable. That is, everybody can check whether $e'=H_4(\bar{P},\bar{U},\bar{W})$, where $W=fP-e'U$, $\bar{P}=H_3(U,V,W)$ and $\bar{W}=f\bar{P}-e'\bar{U}$. If not, output "Invalid Ciphertext".

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

(1) (Key share verification) First, Γ_i checks the validity of $(S_{ID}^{(i)},V_{ID}^{(i)})$ with $e(S_{ID}^{(i)},P)=V_{ID}^{(i)}\cdot e(P_i,P_{\text{pub}})$. And everybody can check whether $\prod_{i\in T}(V_{ID}^{(i)})^{L_i^T}=e(Q_{ID},Y)$ for any subset $T\subset\{1,2,\cdots,n\}$ such that $|T|=t$, where L_i^T denotes the appropriate Lagrange coefficient with respect to the set T. If $(S_{ID}^{(i)},V_{ID}^{(i)})$ can not pass this test, Γ_i outputs $(i,$ "Invalid Key Share").

(2) Else Γ_i checks the validity of the ciphertext as in the ciphertext validity test. If it does not hold, output $(i,$ "Invalid Ciphertext").

(3) Otherwise, both tests succeed. Compute $k_{ID}^i=e(S_{ID}^{(i)}-s_iP_{\text{pub}},U)$, $R_i=e(T_i',P)$, $\tilde{R}_i=e(T_i',U)$, $h_i=H_5(V_{ID}^{(i)},k_{ID}^i,R_i,\tilde{R}_i)$, $\lambda_i=T_i'+h_i(S_{ID}^{(i)}-s_i(P_{\text{pub}}))$ for random $T_i'\in G_1^*$. Then output the de-

cryption share $\delta^i_{ID,C}=\{i,k^i_{ID},h_i,\lambda_i\}$.

Decryption. Given a ciphertext $C=(V,U,\overline{U},e',f)$ and a set of decryption shares $\{\delta^i_{ID,C}\}_{i\in T'}$, the combiner runs as follows:

(1) (Decryption share verification) For $i\in T'$, check if $h_i=H_5(V^{(i)}_{ID},k^i_{ID},R_i,\widetilde{R}_i)$, where $R_i=e(\lambda_i,P)/(V^{(i)}_{ID})^{h_i}$ and $\widetilde{R}_i=e(\lambda_i,U)/(k^i_{ID})^{h_i}$. If it fails, discard the decryption share and return (i, "Invalid Decryption Server Γ_i").

(2) Else if the combiner collects t valid decryption shares from $\Gamma_i(i\in T,T\subseteq T',|T|=t)$, it computes $K=\prod_{i\in T}(k^i_{ID})^{L^T_i}$ and $M=V\oplus H_2(K)$. Then the ciphertext is decrypted by the combiner.

Γ_i's-Public-Key-Updating. In this scheme, we allow the decryption server $\Gamma_i(i\in\{1,2,\cdots,n\})$ to renew his private key s_i as follows:

(1) Γ_i chooses $s'_i\in Z^*_q$. Compute $P'_i=s'_iP$ and $\Delta_i=s'_iP_{\text{pub}}$. Then transmit $\langle i,P'_i,\Delta_i\rangle$ to the PKG secretly.

(2) The PKG checks the validity of P'_i by $e(P'_i,P_{\text{pub}})=e(\Delta_i,P)$. If it holds, PKG changes P_i to P'_i publicly and renews $S^{(i)}_{ID}$ in Key-Gen-2 accordingly. Else PKG refuses Γ_i's request.

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

4 Security analysis

4.1 Definition of the adversary

To give the formal definition of the IB-ThDec scheme, we need to define adversaries for it. Since we use a (t,n) threshold scheme, it's reasonable to assume that at most $t-1$ out of n decryption servers will be corrupted by the adversary A. Assume $\{\Gamma_i\}_{i\in S,|S|=t-1}$ be the set of corrupted decryption servers. A can learn the secret keys of them, get all broadcasting messages and decryption shares of uncorrupted ones. Furthermore, the A can make the corrupted decryption servers to deviate from the protocol in an unrestricted fashion. The actions that A against the IB-ThDec are listed below:

(1) Γ_i's private key and key share extraction queries. For $i\in S$, A is allowed to make request for Γ_i's private key s_i. Additionally, A can ask for $P_{ID},(S^{(j)}_{ID},V^{(k)}_{ID})$ of given identity ID for $j\in S,1\leq k\leq n$.

(2) Complete decryption key extraction queries. A is allowed to query on an identity ID's complete decryption key. However, it is not reasonable for A to extract the complete decryption key of the selected challenge identity ID_{ch}.

(3) Decryption queries. A is allowed to query on chosen ciphertexts, to get the plaintexts and decryption shares from uncorrupted decryption servers. A natural restriction is A cannot query on the challenge ciphertext.

(4) Replace Γ_i's public key. Since Γ_i's public key $P_i=s_iP(i=1,2,\cdots,n)$ is not associated with Γ_i's identity, A can choose $s'_i\in Z^*_q$ and try to replace P_i by $P'_i=s'_iP$ for corrupted Γ_i.

4.2 Security model

In this section, we give the formal security models of the Basic-IB-ThDec scheme and the full IB-ThDec scheme.

Definition 3(IND-IDTH-CPA). The basic identity based (t,n) threshold decryption scheme Basic-IB-ThDec is secure against ID-based threshold chosen-plaintext attacks (denoted by IND-IDTH-CPA) if no polynomially bounded adversary B has a non-negligible advantage in the following game:

Init. B corrupts a fixed subset of $t-1$ decryption servers. Then the challenger gives the resulting public parameters to B.

Key extraction queries1. B's challenger runs Share-Key-Gen:

(1) Given ID, the challenger returns the complete decryption key S_{ID} of ID, upon ID's complete decryption key extraction queries.

(2) The challenger gives $P_{ID},P^{(j)}_{ID}(1\leq j\leq n)$ and the private key shares $d_{ID,i}(i\in S)$ of the corrupted decryption servers to B. However, the private key shares of uncorrupted decryption servers are kept secret from B.

Challenge. B chooses two equal length plaintexts (M_0,M_1) and an identity ID^* to be challenged on. Then give them to the challenger. The challenger responds with $C^*=(U,V)=Basic\text{-}IB\text{-}Encrypt(M_b,cp,ID^*)$ for a random $b\in\{0,1\}$.

Key extraction queries2. B issues more key extraction queries as in key extraction queries1, except the complete decryption key of ID^*.

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

Such an adversary B is called an IND-IDTH-CPA adversary[5]. B's advantage is defined to be:

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

Definition 4(IND-IDTH-CCA2). The (t,n) threshold decryption scheme from ID-based cryptosystem is secure against ID-based threshold adaptive chosen-ciphertext attacks (denoted by IND-IDTH-CCA2) if no polynomially bounded adversary A has a non-negligible advantage in the following game:

Init. The adversary A 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 A, including the public key P_i of $\Gamma_i(1\leq i\leq n)$.

Γ_i's private key extraction queries: Given S, the challenger generates $t-1$ corrupted decryption servers' private key $s_i(i\in S)$. Return (i,s_i) to A.

Key extraction queries1. On an identity ID, A performs a number of queries adaptively:

(1) Complete decryption key extraction queries. The challenger generates complete decryption key $d_{ID}=asH_1(ID)$ and $P_{ID}=sH_1(ID)$. Send d_{ID}, P_{ID} to A.

(2) Key share queries. The challenger returns $(S_{ID}^{(j)},V_{ID}^{(i)})$ for $j\in S$ and $i\in\{1,2,\cdots,n\}$.

(3) Replace Γ_i's public key. For $i\in S$, suppose the request is to replace the public key of Γ_i with $\langle P_i'=s_i'P,\Delta_i=s_i'P_{\text{pub}}\rangle$. After receiving $\langle ID,P_i',\Delta_i\rangle$, the challenger accepts A's request, and renews $S_{ID}^{(i)}$ associated with P_i' and ID.

Decryption queries1. A arbitrarily feeds the challenger ciphertexts, and then obtains plaintexts and decryption shares of uncorrupted decryption servers.

Challenge. A 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 A 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 $C^*=Full\text{-}IB\text{-}Encryption(M_{b'},ID_{ch},cp)$. Return C^* to A.

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

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

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

Such an adversary A is called an IND-IDTH-CCA2 adversary. A's advantage is defined to be:

$$Adv(A)=|2Pr[b''=b']-1|.$$

4.3 Security proof

Theorem 1. Let H_1,H_2,H_3,H_4 be random oracles. Then IB-ThDec is an IND-IDTH-CCA2 secure scheme assuming the quadruple Decision BDH problem (4-DBDH) is hard in groups generated by Setup. Concretely, suppose there is an adversary A that has an advantages ε against the IB-ThDec. If A makes at most q_E complete decryption key extraction queries and at most q_{H_1} hash queries to H_1, then there is an algorithm that solves the 4-DBDH problem in groups generated by Setup with advantages at least $\varepsilon''=\varepsilon/2e(1+q_E)q_{H_1}$.

Proof of Theorem 1. The proof is by reduction. First we introduce three lemmas. Then Theorem 1 follows by combining Lemma 1, Lemma 2, and Lemma 3. □

Lemma 1. If H_1,H_2,H_3,H_4 are random oracles, let A be an IND-IDTH-CCA2 adversary that has an advantage ε against IB-ThDec. Suppose A makes q_E complete decryption key extraction queries. Then there is an IND-IDTH-CPA adversary B that has an advantage at least $\varepsilon'=\varepsilon/e(1+q_E)$ against Basic-IB-ThDec.

Lemma 2. Let H_1,H_2 be random oracles, and let B be an IND-IDTH-CPA adversary that has the advantage ε' against the Basic-IB-ThDec. Suppose B makes q_{H_1} hash queries to H_1. Then there is an IND-CPA adversary C that has the advantage at least $\varepsilon''=\varepsilon'/q_{H_1}$ against BasicPub-CL-PKE.

Lemma 3. Let H_2 be a random oracle, and let C be an IND-CPA adversary that has the advantage ε'' against the BasicPub-CL-PKE. Then there is an algorithm E that solves the 4-DBDH problem with advantage at least $\varepsilon''/2$.

Proof of Lemma 1. B works by interacting with A in an IND-IDTH-CCA2 game as follows:

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

Setup. Algorithms B starts by receiving Basic-IB-ThDec's public parameters $cp=\{q,l,G_1,G_2,e,P,P_{\text{pub}},H_1,H_2,X,Y\}$ from his challenger, and gives A the IB-ThDec system parameters $\{q,l,G_1,G_2,e,P,P_{\text{pub}},X,Y,\{H_i\}_{1\leq i\leq 5},\{P_j\}_{1\leq j\leq n}\}$, where

(1) $q,l,G_1,G_2,e,P,P_{\text{pub}},X,Y$ are taken from cp.

(2) H_1,H_2,H_3,H_4 are random oracles controlled by B. H_5 is a one-way hash function.

(3) Randomly pick $m_i\in Z_q^*(1\leq i\leq n)$. Keep m_i in secret, return $P_i=m_iP$ to A.

***H_1*-queries**. A can query H_1 at any time. Let ID_i be the i-th distinct identity asked by A, B flips a $coin_i(coin_i\in\{0,1\},Pr[coin_i=0]=\delta)$ and maintains a list L_1 of tuples $\langle coin_i,ID_i,b_i,Q_{ID_i}\rangle$, where:

(1) If $coin_i=0$, then B picks b_i at random from Z_q^*. Output $Q_{ID_i}=b_iP$, add the tuple $\langle coin_i=0,ID_i,b_i,Q_{ID_i}\rangle$ to L_1.

(2) If $coin_i=1$, then B forwards ID_i to B's challenger and returns the answer Q_{ID_i} to A. Add

the tuple $\langle coin_i=1, ID_i, \perp, Q_{ID_i}\rangle$ to L_1.

***H_2*-queries.** A can issue H_2 queries at any time. B forwards it to B's challenger and returns the answer to A.

***H_3*-queries.** When A queries H_3 at a distinct point (U_i, V_i, W_i), B defines $\overline{P}_i$ at that point by choosing $T_i \in Z_q^*$ uniquely, and maintains an initially empty list L_3 of tuples $\langle T_i, (U_i, V_i, W_i)\rangle$. Return $\overline{P}_i = T_i X$.

***H_4*-queries.** When A queries H_4 at a distinct point $(\overline{P}_i, \overline{U}_i, \overline{W}_i)$, B chooses $e_i \in Z_q^*$ uniquely at random as the answer and maintains an initially empty list L_4 of tuples $\langle e_i, (\overline{P}_i, \overline{U}_i, \overline{W}_i)\rangle$.

***Γ_i*'s private key extraction queries.** To answer A's private key extraction queries on $t-1$ corrupted decryption servers, B returns $m_i (i \in S)$ to A.

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

(1) Complete private key extraction queries.

If $coin_i = 0$, B outputs complete decryption key of ID_i as $d_{ID_i} = b_i Y$.

Else if $coin_i = 1$, B terminates the game and outputs "Abort".

(2) Key share queries.

If $coin_i = 0$, B randomly chooses a polynomial of degree $t-1$ over Z_q^*: $f_{ID_i}(x) = b_i + \sum_{j=1}^{t-1} a_j x^j$. Then compute $S_{ID_i}^{(k)} = f_{ID_i}(k) Y + m_k P_{\text{pub}}$, $V_{ID_i}^{(k)} = e(f_{ID_i}(k) P_{\text{pub}}, X)$ for $1 \le k \le n$. Return $(S_{ID_i}^{(k)}, V_{ID_i}^{(k)})(1 \le k \le n)$ to A. Add $<ID_i, \{S_{ID_i}^{(j)}\}_{t \le j \le n}>$ to the list L_{ks}. And return the verification key $P_{ID_i} = b_i P_{\text{pub}}$.

Else if $coin_i = 1$,

B forwards ID_i and S to its challenger and gets $P_{ID_i}, \{d_{ID_i,j}\}_{j \in S}, \{P_{ID_i}^{(k)}\}_{1 \le k \le n}$. Return P_{ID_i}.

For $j \in S$, return $S_{ID_i}^{(j)} = d_{ID_i,j} + m_j P_{\text{pub}}$.

For $1 \le k \le n$, return $V_{ID_i}^{(k)} = e(P_{ID_i}^{(k)}, X)$.

It's easy to prove that $S_{ID_i}^{(j)}, V_{ID_i}^{(k)}$ ($j \in S$ and $k \in \{1, 2, \cdots, n\}$) can pass the validity test of key shares. (When $coin_i = 1$, we make use of the fact that $\{d_{ID_i,j}\}_{j \in S}$, $\{P_{ID_i}^{(k)}\}_{1 \le k \le n}$ can pass the validity test of B).

(3) Replace Γ_i'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 $\langle P_j', \Delta_j\rangle$ to B (It should be a valid pair, i. e. $e(P_j', P_{\text{pub}}) = e(\Delta_j, P)$. B accepts A's request and computes partial keys upon ID_i as:

If $coin_i = 0$, $S_{ID_i}^{(j)} = f_{ID_i}(j) Y + \Delta_j$,

If $coin_i = 1$, $S_{ID_i}^{(j)} = d_{ID_i,j} + \Delta_j$.

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

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

(1) First, B computes $W_i = f_i P - e_i U_i$, and searches the L_3 list for a tuple $\langle T_i, (U_i, V_i, W_i)\rangle$ containing (U_i, V_i, W_i). If it is nonexistent, B returns "Invalid Ciphertext".

(2) Else B searches L_4 for a tuple $\langle e_i, (\overline{P}_i, \overline{U}_i, \overline{W}_i)\rangle$, where $\overline{P}_i = T_i X$, $\overline{W}_i = f_i \overline{P}_i - e_i \overline{U}_i$. If B fails, return "Invalid Ciphertext".

(3) Else, M_i and $\delta_{ID_j, C_i}^l (l = t, t+1, \cdots, n)$ can be computed as follows:

If $coin_j = 0$, when A queries B at C_i, B performs the following:

Since $e(d_{ID_j}, U_i) = e(b_j Y, U_i)$, output $M_i = V_i \oplus H_2(e(b_j Y, U_i))$.

With $V_{ID_j}^{(l)}$, $S_{ID_j}^{(l)}$ and $k_{ID_j}^l = e(S_{ID_j}^{(l)} - m_l P_{\text{pub}}, U_i)$, B can readily run Proof-Log to output the decryption share $\delta_{ID_j, C_i}^l = \{l, k_{ID_j}^l, h_l, \lambda_l\}$.

If $coin_j = 1$, although B cannot get r_i from $U_i = r_i P$, he can assume $\overline{U}_i = r_i \overline{P}_i$ and simulate the decryption of C_i as:

Since $e(d_{ID_j}, U_i) = e(asQ_{ID_j}, r_i P) = e(sQ_{ID_j}, r_i (T_i X))^{\frac{1}{T_i}}$, output $M_i = V_i \oplus H_2(e(P_{ID_j}, \frac{1}{T_i} \overline{U}_i))$.

With $\{d_{ID_j,k}\}_{1 \le k \le t-1}$, $k_{ID_j}^l$ can be computed as $k_{ID_j}^l = e(S_{ID_j}^{(l)} - m_l P_{\text{pub}}, U_i) = e(L_{l0}^{S'} P_{ID_j}, \overline{U}_i)^{\frac{1}{T_i}} \cdot \prod_{m=1}^{t-1} e(L_{lm}^{S'} d_{ID_j}, k, U_i)$. Where $L_{lm'}^{S'}$ is the Lagrange coefficient with respect to $S' = \{0\} \cup S$, for $0 \le m' \le t-1$. Run Proof-Log, and return $\delta_{ID_j, C_i}^l = \{l, k_{ID_j}^l, h_l, \lambda_l\}$.

Challenge. Adversary A issues two equal length plaintexts (M_0, M_1) and an identity ID_{ch} which it decided to be challenged on. B responds as follows:

(1) If $coin_{ch} = 0$ then B terminates the game and reports "Abort".

(2) If $coin_{ch} = 1$ then B forwards (M_0, M_1), ID_{ch} to its challenger. When receiving the Basic-IB-ThDec ciphertext: $C' = (U', V') = Basic\text{-}IB\text{-}Encrypt(M_{b'}, cp, ID_{ch})(b' \in \{0, 1\})$, B simply chooses $e^*, f^*, l^* \in Z_q^*$ and sets $V^* = V'$, $U^* = U'$, $\overline{P}^* = l^* P$, $\overline{U}^* = l^* U^*$, $W^* = f^* P - e^* U$, $\overline{W}^* = f^* \overline{P}^* - e^* \overline{U}^*$ Then B backpatches and defines the challenge ciphertext $C^* = (V^*, U^*, \overline{U}^*, e^*, f^*)$. Then C^* is the IB-ThDec encryption of $M_{b'}$ for a random

$b' \in \{0,1\}$ under the public key ID_{ch} as required.

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

Decryption queries2. A issues more decryption queries, 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, A outputs a guess $b'' \in \{0,1\}$. B outputs b'' as its guess for b'.

Analysis.

(1) Suppose B is given a ciphertext $C \neq C^*$, where $C=\langle (V,U,\overline{U},e,f),W,\overline{W}\rangle$. If C can pass the ciphertext-validity-test, and $\langle V,U,W\rangle \neq \langle V^*,U^*,W^*\rangle$, then A must has queried H_3 at the point $\langle U,V,W\rangle$. So B has $\overline{P}=H_3(U,V,W)=TX$ with $T \in Z_q^*$. Thus, B can exactly decrypt C as described above.

Else if C can pass the ciphertext-validity-test while $\langle V,U,W\rangle = \langle V^*,U^*,W^*\rangle$ and $\langle \overline{U},e,f\rangle \neq \langle \overline{U}^*,e^*,f^*\rangle$, then $\overline{P}=\overline{P}^*$. Set $W=t'P$, $\overline{W}=t''\overline{P}$, $U=r'P$ and $\overline{U}=r''\overline{P}$ with $r' \neq r''$ (if $r'=r''$, then $t'=t''$ and $C=C^*$). Because 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 B, this happens with the probability at most $1/q$. It's negligible when q is large enough.

Note that if we additionally check whether $e(P,\overline{U})=e(\overline{P},U)$ in the ciphertext-validity-test, making use of the decision Diffie-Hellman problem is polynomially solvable in $\langle G_1,G_2,e\rangle$, 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. Thus, our scheme cancels pairing computations in the ciphertext-validity-test phase. Comparing with [3], our scheme is more efficient.

(2) If B does not abort during the game, then A's view is identical to its view in the real attack. Because B's responses to all hash queries are uniformly and independently distributed as in the real attack, and all responses to A's request can pass validity test unless B aborts in the game. Furthermore, $e(d_{ID_{ch}},U')=e(d_{ID_{ch}},U^*)$. Thus, by the definition of A, we have $|2Pr(b''=b')-1|=Adv(A)=\varepsilon$. Let H denote the event that B does not abort in the game, then the advantage of B is $\varepsilon' > \varepsilon \cdot Pr[H]$. We name the event that A made a complete private key extraction queries on ID_i with $coin_i=1$ at some points as E_1, and the event that A chose ID_{ch} with $coin_{ch}=0$ as E_2. If $Pr[coin=0]=\delta$, then $Pr[H]=Pr[\neg E_1 \wedge \neg E_2]=\delta^{q_E}(1-\delta)$. This value is maximized when $\delta_{opt}=1-1/(q_E+1)$[1]. Using δ_{opt}, $Pr[H]$ is at least $1/e(1+q_E)$. This shows that B's advantage is at least $\varepsilon'=\varepsilon/e(1+q_E)$. This finishes the proof. □

Proof of Lemma 2. This proof techniques is similar to that used in [5, Theorem 3.1], with modifications to handle the adversary B who gets X,Y from C by setting $X=X_A$, $Y=Y_A$, where X_A,Y_A are taken from C's challenger. Additionally, B can never query on the challenge identity ID^* for its complete decryption key. □

Proof of Lemma 3. This proof techniques is modeled on the proof of [1, Lemma 4.3]. E takes $(P,aP,bP,cP,abP,acP,bcP,dP,D)$ as input, and sets $P_{pub}=cP$, $Q_A=bP$, $X_A=aP$, $Y_A=acP$, $D_A=bcP$, $D'_A=abP$. In challenge phase, C submits two equal length plaintexts (M_0,M_1). E returns the challenge ciphertext $C^*=(U,V)=(dP, M_{b'} \oplus R)$ with $R \in \{0,1\}^l$, set $H_2(D)=R$. In guess phase, C outputs a guess $b'' \in \{0,1\}$. If $b''=b'$, then E outputs 1 meaning $D=e(P,P)^{abcd}$. Otherwise, return 0 meaning $D \neq e(P,P)^{abcd}$. If $D=e(P,P)^{abcd}$, then C's view is identical to its view in the real game, and $Pr[b'=b'']=\frac{1}{2}\pm\frac{\varepsilon''}{2}$. Else if $D \neq e(P,P)^{abcd}$ then D is uniform and independent in G_2, and the challenge ciphertext C^* is independent of b', that is $Pr[b'=b'']=1/2$. Therefore, the advantage of E is $\left|\left(\frac{1}{2}\pm\frac{\varepsilon''}{2}\right)-\frac{1}{2}\right|=\varepsilon''/2$. This finishes the proof. □

5 Further discussion

The IB-ThDec can readily be converted into a fully secure certificateless threshold decryption scheme, in which the communication users are required to choose their own public key pair $\langle X_A, Y_A\rangle$ as in CL-PKE. And each decryption server's private key share is given by the communication users. The security reduction is similar to IB-ThDec.

Another potential application of IB-ThDec is an ID-based dynamic threshold decryption scheme (IB-D-ThDec)[15]. Intuitively, each decryption server 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 PKG can update the master key s, or add/remove any decryption servers without changing its private information. That is, the decryption servers and PKG may communicate via broadcast channels after the key generation process has

taken place, and the secret channels are never needed after that. Obviously, this scheme is very practical. However, we can only prove its security in the selective-identity model[2]. That is, the adversary commits the target identity ID^* ahead of time, and the challenge will not answer any key extraction queries about ID^*. The security reduction is also similar to IB-ThDec. Virtually, the security proof of the corresponding IB-D-ThDec scheme has been implied in the proof of Theorem 1, under selective-identity model.

6 Conclusions

In this paper, we propose an ID-based threshold decryption scheme that can resist against both the adaptive chosen-ciphertext attack, and against the active attacker who can modify the behaviors of decryption servers. At last, we illustrate the applications of IB-ThDec.

References

1 Boneh D., Franklin M.. Identity-based encryption from the weil pairing. SIAM Journal on Computing, 2003, 32(3): 586～615

2 Boneh D., Boyen X., Halevi S.. Chosen ciphertext secure public key threshold encryption without random oracles. In: Proceedings of the RSA-CT '06, San Jose, USA, 2006, 226～243

3 Baek J., Zheng Y.. Identity-based threshold decryption. In: Proceedings of the PKC'04, Singapore, 2004, 262～276

4 Chai Z. C., Cao Z. F., Lu R. X.. Id-based threshold decryption without random oracles and its application in key escrow. In: Proceedings of the Inforsec 2004, Shanghai, 2004, 119～124

5 Libert B., Quisquater J.. Efficient revocation and threshold pairing based cryptosystems. In: Proceedings of the PODC2003, Boston, USA, 2003, 163～171

6 Desmedt Y., Frankel Y.. Threshold cryptosystems. In: Proceeding of the CRYPTO'89, New York, USA, 1989, 307～315

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

8 Fouque P., Pointcheval D.. Threshold cryptosystems secure against chosen-ciphertext attacks. In: Proceedings of the Asiacrypt 2001, Australia, 2001, 351～368

9 Shoup V., Gennaro R.. Securing threshold cryptosystems against chosen ciphertext attack. In: Advances in Cryptology-Eurocrypt'98, Finland, 1998, 1～16

10 Bellare M., Desai A., Pointcheval D., Rogaway P.. Relations among notions of security for public-key encryption schemes. In: Proceedings of the Crypto'98, Santa Barbara, California, USA, 1998, 26～45

11 Blum M., Santis A. D., Micali S., Persiano G.. Non-interactive zero knowledge. SIAM Journal on Computing, 1991, 6(4): 1084～1118

12 Lim C., Lee P.. Another method for attaining security against adaptively chosen ciphertext attack. In: Proceedings of the Crypto'93, New York, USA, 1993, 420～434

13 Bellare M., Rogaway P.. Random oracles are practical — A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, USA, 1993, 62～73

14 Al-Riyami S. S., Paterson K. G.. Certificateless public key cryptography. In: Proceedings of the Asiacrypt 2003, Taiwan, 2003, 452～473

15 Sun H. M., Shieh S. P.. Construction of dynamic threshold schemes. Electronics Letters, 1994, 30(24): 2023～2025

LONG Yu, born in 1980, Ph. D. candidate. Her research interests include information theory and modern cryptography etc.

CHEN Ke-Fei, born in 1959, Ph. D., professor, Ph. D. supervisor. His main research areas are classical and modern cryptography, theory and technology of network security, etc.

HONG Xuan, born in 1982, Ph. D. candidate. Her research interests include signature and the provably security.

Background

With the development of networks, distributed cryptography has received a lot of attention in modern cryptographic research. Threshold cryptosystems provide security to distributed applications, and can avoid single point of failure in a network system. But it's very difficult to design a provably secure threshold decryption scheme. Combining ID-based cryptosystem with threshold decryption has recently been considered in many important works. However almost all the existing works which treated threshold decryption in the context of ID-based cryptography are only provably chosen plaintext secure. The authors propose a fully secure ID-based threshold decryption scheme and give the formal security proof. This scheme can resist against adaptive chosen ciphertext attack, and can be readily used to construct certificateless threshold decryption systems and ID-based dynamic threshold decryption systems. This work is supported by NSFC under the grants No. 60303026, 60473020, 60573030 and the Foundation of NLMC under the grant No. 51436040405JW0304.