10-2015

A note on the security of KHL scheme

Jian WENG

Yunlei ZHAO

DENG, Robert H.
Singapore Management University, robertdeng@smu.edu.sg

Shengli LIU

Yanjiang YANG

See next page for additional authors

DOI: https://doi.org/10.1016/j.tcs.2015.07.051

Citation

WENG, Jian; ZHAO, Yunlei; DENG, Robert H.; LIU, Shengli; YANG, Yanjiang; and SAKURAI, Kouichi. A note on the security of KHL scheme. (2015). *Theoretical Computer Science*. 602, 1-6. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/2820

Author

Jian WENG; Yunlei ZHAO; DENG, Robert H.; Shengli LIU; Yanjiang YANG; and Kouichi SAKURAI

Published in Theoretical Computer Science, 2015 October, Volume 602, Pages 1-6
https://doi.org/10.1016/j.tcs.2015.07.051

A note on the security of KHL scheme

Jian Weng[a,b,*], Yunlei Zhao[c], Robert H. Deng[d], Shengli Liu[e], Yanjiang Yang[f], Kouichi Sakurai[b]

[a] *Department of Computer Science, Jinan University, Guangzhou, China*
[b] *Department of Informatics, Kyushu University, Fukuoka, Japan*
[c] *Software School, Fudan University, Shanghai, China*
[d] *School of Information Systems, Singapore Management University, Singapore*
[e] *Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China*
[f] *Institute for Infocomm Research (I2R), A*STAR, Singapore*

ARTICLE INFO

Article history:
Received 16 April 2013
Received in revised form 24 July 2015
Accepted 28 July 2015
Available online 31 July 2015
Communicated by X. Deng

Keywords:
Cryptography
Traitor-tracing
Broadcast encryption
Chosen-ciphertext attacks

ABSTRACT

A public key trace and revoke scheme combines the functionality of broadcast encryption with the capability of traitor tracing. In Asiacrypt 2003, Kim, Hwang and Lee proposed a public key trace and revoke scheme (referred to as KHL scheme), and gave the security proof to support that their scheme is z-resilient against adaptive chosen-ciphertext attacks, in which the adversary is allowed to adaptively issue decryption queries as well as adaptively corrupt up to z users. In the passed ten years, KHL scheme has been believed as one of the most efficient public key trace and revoke schemes with z-resilience against adaptive chosen-ciphertext attacks *under the well-studied DDH assumption*. However, in this paper, by giving a concrete attack, we indicate that KHL scheme is actually *not* secure against adaptive chosen-ciphertexts, *even without corruption of any user*. We then identify the flaws in the security proof for KHL-scheme, and discuss the consequences of the attack.

1. Introduction

A broadcast encryption system [1,2] allows the sender to encrypt a message for a dynamically changing set S of authorized users. Any user in S can use his private key to decrypt the broadcast, while users outside of S cannot obtain any information about the contents of the broadcast. Broadcast encryption has found many practical applications, such as pay-TV, satellite-based commerce, and distribution of copyrighted materials, etc. In the past two decades, broadcast encryption has attracted great interests, and many broadcast encryption schemes have been proposed, e.g. [3–8].

The risk for broadcast encryption here is that users may collude and produce a pirate decoder, which is not registered with the authority but can decrypt the broadcast. Traitor tracing schemes, introduced in [9], enable the authority to trace users who collude to produce the pirate decoder.

Trace and revoke systems [10,11] provide the functionalities of both broadcast encryption and traitor tracing. When pirate decoding happens, a traitor-tracing algorithm [12,13] is able to identify at least one user who has colluded to build the pirate decoder. After the traitors are traced, or some users' private keys are compromised, the system will revoke them by excluding them from the receiving set. Some trace and revoke systems concentrate on the centralized setting, where

[*] Corresponding author at: Department of Computer Science, Jinan University, Huangpu Avenue West 601, Guangzhou, Guangdong, 510632, China, Tel.: +86-15920484703.
E-mail address: cryptjweng@gmail.com (J. Weng).

only the trusted center (i.e., the entity who generates all the secret keys) can send messages to the receivers. In contrast, in public key setting, the center also generates a fixed public key which allows any entity to play the role of the sender.

In PKC 2003, Dodis and Fazio [14] proposed the first public key trace and revoke scheme (referred to as DF scheme) with z-resilience against adaptive chosen-ciphertext attacks (CCA2), where the adversary is allowed to corrupt up to z users and adaptively issue decryption queries. In Asiacrypt 2003, based on DF scheme, Kim, Hwang and Lee proposed an efficient public key trace and revoke scheme, and gave a security proof to claim that their scheme is z-resilient against CCA2 attacks, under the decisional Diffie–Hellman (DDH) assumption. In some sense, KHL-scheme can be viewed as a simplified version of DF scheme, and enjoys much better efficiency advantages over DF scheme. In the passed ten years, KHL scheme has been viewed as one of the most efficient public key trace and revoke schemes with CCA2 security *under the well-studied DDH assumption*.

In this paper, we shall make a careful observation on KHL scheme. By giving a concrete attack, we indicate that KHL-scheme is actually *not* CCA2 secure *even without corruption of any user*. We then identify the flaws in the security proof of KHL-scheme, and discuss the consequences of the attack.

2. Framework of broadcast encryption

In [15], Kim et al. merely concentrated on the construction of public key broadcast encryption scheme, and mentioned that, by slightly modifying standard tracing algorithm from previous schemes (e.g., [16]), their scheme can be a fully functional trace and revoke scheme. Thus, in this section, we shall also only review the framework for broadcast encryption.

2.1. Definition of broadcast encryption

In a public key broadcast encryption scheme BE, a session key s is encrypted and broadcasted with the symmetric encryption of the "actual" message. Generally, the encryption of s is called the *enabling block*. A public key broadcast encryption scheme consists of four poly-time algorithms ($\mathsf{KeyGen}, \mathsf{Reg}, \mathsf{Enc}, \mathsf{Dec}$), where:

$\mathsf{KeyGen}(1^\kappa, z)$: The key generation algorithm, is a probabilistic algorithm used by the center to set up all the parameters of the scheme. Taking as input a security parameter 1^κ and a revocation threshold z (i.e., the maximum number of users that can be revoked), this algorithm returns the public key PK and the master secret key $\mathrm{SK}_{\mathsf{BE}}$.

$\mathsf{Reg}(\mathrm{SK}_{\mathsf{BE}}, i)$: The registration algorithm, is a probabilistic algorithm used by the center to generate the secret key needed to construct a new decoder each time a new user subscribes to the system. This algorithm takes as input the master key $\mathrm{SK}_{\mathsf{BE}}$ and a (new) index i associated with the user, returns the user's secret key SK_i.

$\mathsf{Enc}(\mathrm{PK}, \mathcal{R}, s)$: The encryption algorithm, is a probabilistic algorithm used to encapsulate a given session key K. This algorithm takes as input the public key PK, the session key s and a set $\mathcal{R}$ of revoked users (with $|\mathcal{R}| \le z$), and returns the enabling block T.

$\mathsf{Dec}(\mathrm{SK}_i, T)$: The decryption algorithm, is a deterministic algorithm that takes as input the secret key SK_i of user i and the enabling block T, and returns the session key s that was encapsulated within T if i was a legitimate user when T was constructed, or the special symbol $\perp$.

2.2. Security model

We review the adaptive chosen-ciphertext security model for broadcast encryption as defined in [14,15]. Concretely, the adaptive chosen-ciphertext Security is defined using the following game between an attacker $\mathcal{A}$ and a challenger (both given the security parameter 1^λ and the revocation threshold z as input)):

Stage 1: The challenger runs $(\mathrm{PK}, \mathrm{SK}_{\mathsf{BE}}) \leftarrow \mathsf{KeyGen}(1^\kappa, z)$ and gives the public key PK to adversary $\mathcal{A}$.

Stage 2: Adversary $\mathcal{A}$ adaptively makes a series of queries to the *User Corruption Oracle* $\mathsf{Cor}_{\mathrm{SK}_{\mathsf{BE}}}(\cdot)$ and *Decryption Oracle* $\mathsf{D}_{\mathrm{SK}_{\mathsf{BE}}}(\cdot,\cdot)$. The user corruption oracle receives as input the index i of the user to be corrupted, computes $\mathrm{SK}_i \leftarrow \mathsf{Reg}(\mathrm{SK}_{\mathsf{BE}}, i)$ and returns the user secret key SK_i to adversary $\mathcal{A}$. Note that this user corruption oracle can be called adaptively for at most z times during the whole game. Let us say that at the end of this stage the set $\mathcal{R}$ of at most z users is corrupted. The decryption oracle, taking as input a pair $\langle i, T\rangle$ (where i is the index of some users and T is an enabling block), returns the result of $\mathsf{Dec}(\mathrm{SK}_i, T)$, where SK_i is user i's secret key.

Stage 3: The adversary submits two session keys s_0 and s_1. The challenger picks a random $\sigma \in \{0,1\}$, computes $T^* \leftarrow \mathsf{Enc}(\mathrm{PK}, \mathcal{R}, s_\sigma)$, and responds with the target enabling block T^*.

Stage 4: The adversary continues to issue decryption oracle queries, subject only to the restriction that a submitted enabling block T is not identical to T^* (and, of course, up to z users can be corrupted).

Stage 5: The adversary outputs a guess $\sigma^* \in \{0,1\}$.

We define the advantage of $\mathcal{A}$ in attacking scheme BE as $Adv^{\mathrm{CCA2}}_{\mathcal{A},\mathsf{BE}} = \left|\Pr[\sigma^* = \sigma] - \frac{1}{2}\right|$.

Definition 1. We say that a public key broadcast scheme BE is z-resilient against CCA2 attacks, if for all probabilistic, polynomial time adversary $\mathcal{A}$, his advantage $Adv_{\mathcal{A},\mathsf{BE}}^{\mathsf{CCA2}}$ is negligible in λ.

3. Cryptanalysis of KHL scheme

As in DF scheme, Lagrange interpolation (in the exponent) plays an important role in the construction of KHL scheme. We shall first review some necessary facts about Lagrange interpolation. Then we shall review KHL scheme, and present a chosen-ciphertext attack against KHL scheme.

3.1. Lagrange interpolation

Let $f(x)=\sum_{t=0}^{z} a_t x^t$ be a z-degree polynomial over $\mathbb{Z}_q$. Then given $z+1$ pairwise distinct points $\{(x_t, f(x_t))\}_{t=0,1,\cdots,z}$, one can reconstruct this polynomial $f(x)$ as

$$f(x)=\sum_{t=0}^{z}(f(x_t)\cdot\lambda_t(x)),$$

where $\lambda_t(x)=\prod_{0\le k\ne t\le z}\frac{x_k-x}{x_k-x_t}$.

We define the Lagrange interpolation operator as

$$\mathsf{LI}\left[x_0,\cdots,x_z;\, f(x_0),\cdots,f(x_z)\right](x)=\sum_{t=0}^{z}(f(x_t)\cdot\lambda_t(x)).$$

Next, we consider a cyclic multiplicative group $\mathbb{G}$ with prime order q and a generator g. Let $F_t=g^{f(x_t)}$, $0\le t\le z$, where $x_t\in\mathbb{Z}_q$. Then we define the Lagrange interpolation operator in the exponent as

$$\begin{aligned}\mathsf{ExpLI}\left[x_0,\cdots,x_z;\, F_0,\cdots,F_z\right](x)&=g^{\mathsf{LI}[x_0,\cdots,x_z;\, f(x_0),\cdots,f(x_z)](x)}\\&=\prod_{t=0}^{z}g^{f(x_t)\cdot\lambda_t(x)}=\prod_{t=0}^{z}F_t^{\lambda_t(x)}.\end{aligned}$$

Note that the following equality holds:

$$\mathsf{ExpLI}\left[x_0,\cdots,x_t;\, f(x_0)^r,\cdots,f(x_t)^r\right](x)=\left(\mathsf{ExpLI}\left[x_0,\cdots,x_t;\, f(x_0),\cdots,f(x_t)\right](x)\right)^r.$$

3.2. Review of KHL scheme

KHL scheme is specified by the following algorithms:

$\mathsf{KeyGen}(1^\kappa, z)$: Given the security parameter 1^λ, this key generation algorithm first chooses two random generators $g_1, g_2 \leftarrow_r \mathbb{G}$, where $\mathbb{G}$ is a group with prime order q (here q is a large prime such that $p=2q+1$ is also a large prime). Next, it chooses $x_1, x_2, y_1, y_2 \leftarrow_r \mathbb{Z}_q$ and z-degree polynomials $X_1(\xi), X_2(\xi), Y_1(\xi), Y_2(\xi)$ over $\mathbb{Z}_q$ such that $X_1(0)=x_1, X_2(0)=x_2, Y_1(0)=y_1, Y_2(0)=y_2$. It also chooses two z-degree polynomials $Z_1(\xi), Z_2(\xi)$ over $\mathbb{Z}_q$, and computes $c=g_1^{x_1}g_2^{x_2}, d=g_1^{y_1}g_2^{y_2}$. Next, for $t=0,\cdots,z$, it computes $h_t=g_1^{Z_1(t)}g_2^{Z_2(t)}, x_{1,t}=g_1^{X_1(t)}, x_{2,t}=g_2^{X_2(t)}, y_{1,t}=g_1^{Y_1(t)}, y_{2,t}=g_2^{Y_2(t)}$. Finally, it chooses a hash function H from a family $\mathcal{F}$ of collision resistant hash functions, and outputs $(\mathrm{PK}, \mathrm{SK}_{\mathsf{BE}})$, where $\mathrm{PK}=(p, q, g_1, g_2, c, d, x_{1,0},\cdots,x_{1,z}, x_{2,0},\cdots,x_{2,z}, y_{1,0},\cdots,y_{1,z}, y_{2,0},\cdots, y_{2,z}, h_0,\cdots,h_z)$ and $\mathrm{SK}_{\mathsf{BE}}=(X_1, X_2, Y_1, Y_2, Z_1, Z_2)$.

$\mathsf{Reg}(\mathrm{SK}_{\mathsf{BE}}, i)$: Each time a new user $i>z$ decides to subscribe to the system, the center provides him with a decoder box containing the secret key $\mathrm{SK}_i=(i, X_1(i), X_2(i), Y_1(i), Y_2(i), Z_1(i), Z_2(i))$.

$\mathsf{Encrypt}(\mathrm{PK}, \mathcal{R}, s)$: On input the public key PK, a set $\mathcal{R}=\{j_1,\cdots,j_z\}$ of z revoked users, and the session key s, this algorithm proceeds as follows:

1. Pick $r_1\leftarrow_r\mathbb{Z}_q$, and compute $u_1=g_1^{r_1}, u_2=g_2^{r_1}$.
2. For $t=0,\cdots,z$, compute $H_t=h_t^{r_1}$.
3. For $t=1,\cdots,z$, compute $H_{j_t}=\mathsf{ExpLI}(0,\cdots,z;\, H_0,\cdots,H_z)(j_t)$.
4. Compute $S=s\cdot H_0$, $\alpha=H(S,u_1,u_2)$ and $C=(cd^\alpha)^{r_1}$.
5. For $t=0,\cdots,z$, compute $C_t=\left((x_{1,t}x_{2,t})(y_{1,t}y_{2,t})^\alpha\right)^{r_1}$.
6. For $t=1,\cdots,z$, compute $C_{j_t}=\mathsf{ExpLI}(0,\cdots,z;\, C_0,\cdots,C_z)(j_t)$ and $F_{j_t}=H_{j_t}\frac{C}{C_{j_t}}$.
7. Finally, output $T=\left(S, u_1, u_2, C, (j_1, F_{j_1}),\cdots,(j_z, F_{j_z})\right)$.

Note that as mentioned in [15], F_{j_t} can be viewed as $g_1^{Q(j_t)}$ where $Q(\xi)$ is a z-degree polynomial over $\mathbb{Z}_q$.

$\mathsf{Dec}(\mathrm{SK}_i, \mathtt{CT})$: On input the secret key SK_i of user i and the enabling block T, this decryption algorithm first parses $T = \big(S, u_1, u_2, C, (j_1, F_{j_1}), \cdots, (j_t, F_{j_t})\big)$, and then executes the following steps:

1. Compute $\alpha = H(S, u_1, u_2)$.
2. Compute $C_i = u_1^{X_1(i)+Y_1(i)\alpha} u_2^{X_2(i)+Y_2(i)\alpha}$.
3. Compute $H_i = u_1^{Z_1(i)} u_2^{Z_2(i)}$
4. Compute $F_i = H_i \frac{C}{C_i}$.
5. Finally, output $s \leftarrow \frac{S}{\mathsf{ExpLI}(i, j_1, \cdots, j_z; F_i, F_{j_1}, \cdots, F_{j_z})(0)}$.

Remark 1. As explained in [15], KHL scheme can be viewed as an improved version derived from the CCA2 secure Dodis–Fazio scheme (denoted by DF-CCA2 scheme) [14]. In DF-CCA2 scheme, a one-time message authentication code (MAC) is additionally used to check the validity of the ciphertext (i.e., enabling block). To shorten the ciphertext size, KHL scheme does not use MACs to explicitly check the validity of the ciphertext. As explained [15], KHL scheme borrows the idea of the modified Cramer–Shoup scheme in [17–19], i.e., the decrypting algorithm will output the original plaintext if the ciphertext is valid, and otherwise a random value independent of the original plaintext. However, as we shall indicate in the next subsection, if the ciphertext in KHL scheme is deliberately modified, the output of the decryption algorithm is *not* a random value independent of the original plaintext. The insecurity of KHL scheme exactly lies in this fact.

3.3. Attack

Before presenting our concrete attack, we here explain some necessary facts about the Lagrange interpolation operator in the exponent. As mentioned before, $z+1$ pairwise distinct points $\{(j_t, F_{j_t})\}_{t=0}^{z}$ determines a z-degree polynomial $f(x)$, and the Lagrange interpolation operator in the exponent is

$$\mathsf{ExpLI}\left[j_0, \cdots, j_z; F_{j_0}, \cdots, F_{j_z}\right](x) = \prod_{t=0}^{z} g^{f(j_t)\cdot\lambda_t(x)} = \prod_{t=0}^{z} F_{j_t}^{\lambda_t(x)}.$$

For another $z+1$ pairwise distinct points $\{(j_0, F_{j_0}), (j_1, F'_{j_1}), \cdots, (j_z, F_{j_z})\}$ which is almost identical to $\{(j_t, F_{j_t})\}_{t=0}^{z}$ except that $F'_{j_1} = F_{j_1}L$ where $L \in \mathbb{G}$, it also determines a z-degree polynomial $f'(x)$ in $\mathbb{Z}_q$, and the Lagrange interpolation operator in the exponent is

$$\begin{aligned}
&\mathsf{ExpLI}\left[j_0, j_1, \cdots, j_z; F_{j_0}, F'_{j_1}, \cdots, F_{j_z}\right](x) \\
&= \prod_{t=0}^{z} g^{f'(j_t)\cdot\lambda_t(x)} = {F'_{j_1}}^{\lambda_1(x)} \prod_{t=0,t\neq 1}^{z} F_{j_t}^{\lambda_t(x)} \\
&= (F_{j_1}L)^{\lambda_1(x)} \prod_{t=0,t\neq 1}^{z} F_{j_t}^{\lambda_t(x)} = L^{\lambda_1(x)} \prod_{t=0}^{z} F_{j_t}^{\lambda_t(x)} \\
&= L^{\lambda_1(x)} \mathsf{ExpLI}\left[j_0, \cdots, j_z; F_{j_0}, \cdots, F_{j_z}\right](x).
\end{aligned}$$

Thus, in particular, we have that

$$\begin{aligned}
&\mathsf{ExpLI}\left[j_0, j_1, \cdots, j_z; F_{j_0}, F'_{j_1}, \cdots, F_{j_z}\right](0) \\
&= L^{\lambda_1(0)} \mathsf{ExpLI}\left[j_0, \cdots, j_z; F_{j_0}, \cdots, F_{j_z}\right](0),
\end{aligned}$$

where $\lambda_1(0) = \prod_{0\le k\neq 1\le z} \frac{j_k}{j_k - j_1}$, which can be computed from $\{j_t\}_{t=0}^{z}$.

Now, we present the attack against the CCA2 security of KHL scheme: Given the target enabling block $T^* = \mathsf{Enc}(\mathrm{PK}, \mathcal{R}, s_\sigma) = \big(S, u_1, u_2, C, (j_1, F_{j_1}), (j_2, F_{j_2}), \cdots, (j_z, F_{j_z})\big)$, adversary $\mathcal{A}$'s goal is to correctly guess the bit σ. Note that the encapsulated session key s_σ equals to $\frac{S}{\mathsf{ExpLI}(i, j_1, \cdots, j_z; F_i, F_{j_1}, \cdots, F_{j_z})(0)}$, where $F_i = H_i \frac{C}{C_i} = H_i \frac{C}{u_1^{X_1(i)+Y_1(i)\alpha} u_2^{X_2(i)+Y_2(i)\alpha}}$ is dominated by ciphertext components (C, u_1, u_2). Adversary $\mathcal{A}$ picks $L \in \mathbb{G} \setminus \{1_{\mathbb{G}}\}$, where $1_{\mathbb{G}}$ is the identity element of $\mathbb{G}$, defines $F'_{j_1} = F_{j_1}L$, and then issues a decryption oracle query on $\langle i, T'\rangle$ where $T' = \big(S, u_1, u_2, C, (j_1, F'_{j_1}), (j_2, F_{j_2}), \cdots, (j_z, F_{j_z})\big)$. Note that since $T' \neq T^*$, it is legal for adversary $\mathcal{A}$ to issue this query. Thus according to KHL scheme, adversary $\mathcal{A}$ is then given a session key $s' = \frac{S}{\mathsf{ExpLI}(i, j_1, j_2, \cdots, j_z; F_i, F'_{j_1}, F_{j_2}, \cdots, F_{j_z})(0)}$.

Note that, T' has the same components (C, u_1, u_2) with the target enabling block T^*, and hence the value F_i computed from T' is the same as that computed from T^*. Thus we have

$$\begin{aligned}
s' &= \frac{S}{\mathsf{ExpLI}(i, j_1, j_2, \cdots, j_z; F_i, F'_{j_1}, F_{j_2}, \cdots, F_{j_z})(0)} \\
&= \frac{S}{L^{\lambda_1(0)}\mathsf{ExpLI}(i, j_1, j_2, \cdots, j_z; F_i, F_{j_1}, F_{j_2}, \cdots, F_{j_z})(0)} \\
&= \frac{1}{L^{\lambda_1(0)}} \frac{S}{\mathsf{ExpLI}(i, j_1, j_2, \cdots, j_z; F_i, F_{j_1}, F_{j_2}, \cdots, F_{j_z})(0)} \\
&= \frac{1}{L^{\lambda_1(0)}} s_\sigma .
\end{aligned}$$

Then adversary $\mathcal{A}$ can obtain the original session key by computing $s_\sigma = L^{\lambda_1(0)} s'$. With s_σ, adversary $\mathcal{A}$ can easily decide the bit σ, and hence can break the CCA2 security of KHL scheme.

Remark 2. Actually, given the target enabling block T^*, one could even easily build a different enabling block (by altering two F_j's only) so that the underlying session key remains the same. For example, given $T^* = \big(S, u_1, u_2, C, (j_1, F_{j_1}), (j_2, F_{j_2}), \cdots, (j_z, F_{j_z})\big)$, we can easily see that the new enabling block $\hat{T} = \big(S, u_1, u_2, C, (j_2, F_{j_2}), (j_1, F_{j_1}), \cdots, (j_z, F_{j_z})\big)$ encrypts the same session key s_σ. Thus the adversary can easily break the CCA2 security of KHL scheme by issuing a decryption oracle query on $\langle i, \hat{T} \rangle$.

4. Discussion and conclusion

Note that, in our attack against KHL scheme, the attacker does not need to corrupt any user, which means that KHL scheme is not CCA2 secure even without corruption of any user.

In order to prove the adaptive CCA2 security, the work in [15] proposed six games, $\mathbf{G}_0, \cdots, \mathbf{G}_5$, where $\mathbf{G}_0$ corresponds to the definitional game for adaptive CCA2 security, and T_i denotes the event that $\sigma = \sigma^*$ in $\mathbf{G}_i$. The proof strategy is, roughly speaking, to show that the views of the attacker in any two neighboring games $\mathbf{G}_i$ and $\mathbf{G}_{i+1}$, $0 \le i \le 4$, are indistinguishable (i.e., $|\Pr[T_{i+1}] - \Pr[T_i]|$ is negligible), while in the last game the attacker only gets negligible advantage. However, our attack indicates that the security proofs in [15] must be flawed. Below we shall identify the flaws in the security proofs.

KHL scheme is based on both Cramer–Shoup encryption and Lagrange-Interpolation, i.e., (S, u_1, u_2, C) in the enabling block corresponds to the Cramer–Shoup encryption and $\big((j_1, F_{j_1}), \cdots, (j_z, F_{j_z})\big)$ corresponds to the elements of Lagrange-Interpolation. For efficiency reasons, KHL scheme does not use MAC to bind the elements of Lagrange Interpolation. As a consequence, as indicated in our attack, altering some of these elements does not change the status of the enabling block (valid/invalid), and thus KHL scheme cannot ensure the CCA2 security. The main problem in the security proofs of KHL scheme is that, they never consider the situation where one submits an enabling block with the same Cramer–Shoup part as the target enabling block but with different values for the F_j's. This leads to that, in the security proofs, some neighboring games $\mathbf{G}_i$ and $\mathbf{G}_{i+1}$ are *not* indistinguishable as claimed.

Let's take games $\mathbf{G}_1$ and $\mathbf{G}_2$ as an example. Roughly speaking, game $\mathbf{G}_1$ is a purely conceptual re-formulation of $\mathbf{G}_0$ (note that, in game $\mathbf{G}_1$, $u_1 = g_1^{r_1}$ and $u_2 = g_2^{r_1}$, where r_1 is taken uniformly at random from Z_q). The only difference between $\mathbf{G}_2$ and $\mathbf{G}_1$ is that the value u_2 in $\mathbf{G}_2$ is now set to be $g_2^{r_2}$, where r_2 is taken uniformly at random from $Z_q \setminus \{r_1\}$. It is claimed in [15], under the DDH assumption, $\mathbf{G}_2$ is indistinguishable from $\mathbf{G}_1$. Unfortunately, this is *not* true. To indicate the distinguishability between $\mathbf{G}_1$ and $\mathbf{G}_2$, let's take our concrete attack as an example. In game $\mathbf{G}_1$, by modifying F_{j_1} to $F'_{j_1} = F_{j_1} L$ and submitting $\langle i, T' \rangle$ with $T' = \big(S, u_1, u_2, C, (j_1, F'_{j_1}), (j_2, F_{j_2}), \cdots, (j_z, F_{j_z})\big)$ to the decryption oracle, the adversary can derive s_σ from the response $s' = \frac{1}{L^{\lambda_1(0)}} s_\sigma$. However, in game $\mathbf{G}_2$, by submitting $\langle i, T' \rangle$ to the decryption oracle, it can be verified that the response would be $s' = \frac{1}{L^{\lambda_1(0)} g_2^{(r_2 - r_1)(Z_2(i) - X_2(i) - Y_2(i))\lambda_0(0)}} s_\sigma$. Since $g_2^{(r_2 - r_1)}$ is unknown to the adversary, from the response s' it is impossible for the adversary to derive s_σ. Therefore, $\mathbf{G}_2$ is *not* indistinguishable from $\mathbf{G}_1$ under the DDH assumption.

In addition, we note that $\mathbf{G}_2$ and $\mathbf{G}_3$ are *not* indistinguishable (i.e., $|\Pr[T_3] - \Pr[T_2]|$ is not negligible). The difference between $\mathbf{G}_3$ and $\mathbf{G}_2$ is that, when the adversary submits a ciphertext such that $u_2 \ne u_1^w$, the decryption oracle outputs $\perp$ to reject this ciphertext. In [15], it defines R_3 to be the event that the adversary submits some decryption queries which are rejected in $\mathbf{G}_3$ but passed in $\mathbf{G}_2$, and claimed that $|\Pr[T_3] - \Pr[T_2]| = \Pr[R_3]$ is negligible. However, recall that in KHL scheme (also in $\mathbf{G}_2$), the decryption oracle does not explicitly reject ciphertexts, and hence any ciphertext would pass in $\mathbf{G}_2$. Thus R_3 is in fact the event that the adversary submits some decryption queries that are rejected in $\mathbf{G}_3$. Obviously, $\Pr[R_3]$ is non-negligible, since the adversary can arbitrarily submit ciphertexts with $u_2 \ne u_1^w$ which will be rejected in $\mathbf{G}_3$. Therefore, $|\Pr[T_3] - \Pr[T_2]|$ is not negligible.

We note that our attack may be prevented by using the techniques employed in DF scheme [14], where an additional MAC-key k is encrypted (together with the session-key s) and is also used to authenticate the ciphertext. However, this approach greatly impairs the efficiency of KHL scheme, rendering the resulting scheme no more efficient than DF scheme. Furthermore, abandoning the one-time MAC employed in DF scheme was claimed to be one of the main technical contributions in [15]. It seems to be rather difficult to fix KHL scheme to satisfy CCA2 security without loss of efficiency.

Acknowledgements

This work was supported by National Science Foundation of China (Grant Nos. 61272413, 61133014, 61472165 and 61272415), Fok Ying Tong Education Foundation (Grant No. 131066), Program for New Century Excellent Talents in University (Grant No. NCET-12-0680), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20134401110011), Foundation for Distinguished Young Talents in Higher Education of Guangdong (Grant No. 2012LYM 0027), the Fundamental Research Funds for the Central Universities (Grant No. 11613106) and SERC, A*STAR (Grant No. 102 101 0027 in Singapore). This work is also supported by China Scholarship Council and the Invitation Programs for Foreign-based Researchers provided by NICT (Grant No. 2014-001).

References

[1] S. Berkovits, How to broadcast a secret, in: D.W. Davies (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 547, Springer, 1991, pp. 535–541.

[2] A. Fiat, M. Naor, Broadcast encryption, in: D.R. Stinson (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 773, Springer, 1993, pp. 480–491.

[3] Y. Dodis, N. Fazio, Public key broadcast encryption for stateless receivers, in: J. Feigenbaum (Ed.), Digital Rights Management Workshop, in: Lecture Notes in Computer Science, vol. 2696, Springer, 2002, pp. 61–80.

[4] D. Halevy, A. Shamir, The LSD broadcast encryption scheme, in: M. Yung (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 2442, Springer, 2002, pp. 47–60.

[5] D. Boneh, C. Gentry, B. Waters, Collusion resistant broadcast encryption with short ciphertexts and private keys, in: V. Shoup (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 3621, Springer, 2005, pp. 258–275.

[6] N.-S. Jho, J.Y. Hwang, J.H. Cheon, M.-H. Kim, D.H. Lee, E.S. Yoo, One-way chain based broadcast encryption schemes, in: R. Cramer (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 559–574.

[7] C. Delerablée, P. Paillier, D. Pointcheval, Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys, in: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing, in: Lecture Notes in Computer Science, vol. 4575, Springer, 2007, pp. 39–59.

[8] C. Gentry, B. Waters, Adaptive security in broadcast encryption systems (with short ciphertexts), in: A. Joux (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 5479, Springer, 2009, pp. 171–188.

[9] B. Chor, A. Fiat, M. Naor, Tracing traitors, in: Y. Desmedt (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 839, Springer, 1994, pp. 257–270.

[10] D. Naor, M. Naor, J. Lotspiech, Revocation and tracing schemes for stateless receivers, in: J. Kilian (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 2139, Springer, 2001, pp. 41–62.

[11] M. Naor, B. Pinkas, Efficient trace and revoke schemes, in: Y. Frankel (Ed.), Financial Cryptography, in: Lecture Notes in Computer Science, vol. 1962, Springer, 2000, pp. 1–20.

[12] D. Boneh, M.K. Franklin, An efficient public key traitor tracing scheme, in: M.J. Wiener (Ed.), Advances in Cryptology, Proceedings 19th Annual International Cryptology Conference, CRYPTO '99, Santa Barbara, California, USA, August 15–19, 1999, in: Lecture Notes in Computer Science, vol. 1666, Springer, 1999, pp. 338–353.

[13] E. Gafni, J. Staddon, Y.L. Yin, Efficient methods for integrating traceability and broadcast encryption, in: M.J. Wiener (Ed.), Advances in Cryptology, Proceedings 19th Annual International Cryptology Conference, CRYPTO '99, Santa Barbara, California, USA, August 15–19, 1999, in: Lecture Notes in Computer Science, vol. 1666, Springer, 1999, pp. 372–387.

[14] Y. Dodis, N. Fazio, Public key trace and revoke scheme secure against adaptive chosen ciphertext attack, in: Y. Desmedt (Ed.), Public Key Cryptography, in: Lecture Notes in Computer Science, vol. 2567, Springer, 2003, pp. 100–115.

[15] C.H. Kim, Y.H. Hwang, P.J. Lee, An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack, in: C.-S. Laih (Ed.), ASIACRYPT, in: Lecture Notes in Computer Science, vol. 2894, Springer, 2003, pp. 359–373.

[16] W.-G. Tzeng, Z.-J. Tzeng, A public-key traitor tracing scheme with revocation using dynamic shares, in: K. Kim (Ed.), Public Key Cryptography, in: Lecture Notes in Computer Science, vol. 1992, Springer, 2001, pp. 207–224.

[17] R. Canetti, S. Goldwasser, An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack, in: J. Stern (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 1592, Springer, 1999, pp. 90–106.

[18] R. Cramer, V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, in: H. Krawczyk (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 13–25.

[19] R. Cramer, V. Shoup, Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption, in: L.R. Knudsen (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 2332, Springer, 2002, pp. 45–64.