Amendment to Trace and Revoke Systems with Short Ciphertexts

Xingwen Zhao
(Corresponding author: Xingwen Zhao)
School of Telecommunications Engineering, Xidian University
Mbox#106, Taibai South Road #2, Yanta District, Xi'an, 710071, China
(Email: sevenzhao@hotmail.com)
(Received Dec. 18, 2011; revised and accepted Feb. 1, 2012)

Abstract

Traitor tracing is needed because some users in broadcast encryption system may give out their decryption keys to construct pirate decoders. Recently, Liu and Yuan described a trace and revoke systems with short ciphertexts. In this paper, we show that their scheme cannot achieve traitor tracing, since any receiver can decide whether the given ciphertext is well-formed or not so as to decide whether the system is now in normal broadcasting mode or in traitor tracing mode. Thus, any user can construct decoders that will decrypt well-formed ciphertexts (ciphertexts for normal broadcasting) and refuse the badly-formed ciphertexts (ciphertexts for traitor tracing), so that traitors cannot be identified. In such case, innocent users will be framed. We provide an amendment to their scheme and render it useful in traitor tracing against both perfect and imperfect decoders.

Keywords: Broadcast Encryption, cryptanalysis, traitor tracing

1 Introduction

Broadcast encryption provides a convenient method to distribute digital content to subscribers over an insecure broadcast channel so that only the qualified users can recover the data. Broadcast encryption is quite useful and enjoys many applications including Pay-TV systems, distribution of copyrighted materials such as DVD.

Because some users (called traitors) may give out their decryption keys to construct pirate decoders, the ability of traitor tracing is needed for broadcast encryption system. The first traitor tracing scheme against pirate decoders was presented by Chor, Fiat and Naor in [4]. Since then, many works have been presented, and among them some works [3, 5, 7, 8, 9, 10] combine the tracing and revoking abilities to make the broadcast encryption systems more practical.

As we notice that, in the these trace and revoke schemes, the ciphertext length is not short. It is either linear to the number of revoked users [7, 8, 9, 11] or in proportion to $\sqrt{N}$ [3, 5], where N is the total number of users in the system.

Recently, Liu and Yuan [6] described a trace and revoke systems with short ciphertexts, by combining public key broadcast encryption scheme [1] with collusion secure codes based traitor tracing scheme [2]. However, their scheme cannot achieve traitor tracing, since any receiver can decide whether the given ciphertext is well-formed or not so as to decide whether the system is now in normal broadcasting mode or in traitor tracing mode. Thus, any user can construct decoders that will decrypt well-formed ciphertexts (ciphertexts for normal broadcasting) and refuse the badly-formed ciphertexts (ciphertexts for traitor tracing), so that traitors cannot be identified.

1.1 Our contributions

1) We show that their scheme cannot achieve traitor tracing, by describing how a receiver can decide whether a given ciphertext is well-formed or not so as to decide whether the system is now in normal broadcasting mode or in traitor tracing mode. We also show that innocent users will be framed with high probability.

2) We provide an amendment to Liu and Yuan's scheme and render it useful in traitor tracing against both perfect and imperfect decoders.

1.2 Organization

The remainder of this paper is organized as follows. In Section 2 we briefly review Liu and Yuan's trace and revoke scheme. In Section 3 we show that any one can decide whether the ciphertext is well-formed or not, so the tracing algorithm does not work. We give out an amendment to this flaw in Section 4. Section 5 concludes our paper.

2 Review of Liu and Yuan's Trace and Revoke Scheme

Firstly, we review Liu and Yuan's trace and revoke scheme [6] in brief. Their scheme is based on δ-robust collusion secure code denoted as two algorithms (G, T), where G is the code generating algorithm and T is the tracing algorithm. We refer our readers to [2] for the detailed definition of collusion secure code and [6] for detailed definition of protocol model for the trace and revoke system.

Table 1: Partial list of symbols

Symbols	Descriptions
δ	the rate that the captured decoder fails to decrypt well-formed ciphertexts
λ	a security parameter used for the system setup
$G(\cdot)$	the code generating algorithm for the δ-robust collusion secure code
$T(\cdot)$	the tracing algorithm for the δ-robust collusion secure code
Γ	a set of codeword generated by $G(\cdot)$
tk	the tracing key for Γ
l	the length of codeword
$w^{(i)}$	the codeword for user i
$w_h^{(i)}$	the bit value of position h in the codeword $w^{(i)}$
bk	the broadcast key of the system
$sk[i,s,h]$	the secret key for user i on codeword bit position h (with bit value $w_h^{(i)}=s$)
sk_i	the set of secret keys for user i
$\mathcal{D}$	the captured pirate decoder
$k \xleftarrow{R} E_k(bk,r)$	a temporary session key k is generated using random number r.
$(c_0,c_1,c_2) \xleftarrow{R} E_c(bk,S,r)$	a set of ciphertext is generated for receiver set S using random number r.

The four algorithms ($Setup$, $Encrypt$, $Decrypt$, $Trace$) of Liu and Yuan's trace and revoke scheme are described as follows:

- $Setup(n-1,\ \lambda)$
  Let $\epsilon = 1/2^{\lambda}$. The algorithm works as follows:

  1) Generate a pair of code and tracing key by running

  $$(\Gamma, tk) \quad = \quad G(n-1,\epsilon).$$

  Let $\Gamma = \{w^{(2)},\ldots,w^{(n)}\} \subseteq \{0,1\}^l$, where l is the codeword length.

  2) Let $\mathbb{G}$ be a bilinear group of prime order p. Pick a random generator $g \in \mathbb{G}$ and a random $\alpha \in \mathbb{Z}_p$. Compute $g_i = g^{\alpha^i} \in \mathbb{G}$ for $i = 1,\ldots,n,n+2,\ldots,2n$. Pick l random elements $\gamma_1,\ldots,\gamma_l \in \mathbb{Z}_p$ and set $v_1 = g^{\gamma_1}$, $\ldots$, $v_l = g^{\gamma_l} \in \mathbb{G}$. For $i = 2,\ldots,n$ and $s = 0,1$ and $h = 1,\ldots,l$, set

  $$sk[i,s,h] = g_{i+sn}^{\gamma_h} = v_h^{(\alpha^{i+sn})} \in \mathbb{G}.$$

  3) Define broadcast key as

  $$bk \quad \leftarrow \quad \left(\begin{array}{c} g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, \\ v_1, v_1^{(\alpha^n)}, \ldots, v_l, v_l^{(\alpha^n)} \end{array} \right).$$

  4) For $i = 2,\ldots,n$ define secret key

  $$sk_i \quad \leftarrow \quad (w^{(i)}, sk[i,w_1^{(i)},1],\ldots,sk[i,w_l^{(i)},l]).$$

  Table 2 shows an example of secret key.

  5) Output tk, bk and $(sk_2,\ldots,sk_n)$.

Table 2: An example of secret keys: the keys in italics are assigned to the user with codeword $w^{(i)}$

$w^{(i)}$	sk_i	
0	***sk[i,0,1]***	$sk[i,1,1]$
1	$sk[i,0,2]$	***sk[i,1,2]***
0	***sk[i,0,3]***	$sk[i,1,3]$
...	...	...
0	***sk[i,0,l]***	$sk[i,1,l]$

- $Encrypt(bk,S)$
  Pick a random $r \in \mathbb{Z}_p$. For the receiver set $S \subseteq \{2,\ldots,n\}$, the algorithm works as follows:

  1) Set $k = e(g,g_{n+1})^r \in \mathbb{G}_1$ (here the key space is set as $\mathbb{G}_1$). The value $e(g,g_{n+1})$ can be computed as $e(g_2,g_{n-1})$. As shorthand, this process is denoted as $k \xleftarrow{R} E_k(bk,r)$.

  2) Pick a random $h \in \{1,\ldots,l\}$. For $s = 0,1$ set

  $$\begin{aligned} c_s &= (v_h^{(\alpha^{sn})} \cdot \prod_{j\in S} g_{n+1-j})^r \in \mathbb{G}, \\ c_2 &= g^r \in \mathbb{G}. \end{aligned}$$

  This process is denoted as $(c_0,c_1,c_2) \xleftarrow{R} E_c(bk,S,r)$.

  3) Define ciphertext $c \leftarrow (h,c_0,c_1,c_2)$.

  4) Output k and c.

- $Decrypt(bk,\ i,\ sk_i,\ S,\ c)$
  Let $(h,c_0,c_1,c_2) \leftarrow c$ and $s = w_h^{(i)}$ $(i \in \{2,\ldots,n\})$. Output

  $$k \quad = \quad e(g_i,c_s)/e(sk[i,s,h] \cdot \prod_{j\in S, j\neq i} g_{n+1-j+i}, c_2).$$

- $Trace^{\mathcal{D}}(tk, S)$
  Suppose the adversary $\mathcal{A}$ obtains a set of secret keys of t users $T \subseteq \{2, \ldots, n\}$ $(S \cap T \neq \emptyset)$ and uses them to build a pirate decoder $\mathcal{D}$. Let $C \subseteq \{0,1\}^l$ be the set of fingerprinting codewords that are corresponding to decryption keys held by $\mathcal{A}$.

  Definition 1. *For a given broadcast key bk, the error-rate δ of $\mathcal{D}$ is the probability that $\mathcal{D}$ fails to decrypt well-formed ciphertexts:*

  $$\delta := Pr[(k, c) \xleftarrow{R} Encrypt(bk, S) : \mathcal{D}(c) \neq k].$$

  – *Tracing Perfect Decoders*
  Assume that $\mathcal{D}$ is a perfect decoder, namely $\delta = 0$. For $h = 1, \ldots, l$, experiment TR_h is defines as follows:

  $$\begin{array}{rcll} (r_0, r_1) & \xleftarrow{R} & \mathbb{Z}_p & (r_0 \neq r_1) \\ k & \xleftarrow{R} & E_k(bk, r_0) & \\ (c_0, c_2) & \leftarrow & E_c(bk, S, r_0), & \\ c_1 & \leftarrow & E_c(bk, S, r_1) & \\ c^* & = & (h, c_0, c_1, c_2) & \\ \hat{k} & = & \mathcal{D}(c^*). & \end{array}$$

  If decoder $\mathcal{D}$ outputs $\hat{k} = k$, w_h is set to 0, which means $\mathcal{A}$ knows $sk[i, 0, h]$. Else, w_h is set to 1, which means $\mathcal{A}$ knows $sk[i, 1, h]$. When the tracing on $h = 1, \ldots, l$ is completed, the recovered codeword w^* is set as

  $$w^* := w_1 \ldots w_l \in \{0,1\}^l.$$

  A set of traitor is output as $T(tk, w^*)$.

  – *Tracing Imperfect Decoders*

  As for imperfect decoders with error-rate less than some fixed δ, a δ'-robust fingerprinting code is needed [2] to ensure tracing sucessful, with

  $$\delta' = \frac{\delta}{1 - \frac{2}{\sqrt{\lambda}}}.$$

  For $h = 1, \ldots, l$, experiment $\mathrm{RobustTR}_h$ is defines as follows:

  Repeat the following steps $\lambda^2 \ln l$ times:

  $$\begin{array}{rcl} (r_0, r_1) & \xleftarrow{R} & \mathbb{Z}_p, \text{ with } (r_0 \neq r_1) \\ k & \xleftarrow{R} & E_k(bk, r_0), \\ (c_0, c_2) & \leftarrow & E_c(bk, S, r_0), \\ c_1 & \leftarrow & E_c(bk, S, r_1), \\ c^* & = & (h, c_0, c_1, c_2), \\ \hat{k} & = & \mathcal{D}(c^*). \end{array}$$

  Let p_h be the fraction of times that $k = \hat{k}$;

  Repeat the following steps $\lambda^2 \ln L$ times:

  $$\begin{array}{rcl} r & \xleftarrow{R} & \mathbb{Z}_p, \\ k & \xleftarrow{R} & E_k(bk, r), \\ (c_0, c_1, c_2) & \leftarrow & E_c(bk, S, r), \\ c & = & (h, c_0, c_1, c_2), \\ \hat{k} & = & \mathcal{D}(c^*). \end{array}$$

  Let q_h be the fraction of times that $k = \hat{k}$.

  Define $w_h \in \{0, 1, ?\}$ as:

  $$w_h = \begin{cases} 0 & \text{if } p_h > 0 \quad (\mathcal{A} \text{ knows } sk[i,0,h]) \\ 1 & \text{elseif } q_h > \frac{1}{\sqrt{\lambda}} \quad (\mathcal{A} \text{ knows } sk[i,1,h]) \\ \text{`?'} & \text{otherwise} \quad (p_h = 0 \text{ and } q_h < \frac{1}{\sqrt{\lambda}}) \end{cases}$$

  and let $w^* = w_1 \ldots w_l \in \{0, 1, ?\}^l$. A set of traitors is output as $T(tk, w^*)$.

3 Distinguish Broadcasting Mode from Traitor Tracing Mode

We show how to distinguish normal broadcasting ciphertexts (well-formed) from tracing ciphertexts (badly-formed or invalid).

In a traitor tracing algorithm, the tracing ability is obtained by finding out the distinct decrypting ability of the captured decoder. As for Liu and Yuan's scheme [6], the tracing algorithm is used to test whether the captured decoder has ability to decrypt the ciphertext that is encrypted for tracing position h and codeword bit $w_h = 0$ only. The following tracing logic is implied in their tracing algorithm:

- If all codewords in the decoder contain a $w_h = 0$ in tracing position h, the decoder will always output $\hat{k} = k$ since it does not hold $w_h = 1$ and cannot decide whether the ciphertext is well-formed or not.
- If all the codewords in the decoder contain a $w_h = 1$ in tracing position h, the decoder will output a $\hat{k} \neq k$ with high probability.
- If the codewords in the decoder contain both $w_h = 0$ and $w_h = 1$, no matter what the decoder outputs,

the recovered codeword bit is always in the feasible set of the codewords [2, 6] in the decoder.

However, if the decoder can always decide whether the ciphertext is well-formed or not regardless of what codeword bits it holds, the traitor tracing algorithm is broken. Now we show that any one in Liu and Yuan's scheme [6] can check whether the ciphertext is well-formed or not, so that any decoder can reject to decrypt the traitor tracing ciphertexts. The process is described as follows:

1) On receiving a ciphertext c, the receiver parses it as (h, c_0, c_1, c_2);
2) For $s = 0, 1$ checks whether the following equation holds or not:

$$e(c_s, g) = e(v_h^{(\alpha^{sn})}, c_2) \cdot e(\prod_{j \in S} g_{n+1-j}, c_2).$$

If any of the two equations does not hold, the ciphertext is invalid and the receiver decides that the system is now running in traitor tracing mode. If so, the receiver outputs a random key $\hat{k}$. Else, the system is now running in normal broadcasting mode, and the receiver decrypts as usual.

The reason is that, in the tracing algorithm of Liu and Yuan's Scheme, c_1 uses a random number r_1 that is different from the number r_0 used in c_2. Thus, the tracing ciphertext cannot pass the validity check.

Since any one can check whether the system is in traitor tracing mode or not, the pirate decoder can be programmed to frame innocent users. If the decoder rejects all invalid ciphertexts (returns random keys), no matter the system is in the mode of tracing perfect decoders or in the mode of tracing imperfect decoders, the tracer will output a recovered codeword as $w = 11\ldots1$. The users with feasible codeword $w = 11\ldots1$ are framed, if the codeword $w = 11\ldots1$ is not held by the decoder.

4 The Proposed Amendment

The goal of the amendment is to ensure that only the colluding traitors whose codewords contain both $w_h = 0$ and $w_h = 1$ in tracing position h can decide whether the system is in traitor tracing mode or not. In such case, the tracing logic works fine.

Our idea is to generate two independent sets of ciphertexts for $w_h = 0$ and $w_h = 1$ of tracing position h respectively, both encrypting a same temporary session key in normal broadcasting mode while encrypting two different keys in traitor tracing mode. Such modification prevents the colluding traitors from checking the validity of the ciphertexts, unless they hold decryption keys of both $w_h = 0$ and $w_h = 1$ for tracing position h. After the modification, as we notice that, if the codewords of the colluding traitors all contain a same $w_h = 0$ (or a same $w_h = 1$), they cannot detect the tracing mode and will decrypt as usual. If the codewords in the decoder contain both $w_h = 0$ and $w_h = 1$, the decoder can detect the tracing mode. However, no matter what the decoder outputs, the recovered codeword bit is always in the feasible set of the codewords in the decoder.

The modified algorithms of *Encrypt*, *Decrypt*, and *Trace* are described as follows (*Setup* algorithm remains unchanged):

- *Encrypt*(bk,S)
  Pick two random numbers $r_0, r_1 \in \mathbb{Z}_p$. The algorithm works as follows:
  1) Set $k_0 = e(g, g_{n+1})^{r_0}, k_1 = e(g, g_{n+1})^{r_1} \in \mathbb{G}_1$. $e(g, g_{n+1})$ can be computed as $e(g_2, g_{n-1})$. Select a random temporary session key TSK from $\mathbb{G}_1$.
  2) For $S \subseteq \{2, \ldots, n\}$, pick a random $h \in \{1, \ldots, l\}$ and set

$$\begin{aligned} c_{0,1} &= (v_h \cdot \prod_{j \in S} g_{n+1-j})^{r_0} \in \mathbb{G}, \\ c_{0,2} &= g^{r_0} \in \mathbb{G}, \\ c_{0,3} &= k_0 \oplus TSK \in \mathbb{G}_1, \\ c_{1,1} &= (v_h^{(\alpha^n)} \cdot \prod_{j \in S} g_{n+1-j})^{r_1} \in \mathbb{G}, \\ c_{1,2} &= g^{r_1} \in \mathbb{G}, \\ c_{1,3} &= k_1 \oplus TSK \in \mathbb{G}_1. \end{aligned}$$

  This process is denoted as $(c_{0,1}, c_{0,2}, c_{0,3}) \xleftarrow{R} E_c(bk, S, r_0, TSK)$ and $(c_{1,1}, c_{1,2}, c_{1,3}) \xleftarrow{R} E_c(bk, S, r_1, TSK)$.
  3) Define ciphertext c as $c \leftarrow (h, c_{0,1}, c_{0,2}, c_{0,3}, c_{1,1}, c_{1,2}, c_{1,3})$.
  4) Output TSK and c.

- *Decrypt*(bk, i, sk_i, S, c)
  Let $(h, c_{0,1}, c_{0,2}, c_{0,3}, c_{1,1}, c_{1,2}, c_{1,3}) \leftarrow c$ and $s = w_h^{(i)}$ ($i \in \{2, \ldots, n\}$). Compute

$$k_s = e(g_i, c_{s,1}) / e(sk[i, s, h] \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}, c_{s,2}).$$

  $TSK = k_s \oplus c_{s,3}$ is output as the temporary session key.

- $Trace^{\mathcal{D}}(tk, S)$
  Suppose the adversary $\mathcal{A}$ obtains a set of secret keys of t users $T \subseteq \{2, \ldots, n\}$ ($S \cap T \neq \emptyset$) and uses them to build a pirate decoder $\mathcal{D}$. Let $C \subseteq \{0, 1\}^l$ be the set of fingerprinting codewords that are corresponding to decryption keys held by $\mathcal{A}$.
  - *Tracing Perfect Decoders*
    Assume that $\mathcal{D}$ is a perfect decoder, namely $\delta = 0$. For $h = 1, \ldots, l$, experiment TR_h is defines as follows:

$$
\begin{aligned}
(r_0, r_1) &\xleftarrow{R} \mathbb{Z}_p \quad (r_0 \neq r_1),\\
TSK &\xleftarrow{R} \mathbb{G}_1,\\
K_R &\xleftarrow{R} \mathbb{G}_1 \quad (TSK \neq K_R),\\
(c_{0,1}, c_{0,2}, c_{0,3}) &\xleftarrow{R} E_c(bk, S, r_0, TSK),\\
(c_{1,1}, c_{1,2}, c_{1,3}) &\xleftarrow{R} E_c(bk, S, r_1, K_R),\\
c^* &= (h, c_{0,1}, c_{0,2}, c_{0,3}, c_{1,1}, c_{1,2}, c_{1,3}),\\
TSK^* &= \mathcal{D}(c^*).
\end{aligned}
$$

If decoder $\mathcal{D}$ outputs $TSK^* = TSK$, w_h is set to 0, which means $\mathcal{A}$ knows $sk[i, 0, h]$. Else, w_h is set to 1, which means $\mathcal{A}$ knows $sk[i, 1, h]$. When the tracing on $h = 1, \ldots, l$ is completed, the recovered codeword w^* is set as

$$w^* := w_1 \ldots w_l \in \{0, 1\}^l.$$

A set of traitor is output as $T(tk, w^*)$.

– *Tracing Imperfect Decoders*
As for imperfect decoders with error-rate less than some fixed δ, a δ'-robust fingerprinting code is needed [2] to ensure tracing sucessful, with

$$\delta' = \frac{\delta}{1 - \frac{2}{\sqrt{\lambda}}}.$$

For $h = 1, \ldots, l$, experiment $\mathrm{RobustTR}_h$ is defines as follows:

Repeat the following steps $\lambda^2 \ln l$ times:

$$
\begin{aligned}
(r_0, r_1) &\xleftarrow{R} \mathbb{Z}_p \quad (r_0 \neq r_1),\\
TSK &\xleftarrow{R} \mathbb{G}_1,\\
K_R &\xleftarrow{R} \mathbb{G}_1 \quad (TSK \neq K_R),\\
(c_{0,1}, c_{0,2}, c_{0,3}) &\xleftarrow{R} E_c(bk, S, r_0, TSK),\\
(c_{1,1}, c_{1,2}, c_{1,3}) &\xleftarrow{R} E_c(bk, S, r_1, K_R),\\
c^* &= (h, c_{0,1}, c_{0,2}, c_{0,3}, c_{1,1}, c_{1,2}, c_{1,3}),\\
TSK^* &= \mathcal{D}(c^*).
\end{aligned}
$$

Let p_h be the fraction of times that $TSK^* = TSK$;
Repeat the following steps $\lambda^2 \ln L$ times:

$$
\begin{aligned}
(r_0, r_1) &\xleftarrow{R} \mathbb{Z}_p \quad (r_0 \neq r_1),\\
TSK &\xleftarrow{R} \mathbb{G}_1,\\
(c_{0,1}, c_{0,2}, c_{0,3}) &\xleftarrow{R} E_c(bk, S, r_0, TSK),\\
(c_{1,1}, c_{1,2}, c_{1,3}) &\xleftarrow{R} E_c(bk, S, r_1, TSK),\\
c^* &= (h, c_{0,1}, c_{0,2}, c_{0,3}, c_{1,1}, c_{1,2}, c_{1,3}),\\
TSK^* &= \mathcal{D}(c^*).
\end{aligned}
$$

Let q_h be the fraction of times that $TSK^* = TSK$.

Define $w_h \in \{0, 1, ?\}$ as:

$$
w_h = \begin{cases}
0 & \text{if } p_h > 0 & (\mathcal{A} \text{ knows } sk[i, 0, h])\\
1 & \text{elseif } q_h > \frac{1}{\sqrt{\lambda}} & (\mathcal{A} \text{ knows } sk[i, 1, h])\\
\text{`?'} & \text{otherwise} & (p_h = 0 \text{ and } q_h < \frac{1}{\sqrt{\lambda}})
\end{cases}
$$

and $w^* = w_1 \ldots w_l \in \{0, 1, ?\}^l$. A set of traitors is output as $T(tk, w^*)$.

5 Conclusion

Recently, Liu and Yuan [6] proposed a trace and revoke scheme with short ciphertexts. We show that their scheme cannot achieve traitor tracing, because any one can decide whether the system is in traitor tracing mode or in normal broadcasting mode so as they can reject to decrypt the traitor tracing ciphertexts. We outline the amendment that renders their scheme useful. The modified algorithms are described in details. The ciphertext is still of constant length though three more elements are added.

References

[1] Dan Boneh, Craig Gentry, and Brent Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proceedings of Advances in Cryptology - 25th Annual International Cryptology Conference (CRYPTO 2005)*, pp. 258–275, Santa Barbara, California, USA, August 2005.

[2] Dan Boneh and Moni Naor, "Traitor tracing with constant size ciphertext," in *Proceedings of the 2008 ACM Conference on Computer and Communications Security (CCS 2008)*, pp. 501–510, Alexandria, Virginia, USA, October 2008.

[3] Dan Boneh and Brent Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, pp. 211–220, Alexandria, VA, USA, October 2006.

[4] Benny Chor, Amos Fiat, and Moni Naor, "Tracing traitors," in *Proceedings of Advances in Cryptology - 14th Annual International Cryptology Conference (CRYPTO 1994)*, pp. 257–270, Santa Barbara, California, USA, August 1994.

[5] Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters, "Building efficient fully collusion-resilient traitor tracing and revocation schemes," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010)*, pp. 121–130, Chicago, Illinois, USA, October 2010.

[6] Lie Liu and Chun Yuan, "Trace and revoke systems with short ciphertexts," in *Proceedings of the 2nd International Conference on Security of Information and Networks (SIN 2009)*, pp. 61–66, Gazimagusa, North Cyprus, October 2009.

[7] Dalit Naor, Moni Naor, and Jeffery Lotspiech, "Revocation and tracing schemes for stateless receivers,"

in *Proceedings of Advances in Cryptology - 21st Annual International Cryptology Conference (CRYPTO 2001)*, pp. 41–62, Santa Barbara, California, USA, August 2001.

[8] Moni Naor and Benny Pinkas, "Efficient trace and revoke schemes," in *Proceedings of Financial Cryptography, 4th International Conference (FC 2000)*, pp. 1–20, Anguilla, British West Indies, February 2000.

[9] Duong Hieu Phan and Viet Cuong Trinh, "Identity-based trace and revoke schemes," in *Proceedings of the 5th International Conference on Provable Security, (ProvSec 2011)*, pp. 204–221, Xi'an, China, October 2011.

[10] Bo Yang, Hua Ma, and Shenglin Zhu, "A traitor tracing scheme based on the rsa system," *International Journal of Network Security*, vol. 5, no. 2, pp. 182–186, 2007.

[11] Xingwen Zhao and Fangguo Zhang, "A new type of id-based encryption system and its application to pay-tv systems," *International Journal of Network Security*, vol. 13, no. 3, pp. 161–166, 2011.

Xingwen Zhao is currently a lecturer in Xidian University. He received the B.S. degree and M.S. degree from School of Telecommunications Engineering, Xidian University in 1999 and 2004. He obtained his Ph.D. degree from School of Information Science and Technology at Sun Yat-sen University in 2011. His main research interests include broadcast encryption, signatures and keymanagement.