Commun. Korean Math. Soc. **24** (2009), No. 1, pp. 153–159

# CRYPTANALYSIS OF A DYNAMIC THRESHOLD DECRYPTION SCHEME

KITAE KIM, SEONGAN LIM, IKKWON YIE, AND KYUNGHEE KIM

ABSTRACT. In this paper, we analyze a dynamic threshold decryption scheme proposed by Long et al. It was claimed that the scheme allows to renew the master key, or to add/remove or update a decryption server, without leaking any information on the master key and changing other decryption server's secret keys. We present an attack to Long et al's scheme by using the fact that it renews a decryption server's secret key without changing other decryption server's secret shares.

## 1. Introduction

The concept of identity based (ID-based) public key was proposed by Shamir in 1984 [6] to solve the difficulty of public key certificate management in public key infrastructure. Following the first practical ID-based encryption scheme by Boneh and Franklin in 2001 [2], several ID-based cryptographic schemes have been suggested-encryption scheme, signature scheme, threshold decryption scheme etc.

Threshold decryption can be used to decentralize the power of decryption. One possible application of threshold decryption in ID-based encryption is to share the secret key of an organization among special members, that is called the set of decryption servers, and encrypted data using the identity of the organization could only be decrypted when the number of participated decryption servers reaches a fixed threshold value [3]. Dynamic threshold decryption in ID-based encryption deals with the case when the set of decryption servers are dynamic. A dynamic system allows to add a new decryption server or remove a decryption server as well as renew each decryption server's secret key. In general, if one consider ID-based threshold decryption in dynamic setting, all of the decryption servers' shares must be updated whenever either the set of decryption servers or the public/secret key pair of one decryption server has changed and it causes communicational and computational overhead. It opens the denial of service attack due to such overhead.

Received May 23, 2007.
2000 *Mathematics Subject Classification.* 94A60.
*Key words and phrases.* bilinear pairing, cryptography, dynamic threshold decryption.

In [5], a dynamic threshold decryption scheme was proposed. The authors in [5] claimed that their threshold decryption scheme efficiently solves such overhead of dynamic setting. In particular, they claimed that one can renew one decryption server's secret key without changing other decryption servers' secret share.

In this paper, we show that the PKG's secret polynomial should be renewed whenever any decryption server needs to update its secret key by constructing a specific attack. This implies that all of the shares of the decryption servers should be renewed whenever a secret share of one decryption server has to be renewed. Hence we conclude that the dynamic threshold decryption scheme proposed in [5] is suitable only in a very limited dynamic setting that will be described later on.

## 2. Review of the dynamic threshold decryption scheme from pairing in [5]

Y. Long and K.-F. Chen proposed a dynamic $(t, n)$ threshold decryption scheme from pairing. We review their scheme.

### 2.1. Bilinear pairing

For two multiplicative cyclic groups $G$, $G_1$ of a prime order $p$, we assume that there is an efficiently computable function $e : G \times G \to G_1$, with following properties:

(i) Bilinearity : for all $A, B \in G$ and $a, b \in \mathbb{Z}_p^*$,

$$e(A^a, B^b) = e(A, B)^{ab}.$$

(ii) Non-degeneracy : There exists $A \in G$ such that $e(A, A)$ has order $p$ in $G_1$. In other words, $e(A, A)$ is a generator of $G_1$, whereas $A$ is a generator of $G$.

In the above case, we call $e : G \times G \to G_1$ as a bilinear map or pairing and the group $G$ as a bilinear group. We assume that the discrete logarithm problem is hard in both groups $G$ and $G_1$.

### 2.2. The scheme

Long et al's dynamic threshold decryption scheme requires a trusted authority private key generator (PKG)

**1) Set up** To setup the system, PKG

- Chooses two bilinear groups $G, G_1$ of the same prime order $p$ with the bilinear pairing $e : G \times G \to G_1$, and $g$ a generator of $G$.
- Selects $x, y \in Z_p^*$ at random and computes $X = g^x, Y = g^y$.
- Sets the public key $P_{pub}$ of PKG as $P_{pub} = (g, X, Y)$ and the secret master key $mskey$ as $mskey = (x, y)$.
- Opens an ID, such as an hash value of telephone number.

**2) Key generation of $n$ Decryption servers** Assume that $n$ decryption servers are $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$.

- Each $\Gamma_i$ chooses a secret key $s_i \in Z_p^*$ and compute the public key $P_i = g^{s_i}$ and publishes $P_i$.
- PKG picks randomly a polynomial $f(\alpha)$ of degree $t-1$ over $Z_p$:

$$f(\alpha) = y + \sum_{i=1}^{t-1} b_i \alpha^i, \text{ where } b_{t-1} \in Z_p^*$$

- PKG computes

$$\begin{aligned} k_i &= g^{\frac{f(i)}{(ID+x)}} \cdot P_i^{-y}, \\ v_i &= e(g,g)^{f(i)}, \end{aligned}$$

and publishes $k_i$ and $v_i$. For notational convenience, we call $g^{\frac{f(i)}{(ID+x)}}$ is the secret share for the decryption server $\Gamma_i$. We understand $k_i$ as the ElGamal encryption of $g^{\frac{f(i)}{(ID+x)}}$ using the agreed key $P_i^{-y} = Y^{-s_i}$ between PKG and the decryption server $\Gamma_i$.(We note that it was given as $k_i = g^{\frac{f(i)}{g^{(ID+x)} P_i^y}}, v_i = e(g,g)^{f(i)}$ in [5], but $k_i$ doesn't make sense in this form and it should be corrected as above.)

- $\Gamma_i$ examines the validity of $k_i$ by checking

$$e(g^{ID} \cdot X, k_i \cdot Y^{s_i}) = v_i.$$

(We note that the equation to be checked was given as $e(g^{ID} \cdot X, (k_i)^{Y^{s_i}}) = v_i$ in [5], but we think they made typing error here and it should be corrected as above.)

**3) Encryption** To encrypt a message $M \in G_1$, one

- Picks a random $s \in Z_p^*$.
- Outputs the ciphertext

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

**4) $\Gamma_i$'s Sub-decryption** For a given ciphertext $C = (A, B)$, $\Gamma_i$ calculates:

$$\begin{aligned} \delta_i &= e(A, k_i \cdot Y^{s_i}) \\ &= e(g,g)^{S \cdot (ID+x) \cdot \frac{f(i)}{(ID+x)}} \\ &= e(g,g)^{S \cdot f(i)} \end{aligned}$$

(We note that it was given as $\delta_i = e(A, g^{Y_i^s})^{k_i}$ in [5], but we think they made typing error and it should be corrected as above.)

**5) Decryption** Without loss of generality, assume that $\Gamma_1, \Gamma_2, \ldots, \Gamma_t$ are $t$ decryption servers who want to decrypt the ciphertext $C = (A, B)$. From the

computed $\Gamma_i$'s sub-decryption $\delta_i$, a dealer(one of the servers) collects $\delta_1, \ldots, \delta_t$ and computes

$$\begin{aligned}\Delta &= \prod_{j=1}^{t}(\delta_j)^{\prod_{i=1,i\neq j}^{k}\frac{-i}{j-i}} \\ &= e(g,g)^{s\cdot(\sum_{j=1}^{t} f(j)\prod_{i=1,i\neq j}^{t}\frac{-i}{j-i})} \\ &= e(g,g)^{s\cdot(f(0))} \\ &= e(g,g)^{s\cdot y}.\end{aligned}$$

Thus the plaintext $M$ can be obtained from the ciphertext $C = (A, B)$ by computing $M = \frac{B}{e(g,g)^{s\cdot y}}$.

### 2.3. Updating private keys and shares in [5]

In order to deal with the dynamic status of decryption servers, the authors in [5] discussed followings.

**Case 1.** In the case when $y$ is renewed, the PKG renews the polynomial and refreshes the corresponding public information for each decryption server.

**Case 2.** In the case when a new decryption server is added with his public key and secret key pair $(P_{n+1} = g^{s_{n+1}}, s_{n+1})$, the PKG computes

$$k_{n+1} = g^{\frac{f(n+1)}{(ID+x)}} \cdot P_{n+1}^{-y}, v_{n+1} = e(g,g)^{f(n+1)},$$

and then publishes $k_{n+1}, v_{n+1}$.

**Case 3.** In the case when a decryption server to be removed, then PKG renews the polynomial and refreshes the corresponding public information for each decryption server.

**Case 4.** In the case when a decryption server $\Gamma_i$'s secret key $s_i$ to be renewed to $s_i'$ with the corresponding public key $P_i'$, then PKG refreshes $\Gamma_i$'s public information to $k_i' = g^{\frac{f(i)}{(ID+x)}} \cdot P_i'^{-y}$.

We note that either in the cases when a new decryption server is added or the secret key (hence the public key) of a decryption server is renewed, the PKG keeps the same polynomial to renew the shares of the corresponding servers. This is important advantage of the dynamic threshold decryption scheme in [5]. But we shall show that one can mount an attack to the scheme because the scheme keeps the same polynomial in Case 4 above.

## 3. Cryptanalysis of the dynamic threshold decryption scheme in [5]

In this section we show that the Long et al's dynamic $(t, n)$ threshold decryption scheme is not secure if it keeps the same polynomial $f$ of degree $t-1$ in the case when a decryption server's secret key to be renewed. More precisely, a decryption server, say $\Gamma_1$, renews his secret key in a malicious way, then a set of colluded $t-1$ decryption servers including $\Gamma_1$ can decrypt any ciphertext.

Suppose we have colluding $t-1$ decryption servers $\Gamma_1, \ldots, \Gamma_{t-1}$. Now we show how these $t-1$ decryption servers obtain the decryption key $g^{\frac{y}{ID+x}}$.

(1) For each $j = 1, \ldots, t-1$, $\Gamma_j$ computes $g^{\frac{f(j)}{ID+x}}$ :
- Since each $\Gamma_j$ knows its secret key $s_j$ and hence he can compute $g^{\frac{f(j)}{ID+x}} = k_j Y^{s_j}$ from the public information $k_j$.

(2) A malicious server, say $\Gamma_1$, renews its public key to let PKG refresh the corresponding public information to be $k_1' = g^{\frac{f(1)}{ID+x}} P_t{}^{-y}$ :
- Note that $\Gamma_1$ knows a public information $P_t = g^{s_t}$ of $\Gamma_t$ (without knowing the secret key $s_t$ of $\Gamma_t$).
- $\Gamma_1$ picks a random $r \in \mathbb{Z}_p^*$, computes $P_1' = P_t^{-r}$ and publishes $P_1'$.
- Since, in PKG's view, $P_1'$ is a random element of the group $G$, PKG will refresh the public information of $\Gamma_1$ to be $k_1' = g^{\frac{f(1)}{ID+x}} P_1'{}^{-y}$. In fact, we have $k_1' = g^{\frac{f(1)}{ID+x}} \left(P_t^{-r}\right)^{-y} = g^{\frac{f(1)}{ID+x}} P_t{}^{ry}$.
- We note that the decryption server $\Gamma_1$ has the information

$$g^{\frac{f(1)}{ID+x}}, r, k_1, k_1', k_t, P_1, P_1'.$$

(3) $\Gamma_1$ can compute $g^{\frac{f(t)}{ID+x}}$ from information $g^{\frac{f(1)}{ID+x}}, r, k_1, k_1', k_t, P_1, P_1'$ in the following manner.
- Compute $b = k_1' \cdot \left\{ g^{\frac{f(1)}{ID+x}} \right\}^{-1} = P_t^{ry} = g^{rs_t y}$.
- Compute $c = b^{r^{-1}} = g^{s_t y} = P_t^y$.
- Compute $d = k_t \cdot c = \left( g^{\frac{f(t)}{ID+x}} P_t^{-y} \right) (P_t^y) = g^{\frac{f(t)}{ID+x}}$.

(4) Finally, $t-1$ decryption servers $\Gamma_1, \ldots, \Gamma_{t-1}$ can compute $g^{\frac{y}{ID+x}}$ as follows
- Collect $g^{\frac{f(j)}{ID+x}}$ for $j = 1, \ldots, t$ from the servers $\Gamma_1, \ldots, \Gamma_{t-1}$.
- Calculate the Lagrange coefficients $\Delta_i = \prod_{j=1, j\neq i}^{t} \frac{-i}{j-i}$.
- Compute $\prod_{j=1}^{t} (g^{\frac{f(j)}{ID+x}})^{\Delta_j} = g^{\frac{\sum_{j=1}^{t} f(j)\Delta_j}{ID+x}} = g^{\frac{y}{ID+x}}$.

Hence we have $g^{\frac{y}{ID+x}}$ and then for any ciphertext $C = (C_1, C_2) = \left( \left(g^{ID} X\right)^s, M \cdot e(g, Y^s) \right)$ of a message $M \in G_1$ , we can decrypt as follows:

$$\frac{C_2}{e(C_1, g^{\frac{y}{ID+x}})} = \frac{Me(g, g^y)^s}{e(g^{(ID+x)s}, g^{\frac{y}{ID+x}})} = \frac{Me(g^s, g^y)}{e(g^s, g^y)} = M.$$

*Remark* 3.1.

(1) If a decryption server, say $\Gamma_1$, can update its secret key at $t-1$ times, then $\Gamma_1$ alone obtains $g^{\frac{y}{ID+x}}$. That is to say, the system is not a threshold decryption scheme. :
- Suppose $t$ public keys $P_i$ of $t$ decryption server $\Gamma_i, i = 1, \ldots, t$ are known.

- First, $\Gamma_1$ calculates $g^{\frac{f(1)}{ID+x}}$ from his secret key $s_1$ and public information $k_1$ from PKG.
- $\Gamma_1$ renews his public key as $P_{1,2} = P_2^{-r_2}$, obtains the corresponding public information $k_{1,2}$ and then computes $g^{\frac{f(2)}{ID+x}}$ as in Step 2, and Step 3 above.
- Next, $\Gamma_1$ renews his public key as $P_{1,3} = P_3^{-r_3}$,and computes $g^{\frac{f(3)}{ID+x}}$ from the corresponding public information as in Step 2, and Step 3 above.
- Proceeding the same process $t-1$ times, $\Gamma_1$ has

$$g^{\frac{f(1)}{ID+x}}, g^{\frac{f(2)}{ID+x}}, \ldots, g^{\frac{f(t)}{ID+x}}$$

and finally it computes $g^{\frac{y}{ID+x}}$ by employing Lagrange interpolation. And thus $\Gamma_1$ can decrypt any ciphertext all by himself.

(2) The attacks described above can be prevented if PKG renews the polynomial $f$ and refreshes all information of decryption servers each time that a server renews the secret $s_i$ of itself. In such a case, the system has no advantage over regular threshold decryption schemes in dynamic setting.

## 4. Conclusion

We analyzed Long et al's dynamic threshold decryption scheme, and presented an attack to Long et al's scheme by using the fact that it renews a decryption server's secret key without changing other decryption server's secret shares. In order to prevent such attack, the system has to renew the secret shares of all decryption servers whenever one of the decryption server renews its secret key. Thus we see that Long et al's dynamic threshold decryption scheme is meaningful only if each decryption server has a fixed public key and secret key pair during the life time of the periodic private key $y$, and adding decryption servers can be frequent but removing decryption servers are very rare.

## References

[1] D. Boneh and X. Boyen, *Efficienty Selective-ID Secure Identity based encryption scheme without random oracles*, Advances in Cryptology-Eurocrypt 04, LNCS, Vol. 3027, pp. 223–238, 2004.

[2] D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairing*, Advances in Cryptology-Crypto 2001, LNCS Vol. 2139, pp. 213–229, Springer-Verlag, 2001.

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

[4] B. Libert and J. J. Quisquater, *Efficient revocation and threshold pairing based cryptosystems*, PODC 2003, ACM press, pp. 163–171, 2003.

[5] Y. Long and K.-F. Chen, *Construction of dynamic threshold decryption scheme from pairing*, International Journal of Network Security, Vol.2, No. 2, pp. 111–113, 2006.

[6] A. Shamir, *Identity based cryptosystems and signature schemes*, Advances in Cryptology-Crypto 1984, LNCS Vol. 196, pp. 47–53, Springer-Verlag, 1984.

KITAE KIM
DEPARTMENT OF MATHEMATICS
INHA UNIVERSITY
INCHEON 402-751, KOREA
*E-mail address*: ktkim@inha.ac.kr

SEONGAN LIM
DEPARTMENT OF MATHEMATICS
INHA UNIVERSITY
INCHEON 402-751, KOREA
*E-mail address*: sglim@inha.ac.kr

IKKWON YIE
DEPARTMENT OF MATHEMATICS
INHA UNIVERSITY
INCHEON 402-751, KOREA
*E-mail address*: ikyie@inha.ac.kr

KYUNGHEE KIM
DEPARTMENT OF MATHEMATICS
YONSEI UNIVERSITY
WONJU 220-710, KOREA
*E-mail address*: khlkim@yonsei.ac.kr