A Provable Security Scheme of ID-Based Threshold Decryption

Wang Xue-Guang and Chai Zhen-Chuan

School of Information Science and Technology, East China University of Political Science and Law, 555 Long Yuan Road shanghai 201620, China
Samsung electronics R&D center
wangxueguang@ecupl.edu.cn

Abstract. This paper presents an ID-based threshold decryption scheme and proves that it is selective chosen ciphertext secure without random oracles based on solving decisional $(t,q,\varepsilon)-BDHI$ problem assumption.

Keywords: provable security, ID based cryptography, Threshold decryption.

1 Introduction

In general public key certification system, user's public key and ID information are bound by certificates. The fact that authenticity of certificates need be verified before using public key results in increasing the amount of computation. For simplifying certificate management and decreasing additional calculating costs, Shamir [1] proposed identity based(ID-based) public key cryptography in 1984, which let users select their unambiguous information(such as E-mail, telephone number, etc.) as public key, then a trusted Private Key Generator(PKG) generates private key and distributes them to users through secret channel.

But, ID-based encryption (IBE) scheme was proposed using bilinear pairing by Boneh and Franklin [2] until 2001, which could be proved to be secure against adaptive chosen ciphertext attack by random oracle model [3-5]. However, this proof is controversial under assumption of random oracle model, because so-called random oracle does not exist in reality after all, i.e., those schemes will be unsafe after they are put in practice.

The fist IBE scheme without random oracles was proposed by Boneh and Boyen [6] in 2004, which was proved to be secure against selective-ID chosen plaintext attack without random oracle model. We need consider three levels for security of public key system in practice: chosen plaintext security, non-adaptive chosen ciphertext security and adaptive chosen ciphertext security. The last has highest security and is also main research direction at present.

This paper suggests an ID-based threshold decryption scheme based on works of Boneh and Boyen, which can be proved to be secure without random oracles. It was validated that it has ID-based adaptive chosen ciphertext security.

M. Sorell (Ed.): e-Forensics 2009, LNICST 8, pp. 122 – 129, 2009.

2 Model of ID-Based Threshold Decryption Scheme

There are several roles in the ID-based threshold decryption scheme. A trusted PKG takes charge of generating user private key and threshold private key, also running in the beginning phrase of system, including selecting public parameters, such as bilinear pairing and its corresponding group, etc. A cluster of *n* decryption servers is denoted by Γ_i(i=1,…,n), which has a public ID, generates private keys, executes encryption and decryption, verifies algorithms, and so on.

This scheme includes six algorithms as follow.

① **Start:** this algorithm run by PKG outputs master key - *mkey*, and system's public parameter - *cp*. *cp* includes the group selected by PKG, bilinear pairing and so on. *cp* is public but *mkey* is secretly saved by PKG.

② **KeyGen(*mkey*, *ID*, *t*, *n*):** given PKG's *mkey*, user's ID, number of all decryption - *n* and threshold *t*, this algorithm returns *n* key slices d_{ID_i}, *i* = *1*,*2*, …, *n*, which corresponds to public key ID.

③ **KeyVer:** this algorithm returns *n* public verification message v_i, *i* = *1*,*2*, …, *n*, which can be used by decryption server Γ_i for verifying private key slices d_{ID_i}, then PKG secretly sends d_{ID_i} to Γ_i but each message v_i will be public.

④ **Encrypt(*cp*, *ID*, *M*):** given ID and plaintext *M*, this algorithm returns ciphertext denoted by *c*.

⑤ **Decrypt(*cp*, d_{ID_i}, *c*):** given ciphertext *c* and key slice d_{ID_i}, this algorithm returns corresponding decryption slice, denoted by δ_i, or returns error information that indicates *c* is invalid ciphertext. At the same time, it verifies decryption slices.

⑥ **Combin(*cp*, $\{\delta_i\}_{i\in\phi}$, *c*):** given *t* decryption slices $\{\delta_i\}_{i\in\phi}$, this algorithm combines many decryption slices into plaintext *M*, $\phi \subset \{1, ..., n\}$ and $|\phi| = t$.

3 IND-CCA Security

Given a public key cipher scheme $E = (K, J, D)$, *K* as secret key generation algorithm, *J* as encryption algorithm, *D* as decryption algorithm, consider the procedure as follow. Here, take *Q* as a assaulter, *S* as a challenger.

Step 1. Assaulter *Q* sends ciphertext *c* to *S*. *S* obtains plaintext *M* by decrypting *c* and sends *M* to *Q*. In this phase, *Q* can freely select satisfying ciphertext and go to next step.

Step 2. Assaulter *Q* selects two equal-length messages M_0 and M_1, sends them to challenger *S*.

Step 3. Challenger *S* randomly selects a bit-value $\beta \in \{0,1\}$ then calculates c^* and sends it to *Q*. Here,

$$c^* = \begin{cases} E_{pk}(M_0) & \beta = 0 \\ E_{pk}(M_1) & \beta = 1 \end{cases}$$, where *pk* denotes user's public key.

Step 4. After receiving c^*, Q can continue to request decryption services like described in Step 1, but can not question for c^*.

Step 5. Q needs make a guess $\beta' \in \{0,1\}$ about β.

If probability advance that assaulter Q successfully attacks decryption algorithm is $Adv_Q = \left|\Pr[0 \leftarrow Q(c^* = E_{pk}(M_0))] - \Pr[0 \leftarrow Q(c^* = E_{pk}(M_1))]\right|$ and Adv_Q is a negligible value about ε, then E is secure for indistinguishable adaptive chosen ciphertext attack, i.e., IND-CCA security.

4 Building ID-Based Threshold Decryption Scheme

The security of the scheme is built on hard problem of bilinear Diffie-Hellman inversion [6].

Given multiplicative group G and G_1 of the same prime order p, p is a large prime number. And g is the generator of G. The mapping e: $G \times G \rightarrow G_1$ is a computable bilinear pairing. Let plaintexts be all in G_1 and IDs as public keys in Z_p^*.

The process of building six algorithms of the scheme is as follow.

① **Start:** Select x, y, $z \in_R Z_p^*$ and compute $X = g^x$, $Y = g^y$, and $Z = g^z$. Public parameter *cp* and master key *mkey* of PKG are respectively:

$$cp = (g, X, Y, Z), \quad mkey = (x, y, z)$$

② **KeyGen(*mkey*, *ID*, *t*, *n*):** To generate n secret key slices for the public key ID, the PKG:

a) randomly selects a polynomial over Z_p^*: $F(u) = z + \sum_{i=1}^{t-1} u^i a_i$, $a_i \in Z_p^*$

b) selects random number $r_i \in_R Z_p^*$, computes $K_i = g^{F(i)/(ID + x + r_i y)}$ and outputs secret key slice $d_{ID_i} = (r_i, K_i)$.

③ **KeyVer:** Generate verification message v_i, $v_i = e(g,g)^{F(i)}$, $i = 1, \ldots, n$.

④ **Encrypt(*cp*, *ID*, *M*):** To encrypt plaintext $M \in G_1$ using public key ID$\in Z_p^*$, select random number $s \in Z_p^*$ and calculate ciphertext using the expression

$$C = (g^{s \cdot ID} X^s, Y^s, e(g,Z)^s \cdot M)$$

Note that the value of pairing $e(g, Z)$ can be pre-computed and stored for following computation in order to save time.

⑤ **Decrypt(*cp*, d_{ID_i}, *c*):** For computing decryption slice δ_i of ciphertext $C = (A, B, C)$, decryption server Γ_i, using its key slice $d_{ID_i} = (r_i, K_i)$, gets that

$\delta_i = e(AB^{r_i}, K_i)$ because of

$\delta_i = e(AB^{r_i}, K_i) = e(g^{s(ID+x+r_i y)}, g^{F(i)/(ID)+x+r_i y}) = e(g,g)^{sF(i)}$.

⑥ **Combin(*cp*, $\{\delta_i\}_{i\in\phi}$, c):** In order to recovery original plaintext M, a proxy server collects t decryption slices $\delta_i \in G_1$ and calculates M as follow

$$C / \prod_{i\in\phi} \delta_i^{L_i^0} = M .$$

Here, $\phi \subset \{1,...,n\}$, $|\phi| = t$, and $L_i^x = \prod_{j\in\phi, j\neq i} \frac{x-j}{i-j}$.

The validity of this computation can be obtained by employing Lagrange interpolation:

$$C / \prod_{i\in\phi} \delta_i^{L_i^0} = C / \prod_{i\in\phi} e(g,g)^{sF(i)L_i^0} = C / e(g,g)^{s\sum_{i\in\phi} L_i^0 F(i)} = C / e(g,g)^{sF(0)} = C / e(g,g)^{s\cdot z} = M$$

After running algorithm KeyGen, PKG secretly distributes key slices d_{ID_i} to decryption server Γ_i, then open all verification message v_i. Γ_i can check the authenticity of $d_{ID_i} = (r_i, K_i)$ by verifying the following equation after receiving d_{ID_i},

$$\prod_{i\in\phi} v_i^{L_i^0} = e(g, Z) \text{ and } e(g^{ID} XY^{r_i}, K_i) = v_i .$$

5 Security Proof of ID-Based Threshold Decryption Scheme without Random Oracles

Use reduction to absurdity to prove the security of ID-based threshold decryption scheme. First, assume that threshold decryption scheme is not secure and there is an assaulter who can attack the scheme by probability advance ε under defined attack model. And assume decisional $(t,q,\varepsilon) - BDHI$ problem is hard. Then, construct an algorithm to solve the decisional $(t,q,\varepsilon) - BDHI$ problem. Its result is contrary to the assumption of hard problem. So the threshold decryption scheme is secure.

5.1 Decisional $q - BDHI$ Problem

Decisional $q - BDHI$ problem [6]:

Given $(q+1)$-tuple $(g, g^x, g^{(x^2)}, \cdots, g^{(x^q)}) \in (G^*)^{q+1}$ and $T \in G_1^*$, decide whether equation $T = e(g,g)^{1/x}$ is correct or not.

The advance of algorithm *A* solves decisional $q-BDHI$ problem is defined as:

$$Adv(A)=\left|\Pr\left[A(g,g^{x},\cdots,g^{(x^{q})},e(g,g)^{1/x})=0\right]-\Pr\left[A(g,g^{x},\cdots,g^{(x^{q})},T=0\right]\right|,$$

where the probability is computed through randomly selecting x on Z_p^*, T on G_1^*, and algorithm *A*.

If any algorithm can not solve computational/decisional $q-BDHI$ problem in time t with a probability advance which is ε at least, then a computational/decisional $q-BDHI$ problem is said to be hard.

5.2 Construct Algorithm *S*

The purpose of *S* is to solve an instance of decisional *BDHI* problem, i.e., given an input $(g,g^{\alpha},g^{\alpha^2},\cdots,g^{\alpha^q},T)\in(G^{q+1})\times G_1^*$ (*S* doesn't know α), decides whether T is equal to $e(g,g)^{1/\alpha}$, if yes, output 1, otherwise 0.

Considering *Q* as assaulter, *S* as challenger, before interacting with *Q*, *S* needs prepare for a generator $h\in G^*$, and corresponding q-1 pairs of two-tuple $(w_i,h^{1/(\alpha+w_i)})$ (*S* doesn't know α). These parameters are as follow:

① Randomly select $w_1,\cdots,w_{q-1}\in Z_p^*$,

let $f(\theta)=\prod_{i=1}^{q-1}(\theta+w_i)=\prod_{i=0}^{q-1}(c_i\theta^i)$ such that $c_0\neq 0$;

② Compute $h=\prod_{i=0}^{q-1}(g^{\alpha^i})^{c_i}=g^{f(\alpha)}$ and $u=\prod_{i=1}^{q}(g^{\alpha^i})^{c_{i-1}}=g^{\alpha f(\alpha)}$. It is easy to know that $u=h^{\alpha}$ and $h\neq 1$, because h = 1 means that there is a $w_j=\alpha$ and *S* can solve decisional *BDHI* problem directly;

③ Let $f_i(\theta)=f(\theta)/(\theta+w_i)=\sum_{i=0}^{q-2}d_i\theta^i$

and $h^{1/(\alpha+w_i)}=g^{f_i(\alpha)}=\prod_{i=0}^{q-2}(g^{\alpha^i})^{d_i}$.

S computes:

$$T_h=T^{(c_0^2)}\cdot T_0,$$

where $T_0=\prod_{k=0}^{q-2}e(g^{(\alpha^k)},g)^{c_0c_{k+1}}\prod_{i=1}^{q-1}\prod_{j=0}^{q-2}e(g^{(\alpha^i)},g^{(\alpha^j)})^{c_ic_{j+1}}$.

Here, if $T=e(g,g)^{1/\alpha}$, then $T_h=e(g^{f(\alpha)},g^{f(\alpha)})^{1/\alpha}=e(h,h)^{1/\alpha}$. Otherwise, T_h only is a random value in $G_1\setminus\{T_0\}$, because T randomly distributes on G_1^*.

5.3 Security Verification

The interaction process between Q and S is as follow:

Select attack ID: Q selects an attack object $ID^* \in Z_p^*$.

Initialization: S executes the following steps.

① Selects random number $a,b \in Z_p^*$ such that $ab = ID^*$;

② Selects random number $z \in Z_p^*$, computes $X = u^{-a}h^{-ab} = h^{-a(\alpha+b)}$, $Y = u = h^{\alpha}$ and $Z = h^z$;

③ Publishes $cp = (h, X, Y, Z)$.

During above computation, the master key *mkey* is implicitly defined as $mkey = (x, y, z) = (-a(\alpha + b), \alpha, z)$. Though S doesn't know x and y, it knows $x + ay = -ab = -ID^*$.

Phrase 1: Q successfully compromises t-1 out of n decryption servers. Without loss of generality, suppose compromised servers are $\Gamma_1, \cdots, \Gamma_{t-1}$.

Phrase 2: Q starts a series of private key queries and decryption queries.

① Private key query about $ID \neq ID^*$: In order to provide n valid key slices and n verification messages, S operates according to the following for Q's query:

a) Randomly selects a polynomial over Z_p^*:

$$F(u) = z + \sum_{i=1}^{t-1} u^i f_i,\ f_i \in Z_p^*;$$

b) Fetches n unused two-tuples $(w_i, h^{1/(\alpha+w_i)})$, without loss of generality, suppose these tuples' subscripts denote by $i = 1, \cdots, n$. Let $h_i = h^{1/(\alpha+w_i)}$.

c) Computes $r_i = a + \dfrac{ID - ab}{w_i}$, returned secret key slices and verification messages are as follow:

$$d_{ID_i} = (r_i, h_i^{F(i)/(r_i - a)}) \text{ and } v_i = (g, g)^{F(i)},\ i = 1, \cdots, n.$$

It is easy to know, $h_i^{F(i)/(r_i-a)} = h^{F(i)/(r_i-a)(\alpha-w_i)} = h^{F(i)/ID + x + r_i y}$. d_{ID_i} is valid secret key slice because w_i is randomly selected by S. So also is $r_i = a + \dfrac{ID - ab}{w_i}$ from view of Q.

② Private key query about ID^*: In order to provide t-1 valid key slices and n verification messages, S operates according to the following for Q's query:

a) Randomly selects $r_i \in Z_p^*$ and $K_i \in G$, $i = 1,\cdots,t-1$;

b) Computes $v_i = \begin{cases} e(g^{ID^*} X \cdot Y^{r_i}, K_i), & i = 1,\cdots,t-1 \\ e(g,g)^{L_0^i z} \prod_{k=1}^{t-1} v_k^{L_k^i} & i = 1,\cdots,n \end{cases}$, where $L_k^x = \prod_{j=0, j\neq k}^{t-1} \frac{x-j}{k-j}$;

c) Returned *t*-1 secret key slices and *n* verification messages are as follow:

$$d_{ID_i} = (r_i, K_i) \quad i = 1,\cdots,t-1 \text{ and } v_i \quad i = 1,\cdots,n.$$

In fact, *S* implicitly selects a polynomial $F(u)$, such that $e(g^{ID^*} X \cdot Y^{r_i}, K_i) = e(g,g)^{F(i)}$ and $F(0) = z$ for $i = 1,\cdots,t-1$.

Challenge: Once assaulter *Q* thinks phrase 2 can be over, *Q* will output two equal bit-length plaintexts (M_1, M_2). After received those plaintexts, *S* randomly selects a bit $\beta \in \{0,1\}$ and $l \in Z_p^*$, computes challenge ciphertext $c = (h^{-al}, h^l, T_h^{zl} \cdot M_\beta)$ and then sends it to *Q*.

Here, if $T_h = e(h,h)^{1/\alpha}$, then *c* is a valid ciphertext on M_β. Because:

let $s = l/\alpha$ (*l* is randomly selected, so *s* also is random distribution on Z_p^*), then

$$h^{-al} = h^{-a\alpha(l/\alpha)} = h^{(x+ab)(l/\alpha)} = h^{(x+ID^*)(l/\alpha)} = h^{sID^*} \cdot X^s$$

$$h^l = Y^{l/\alpha} = Y^s$$

$$T_h^{zl} = e(h,h)^{zl/\alpha} = e(h,h)^{zs} = e(h,Z)^s$$

If T_h only is a random number on $G_1 \setminus \{T_0\}$, then *c* is completely independent of bit β from view of *Q*.

Phrase 3: According to its requirement, *Q* continues to send private key queries like phrase 2, whose time qs is limited by $qs < \lfloor q/n \rfloor$. The challenger still replies *Q*'s queries like phrase 2.

Hypothesize: *Q* output its guess $\beta' \in \{0,1\}$ for β. If $\beta' = \beta$, then *S* returns 1, which means $T = e(g,g)^{1/\alpha}$. Otherwise *S* returns 0, which means $T \neq e(g,g)^{1/\alpha}$.

During above interaction process, if input *T* satisfies $T = e(g,g)^{1/\alpha}$, then the probability advance of *Q* satisfies $Adv = |\Pr[\beta = \beta'] - 1/2| > \varepsilon$, which results in the advance that *S* solves hard problems satisfies $\Pr\left[S(g, g^x, \cdots, g^{(x^q)}, e(g,g)^{1/x}) = 1 \right] > 1/2 + \varepsilon$. If $T = P \neq e(g,g)^{1/\alpha}$, then the probability advance of *Q* satisfies

$\Pr\left[S(g, g^{x}, \cdots, g^{(x^q)}, e(g,g)^{1/x}) = 1\right] > 1/2 + \varepsilon$, because ciphertext is also random number. In this situation, the advance that *S* solves hard problems only is a guess, i.e., $\Pr\left[S(g, g^{x}, \cdots, g^{(x^q)}, P) = 1\right] = 1/2$.

In summary, the probability advance that algorithm S solves decisional $q-BDHI$ problem is

$$Adv_{S} = \left|\Pr\left[S(g, g^{x}, \cdots, g^{(x^q)}, e(g,g)^{1/x}) = 1\right] - \Pr\left[S(g, g^{x}, \cdots, g^{(x^q)}, P) = 1\right]\right| \geq \left|(1/2 + \varepsilon) - 1/2\right| = \varepsilon$$

This is contrary to the assumption.

According to above proof and IND-CCA's definition, the ID-based threshold decryption scheme has IND-CCA security, that is, it is secure for indistinguishable adaptive chosen ciphertext attack.

6 Conclusions

Through reviewing related researches, this paper proposes an ID-based threshold decryption scheme built on Boneh and Boyen's works. After defining IND-CCA and solving decisional $(t, q, \varepsilon) - BDHI$ hard problem, we proved the scheme is secure for selective-ID adaptive chosen ciphertext attack without random oracles.

Acknowledgements. This work is supported by Chinese Society and Science Foundation under Grant No. 06BFX051.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
2. Boneh, D., Franklin, M.: Identity based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the first ACM Conference on Computer and Communication Security, ACM Conference, pp. 62–73 (1993)
4. Bellare, M., Boldyreva, A., Palacio, A.: An uninstantiable random oracle model scheme for a hybrid-encryption problem. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 171–188. Springer, Heidelberg (2004)
5. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The noncommitting encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)