
Designs, Codes and Cryptography, 35, 47–61, 2005

A Public-Key Traitor Tracing Scheme with Revocation Using Dynamic Shares

WEN-GUEY TZENG tzeng@cis.nctu.edu.tw
Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050

ZHI-JIA TZENG zjtzeng@cis.nctu.edu.tw
Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050

Communicated by: P. Wild

Received December 14, 2001; Revised April 23, 2003; Accepted May 19, 2003

Abstract. We proposed a new public-key traitor tracing scheme with revocation capability using dynamic shares and entity revocation techniques. Our scheme's traitor tracing and revocation programs cohere tightly. The size of the enabling block of our scheme is independent of the number of receivers. Each receiver holds one decryption key only. The *distinct feature* of our scheme is that when traitors are found, we can revoke their private keys (up to some threshold z) without updating the private keys of other receivers. In particular, no revocation messages are broadcast and all receivers do nothing. Previously proposed revocation schemes need update existing keys and entail large amount of broadcast messages. Our traitor tracing algorithm works in a black-box way. It is conceptually simple and fully k-resilient, that is, it can find all traitors if the number of them is k or less. The encryption algorithm of our scheme is semantically secure assuming that the decisional Diffie-Hellman problem is hard.

Keywords: broadcast encryption, traitor tracing, revocation

1. Introduction

A broadcast encryption scheme [9] involves a sender and multiple authorized receivers. The sender has an encryption key and each receiver has a decryption (private) key such that the sender can encrypt a message and broadcast the ciphertext so that only the authorized receivers can decrypt the ciphertext. Broadcast encryption schemes have wide applications in multicast services, such as on web broadcast or pay-per-view systems, in which the system end broadcasts messages to a set of privileged receivers through a broadcast channel.

Consider a situation that a content supplier distributes digital content to its subscribers by a broadcast channel. To protect the data from eavesdropping, the content supplier encrypts the data and broadcasts the ciphertext such that only its subscribers can decrypt the ciphertext. The content supplier gives each subscriber a decoder (decoding box) for decrypting the ciphertext. Each decoder consists of a tailored key and a decryption program. However, a *traitor* (malicious subscriber)

may clone his decoder (and the private key in it) and sell the pirate decoders for profits. The traitor may modify the private key and the decryption program in the pirate decoder to avoid leaking his identity. Furthermore, some traitors may together create new private keys and decryption programs. To deter the attack, when a pirate decoder is confiscated, the content supplier wants to reveal the private key in it and trace back to its original owners. A *traitor tracing scheme* is a broadcast encryption scheme with capability of dealing with the above scenario [6]. To enhance protection further, the content supplier wants to revoke the private keys of traitors without too much work, such as, updating each subscriber's key. We focus on providing revocation capability to public-key traitor tracing schemes.

A basic technique of broadcast encryption is as follows. First, the sender selects a session key s to encrypt the message M as the *cipher block* C and embeds s in the *enabling block* T; then, the sender broadcasts $\langle T, C \rangle$. Any decoder with a legal private key can compute s from T and then uses s to compute M from C. A traitor tracing scheme tries to identify traitors by finding the private keys in the confiscated pirate decoder.

To revoke the keys of receivers, the sender broadcasts revocation messages such that only non-revoked receivers can compute their new private keys and the revoked receivers lose the decryption capability.

Efficiency consideration consists of the size of the private key that a receiver holds, the size of the enabling block, the size of revocation messages, and computation time of encryption, decryption and traitor tracing.

1.1. The Results

We propose a new public-key traitor tracing scheme with revocation capability using the dynamic share and entity revocation techniques of [2]. Our scheme's traitor tracing and revocation programs cohere tightly[1]. The enabling block of our scheme is independent of the number of receivers, but dependent on the collusion and revocation thresholds, which are k and z, respectively. Each decoder stores only one private key.

Our traitor tracing algorithm works in a black-box way. It is conceptually simple and fully k-resilient, that is, it can find all traitors if the number of them is k or less. The encryption algorithm of our scheme is semantically secure against the passive adversary assuming hardness of the decisional Diffie-Hellman problem.

The *distinct feature* of our scheme is that when the traitors are found, we can revoke their private keys (up to z keys totally) without updating the keys of other receivers. In particular, no revocation messages are broadcast and all receivers do nothing. Furthermore, we can restore a revoked private key later. We can actually increase the revocation capability beyond the threshold z with dynamic assignment of shares into the enabling blocks. This property makes our scheme highly practical. The above method is suitable for fast revocation when the number of revoked receivers does not exceed the revocation capability of the system. If we need revoke more keys permanently, we can post some revocation messages on a bulletin board. Each non-revoked receiver can update its private key at its convenient

time. To revoke mz receivers, the revocation messages are of size $O(mz)$, which is very efficient.

Our scheme is as efficient as Boneh and Franklin's public-key traitor tracing scheme in many aspects. For example, the encryption and decryption algorithms of our scheme take $O(z)$ modular exponentiations. Our black-box tracing algorithm takes $O(n^k)$ time when $k \ll n$. Note that the encryption key of our scheme dynamically depends on the revoked traitors, while that of Boneh and Franklin's scheme is fixed.

1.2. Related Work

The secret-key and coding approach has each decoder holding a set of keys (or codewords) such that the keys in the pirate decoder can be identified by combinatorial methods [1,4,6,10,15,17–19]. There is a trade-off between the size of the enabling block and the number of keys held by each decoder [5,14]. Generally speaking, if the number of receivers is large, say millions, the schemes become impractical as one of the measures grows proportionally with the number of receivers.

The public-key approach tries to have the size of the enabling block independent of the number of receivers and each decoder holding one key only [3,12]. Kurosawa and Desmedt [12] proposed a public-key traitor tracing scheme [12], on which our scheme is based. But, their scheme does not incorporate the revocation capability. Boneh and Franklin's traitor tracing scheme is algebraic with deterministic tracing such that k or less traitors who create a single-key pirate decoder can be traced efficiently. However, they have to embed a hidden trapdoor in the modulus so that the discrete logarithm problem over $Z^*_{N^2}$ can be solved in polynomial time.

As to other directions, Naor and Pinkas [15] proposed a threshold traitor tracing scheme that can trace the private keys in a pirate decoder if the decoder's decrypting probability is over some threshold. Fiat and Tassa's dynamic traitor tracing scheme [8] uses the watermarking technique to trace traitors of a pirate decoder by observing the watermarks output by the pirate decoder on the fly.

For revocation capability, the revocation scheme of Kumar et al. [11] is based on cover-free sets. To revoke t receivers among n receivers, the scheme need broadcast $O(t \log n)$ revocation messages. The tree-based revocation scheme of Wong et al. [21] need broadcast $O(2 \log n)$ revocation messages. Naor and Pinkas [16] proposed a threshold secret sharing method to provide revocation capability to broadcast encryption schemes. Its efficiency depends on the based broadcast encryption schemes.

Our scheme is an independent work done by Yoshida and Fujiwara [22]. They proposed a similar traitor tracing scheme that uses dynamic shares as well. In comparison, our revocation methods are more flexible. We have a revocation method that can revoke the number of traitors beyond the threshold set by the system.

Recently, Kurosawa and Yoshida [13] proposed a linear code-based scheme that generalizes the work of ours and that of Yoshida and Fujiwara [22].

2. Preliminaries

In this section we review the idea of our scheme, the polynomial interpolation method, the decisional Diffie-Hellman (DDH) problem and semantic security of an encryption scheme.

Polynomial interpolation. Let $f(x)=\sum_{i=0}^{z} a_i x^i$ be a polynomial of degree $z \geq 1$. Assume that each user i is given a share $(x_i, f(x_i))$. Then, a group of $z+1$ users, say users $0, 1, \ldots, z$, can compute the polynomial $f(x)$ by Lagrange's interpolation method, or equivalently solving the system of equations:

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^z \\ 1 & x_1 & \cdots & x_1^z \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_z & \cdots & x_z^z \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_z \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_z) \end{pmatrix}$$

Let $XA=F$ denote the above system of equations. If $\det(X) \neq 0$, we can solve all coefficients of $f(x)$ by $A=X^{-1}F$. The constant term a_0 is equal to the first row vector of X^{-1} multiplying F, which is

$$\sum_{t=0}^{z} \left(f(x_t) \cdot \prod_{0 \leq j \neq t \leq z} \frac{x_j}{x_j - x_t} \right),$$

where $\lambda_t = \prod_{0 \leq j \neq t \leq z} \frac{x_j}{x_j - x_t}$, $0 \leq t \leq z$, are Lagrange coefficients. Furthermore, for the exponent case, if we are given $(x_0, g^{rf(x_0)})$, $(x_1, g^{rf(x_1)})$, $\ldots$, $(x_z, g^{rf(x_z)})$, we can compute

$$g^{ra_0} = \prod_{t=0}^{z} (g^{rf(x_t)})^{\lambda_t}.$$

for arbitrary r. On the other hand, if $\det(X)=0$, we cannot get any information about a_0 or g^{ra_0}.

In traitor tracing, a set of legal users may combine their shares linearly to form a new "share", which is the main threat that haunts some public-key based traitor tracing schemes [12]. For example, the legal users $z+i$ and $z+j$, $i \neq j \geq 1$, can combine their shares to form a new "share"

$$(a+b, ax_{z+i} + bx_{z+j}, \ldots, ax_{z+i}^z + bx_{z+j}^z, af(x_{z+i}) + bf(x_{z+j})). \tag{1}$$

By the new share and the shares $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_{z-1}, f(x_{z-1}))$, one can compute a_0 by solving the system of equations:

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^z \\ 1 & x_1 & \cdots & x_1^z \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{z-1} & \cdots & x_{z-1}^z \\ a+b & ax_{z+i} + bx_{z+j} & \cdots & ax_{z+i}^z + bx_{z+j}^z \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_z \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{z-1}) \\ af(x_{z+i}) + bf(x_{z+j}) \end{pmatrix}$$

We observe that if a pirate P gets a share by linear combination of m shares of traitors $j_1, j_2, \ldots, j_m, m \leq z$, then P and the traitors together cannot compute a_0 or g^{ra_0}. We base our traitor tracing algorithm on this observation. In our system, we give each user i a share $(x_i, f(x_i))$. If we suspect that users $j_1, j_2, \ldots, j_m, m \leq z$, are traitors, we broadcast the cipher block $E'(s, M)$ and the enabling block

$$\langle sg^{ra_0}, g^r, (x_{j_1}, g^{rf(x_{j_1})}), \ldots, (x_{j_m}, g^{rf(x_{j_m})}), (l_1, g^{rf(l_1)}), \ldots, (l_{z-m}, g^{rf(l_{z-m})})\rangle,$$

where $l_1, l_2, \ldots, l_{z-m}$ are arbitrarily chosen and different from $x_{j_1}, x_{j_2}, \ldots, x_{j_m}$. A user who is not a traitor can compute g^{ra_0} and thus s. We confirm that $j_1, j_2, \ldots, j_m$ are traitors if they together cannot decrypt the cipher block properly.

Decisional Diffie-Hellman problem. Let G_q be a group of a large prime order q. Consider the following two distribution ensembles R and D:

- $R = (g_1, g_2, u_1, u_2) \in G_q^4$, where g_1 and g_2 are generators of G_q;
- $D = (g_1, g_2, u_1, u_2)$, where g_1 and g_2 are generators of G_q and $u_1 = g_1^r$ and $u_2 = g_2^r$ for $r \in Z_q$.

The DDH problem is to distinguish the distribution ensembles R and D. That is, we would like to find a probabilistic polynomial-time algorithm A such that, for some positive constant c and all sufficiently large complexity parameter n,

$$|Pr[A(R_n) = 1] - Pr[A(D_n) = 1]| \geq 1/n^c,$$

where R_n and D_n are the size-n distributions of R and D, respectively.

Semantic security against passive adversary. Let PK be the public key of the encryption scheme and m_0 and m_1 be any two messages. The encryption scheme is *semantic secure* against passive adversary if there no probabilistic polynomial-time algorithm $\mathscr{A}$ that takes as input PK, m_0, m_1 and ciphertext c, and determines c's source with a successful probability significantly better than 0.5, where c is encrypted from m_0 and m_1 with equal probability. That is, for any probabilistic polynomial-time algorithm $\mathscr{A}$, any $k > 0$, and large enough security parameter n, we have

$$\Pr_{b \in \{0,1\}, c = E(PK, m_b)}[\mathscr{A}(PK, m_0, m_1, c) = b] \leq 0.5 + 1/n^k.$$

3. Definitions

A traitor tracing scheme consists of the following functions.

- **System setup.** The sender sets up system algorithms and parameters.
- **Registration.** After system setup, a receiver can register to the system and gets a decoder that contains a private key specific to the decoder. A decoder with a legal private key can decode the ciphertext broadcast by the sender.

- **Encryption.** When the sender wants to send M, it uses the secret-key cipher E' and a session key s to encrypt M as a cipher block $C = E'(s, M)$ and embeds s into the enabling block T.
- **Decryption.** A decoder consists of a decryption program and a private key such that it can decrypt $\langle T, C \rangle$ to get the message M.
- **Traitor tracing.** The sender wants to determine the original owner of the private key in a pirate decoder. It may be that some legal receivers conspire to compute some key that is not legal, but able to decrypt the ciphertext, maybe with a different decryption program. The traitor tracing algorithm need reveal at least one conspirator's identity. If traitor tracing is done by observing the input-output relation of the decoder, it is called *black-box* tracing.
  A traitor tracing scheme is *k-resilient* if it can find at least one traitor among the k or less traitors who create the pirate decoder. It is *fully k-resilient* if it can find all of them.

Note. In order to simplify presentation, we omit the security parameter (or complexity measure) n from the related parameters. For example, when we say a probability ϵ is negligible, we mean that for any positive constant c, $\epsilon = \epsilon(n) < 1/n^c$ for large enough n. A probability δ is overwhelming if $\delta = 1 - \epsilon$ for some negligible probability ϵ.

4. The Public-Key Traitor Tracing Scheme

In this section we present our traitor tracing scheme. Let k be the maximum number of colluded receivers (traitors) and z be the revocation threshold, i.e., at most z private keys of traitors can be revoked. We set $z \geq 2k$.

System setup. Let G_q be a group of a large prime order q. The sender selects a degree-z polynomial $f(x) = \sum_{t=0}^{z} a_t x^t \pmod{q}$ with coefficients over Z_q. The sender's secret key is $f(x)$ and his public key is

$$\langle g, g^{a_0}, g^{f(1)}, \ldots, g^{f(z)} \rangle,$$

by which a receiver can verify his private key.

Registration. When a receiver $i, i > z$, registers, the sender gives the receiver i a decoder with the share (private key) $(i, f(i))$. The receiver i verifies his key by checking

$$g^{a_0} = \prod_{t=0}^{z} g^{f(x_t)\lambda_t},$$

where $x_0 = 1, x_1 = 2, \ldots, x_{z-1} = z, x_z = i$. If it is so, the receiver i gets a decoder with the private key $(i, f(i))$.

Hereafter, we call $(j, f(j))$ an *unused share* if it has not been assigned to any receiver. Sometimes, we refer "j" as a share.

Encryption. The sender randomly selects z unused shares

$$(j_1, f(j_1)), (j_2, f(j_2)), \ldots, (j_z, f(j_z))$$

a random number $r \in Z_q$, and a session key s. The sender computes the *enabling block*

$$T = \langle sg^{ra_0}, g^r, (j_1, g^{rf(j_1)}), (j_2, g^{rf(j_2)}), \ldots, (j_z, g^{rf(j_z)})\rangle,$$

and broadcasts $\langle T, E'(s, M)\rangle$, where E' is a secret-key cipher, such as DES.

Decryption. When receiving $\langle T, E'(s, M)\rangle$, the receiver i computes s by

$$sg^{ra_0}/[(g^r)^{f(i)\lambda_z} \cdot \prod_{t=0}^{z-1} (g^{rf(x_t)})^{\lambda_t}] = sg^{ra_0}/g^{r(\sum_{t=0}^{z-1} f(x_t)\lambda_t + f(i)\lambda_z)}$$
$$= sg^{ra_0}/g^{ra_0} = s,$$

where $x_0 = j_1, x_1 = j_2, \ldots, x_{z-1} = j_z$ and $x_z = i$. He then uses s to decrypt $E'(s, M)$ to obtain M.

Traitor tracing. We present two black box traitor tracing algorithms. Assume that n receivers $\{t_1, t_2, \ldots, t_n\}, n \le k$, use their shares to create the confiscated pirate decoder.

Our first black-box traitor tracing algorithm T_1 is shown in Figure 1. For each receiver set $\{c_1, c_2, \ldots, c_m\}$, $m \le k$, we use their shares to create an enabling block

$$\langle sg^{ra_0}, g^r, (c_1, g^{rf(c_1)}), \ldots, (c_m, g^{rf(c_m)}), (j_1, g^{rf(j_1)}), \ldots, (j_{z-m}, g^{rf(j_{z-m})})\rangle,$$

where $j_1, j_2, \ldots, j_{z-m}$ are unused shares. As long as $\{t_1, t_2, \ldots, t_n\} \subseteq \{c_1, c_2, \ldots, c_m\}$, the pirate decoder is not able to decode the enabling block to get s assuming that

1. For every possible m-receiver set $\{c_1, c_2, \ldots, c_m\}$, $m \le k$,

   (a) Randomly select $z - m$ unused shares $\{j_1, \ldots, j_{z-m}\}$ and construct a test $\langle T, E'(s, M)\rangle$, where

   $$T = \langle sg^{ra_0}, g^r, (c_1, g^{rf(c_1)}), (c_2, g^{rf(c_2)}), \ldots, (c_m, g^{rf(c_m)}),$$
   $$(j_1, g^{rf(j_1)}), (j_2, g^{rf(j_2)}), \ldots, (j_{z-m}, g^{rf(j_{z-m})})\rangle.$$

   (b) Feed $\langle T, E'(s, M)\rangle$ to the decoder.

   (c) If the pirate decoder does not output correct M, $\{c_1, c_2, \ldots, c_m\}$ is a possible traitor set.

2. Output the smallest of all possible traitor sets found in Step 1c.

Figure 1. Traitor tracing algorithm T_1

computing the discrete logarithm over G_q is hard. For this case, $\{c_1, c_2, \ldots, c_m\}$ is a possible traitor set. Then, the smallest possible traitor set is $\{t_1, t_2, \ldots, t_n\}$.

Our second traitor tracing algorithm T_2 works for the case that the pirate decoder's key is a linear combination of shares of $t_1, t_2, \ldots, t_n$. Let $\vec{v}$ be such a linear combination, as in equation (1). T_2 uses the opposite direction, that is, the pirate decoder can compute s from the enabling block. For each receiver set $\{c_1, c_2, \ldots, c_m\}$, $m \leq k$, we find a degree-z polynomial $h(x) = \sum_{t=0}^{z} b_t x^t$ that passes points $(c_1, f(c_1)), (c_2, f(c_2)), \ldots, (c_m, f(c_m))$. Polynomials $h(x)$ and $f(x)$ have m common points on $c_1, c_2, \ldots, c_m$ only. We use $h(x)$ to create a test enabling block

$$T = \langle s g^{r b_0}, g^r, (j_1, g^{r h(j_1)}), (j_2, g^{r h(j_2)}), \ldots, (j_z, g^{r h(j_z)}) \rangle.$$

and feed it to the pirate decoder, where $j_i \notin \{c_1, c_2, \ldots, c_m\}$, $1 \leq i \leq z$. If some t_i is not in $\{c_1, c_2, \ldots, c_m\}$, the pirate decoder cannot compute correct s from T since $\vec{v}$ consists of information from a share that is not from $h(x)$. If $\{t_1, t_2, \ldots, t_n\} \subseteq \{c_1, c_2, \ldots, c_m\}$, the pirate decoder can compute s from T since $\vec{v}$ is made of the information from shares of $h(x)$. For this case, $\{c_1, c_2, \ldots, c_m\}$ is a possible traitor set. Then, the smallest possible traitor set is $\{t_1, t_2, \ldots, t_n\}$. The traitor tracing algorithm T_2 is shown in Figure 2.

It may be that the pirate decoder tells difference between the test enabling block and the real one and refuses to respond. Therefore, we cannot identify the traitors. We show that the test enabling block and the real one are computationally indistinguishable. Therefore, this traitor tracing strategy works.

LEMMA 4.1. *For degree-z polynomials $f(x)$ and $h(x)$, the distributions of the enabling blocks constructed by $f(x)$ and $h(x)$ are computationally indistinguishable assuming that the DDH problem is hard.*

---

1. For every possible m-receiver set $\{c_1, c_2, \ldots, c_m\}$, $m \leq k$,
   (a) Randomly select a degree-z polynomial $h(x) = \sum_{i=0}^{z} a_i z^i$ that passes $(c_1, f(c_1)), (c_2, f(c_2)), \ldots, (c_m, f(c_m))$ points.
   (b) Randomly select z unused shares $j_1, j_2, \ldots, j_z$ and construct a test $\langle T, E'(s, M) \rangle$, where
   $$T = \langle s g^{r a_0}, g^r, (j_1, g^{r h(j_1)}), (j_2, g^{r h(j_2)}), \ldots, (j_z, g^{r h(j_z)}) \rangle.$$
   (c) Feed $\langle T, E'(s, M) \rangle$ to the decoder.
   (d) If the pirate decoder outputs correct M, $\{c_1, c_2, \ldots, c_m\}$ is a possible traitor set.
2. Output the smallest of all possible traitor sets found in Step 1c.

Figure 2. Traitor tracing algorithm T_2

Proof. Note that the distinguisher does not know $f(x)$ and $h(x)$. Let g be a fixed generator of G_q and $a \in_R S$ denote that a is chosen from the set S uniformly and independently. Consider the following 3 distributions:

1. $D_1 = \langle S, g^r, (c_1, g_1^r), (c_2, g_2^r), \ldots, (c_z, g_z^r)\rangle$, where $r \in_R Z_q$, $S \in_R G_q$, $c_i \in_R G_q$, $g_i = g^{f(c_i)}$. This is the enabling block constructed by $f(x)$.
2. $R = \langle S, g^r, (c_1, u_1), (c_2, u_2), \ldots, (c_z, u_z)\rangle$, where $r \in_R Z_q$, $S \in_R G_q$, $c_i \in_R G_q$, $u_i \in_R G_q$, $1 \le i \le z$.
3. $D_2 = \langle S, g^r, (c_1, h_1^r), (c_2, h_2^r), \ldots, (c_z, h_z^r)\rangle$, where $r \in_R Z_q$, $S \in_R G_q$, $c_i \in_R G_q$, $h_i = g^{h(c_i)}$. This is the enabling block constructed by $h(x)$.

By the DDH assumption, there are no polynomial-time algorithms to distinguish between D_1 and R (and R and D_2). Therefore, D_1 and D_2 are computationally indistinguishable. ■

Complexity. It takes $O(z)$ modular exponentiations to create an enabling block. This can be pre-computed by the sender. It takes also $O(z)$ modular exponentiations to decrypt an enabling block by each receiver. The traitor tracing algorithm runs in $O(C_k^n)$ modular exponentiations, where n is the number of receivers. When $n \gg k$, the runtime is about $O(n^k)$.

Each receiver holds only one private key. The size of an enabling block is $O(z)$, which is independent of the number of receivers.

4.1. Revocation of Traitors

After a pirate decoder is confiscated and the traitors are revealed, we would like to revoke the private keys of the traitors since thousands of copies of the pirate decoder may be sold.

Assume that $C = \{c_1, c_2, \ldots, c_m\}$, $m \le z$, is the set of found traitors. We can revoke their shares without updating the private keys of receivers. To send out M to receivers, instead of randomly choosing m unused shares for the enabling block, the sender fixes the first m shares as

$$(c_1, g^{rf(c_1)}), (c_2, g^{rf(c_2)}), \ldots, (c_m, g^{rf(c_m)})$$

and randomly chooses $z - m$ unused shares

$$(j_1, g^{rf(j_1)}), (j_2, g^{rf(j_2)}), \ldots, (j_{z-m}, g^{rf(j_{z-m})}).$$

to form the enabling block. The revoked traitors cannot decrypt the enabling block since their shares are in the enabling block. We can revoke at most z traitors totally before updating the shares of receivers.

We can see that to revoke receivers, the sender need not broadcast any revocation message and the receivers do nothing.

4.2. Restoration of a Revoked Key

If for some reason we would like to restore the decryption privilege of a revoked receiver, we simply do not use his share in the enabling block. The restored key can decrypt the broadcast ciphertext again.

4.3. Revocation Beyond the Threshold

It is possible to revoke more than z traitors. Assume that each pirate decoder contains only one pirate share. The idea is that if a pirate decoder can get at most $c\%$ of M, the partial part of M is useless [1]. For example, if a pirate decoder can only decrypt 95% of a movie, the traitor is revoked de facto.

Assume that $C = \{c_1, c_2, \ldots, c_m\}$, $m > z$, is the set of found traitors. To broadcast M to non-revoked receivers, we partition M as $M_1 || M_2 || \cdots || M_l$. For each M_i, $1 \le i \le l$, we construct an enabling block T_i with shares

$$(c_{i_1}, g^{rf(c_{i_1})}), (c_{i_2}, g^{rf(c_{i_2})}), \ldots, (c_{i_r}, g^{rf(c_{i_z})}),$$

where $c_{i_1}, c_{i_2}, \ldots, c_{i_z}$ are chosen from C.

With appropriately chosen l and c, each traitor in C can decrypt at most $c\%$ of M. If c_i appears in $(1-c\%)l$ enabling blocks, the traitor c_i can decrypt $c\%$ of M. Therefore, we have $lz/m \ge (1-c\%)l$, where lz is the total number of shares that may appear in the enabling blocks. We have $m \le z/(1-c\%)$. In particular, when $c\% = 95\%$, we can increase the revocation capability by 20 folds by partitioning M into 20 blocks. That is, each revoked share appears in one enabling block. If we partition M into $20t$ blocks and put each revoked share into t enabling blocks, one in every 20 blocks, a revoked traitor cannot decrypt one block in every 20 blocks.

4.4. Further Revocation

If we want to permanently revoke more keys beyond the revocation capability of the system, we can update the private keys of non-revoked receivers. Though, this work may be costly. The idea is to update the system's polynomial $f(x)$ as $f'(x) = f(x) + h(x)$, where $h(x)$ is also a degree-z polynomial. Then, each non-revoked receiver gets its new private key $(i, f'(i))$ as follows. Assume that there is a public bulletin board and $c_1, c_2, \ldots, c_{mz}$ are the receivers to be revoked.

1. The sender selects degree-z polynomials $h_j(x)$, $0 \le j \le m-1$, and sets the system's polynomial as $f'(x) = f(x) + \sum_{j=0}^{m-1} h_j(x)$.
2. The sender publishes the enabling blocks T_j, $0 \le j \le m-1$,

$$\langle s_j g^{r_j a_0}, g^{r_j}, (c_{jz+1}, g^{rf(c_{jz+1})}), (c_{jz+2}, g^{rf(c_{jz+2})}), \ldots, (c_{jz+z}, g^{rf(c_{jz+z})}) \rangle$$

in the bulletin board, where $s_j = h_j(x)$.

3. Each non-revoked receiver i computes $h_j(x)$, $1 \le j \le m-1$, from the enabling blocks in the bulletin board and computes its new private key $f'(i) = f(i) + \sum_{j=0}^{m-1} h_j(i)$.

We can see that the revoked receiver c_i cannot compute $h_{\lceil i/z \rceil - 1}(x)$, $1 \le i \le mz$. Therefore, it cannot update its private key from $(i, f(c_i))$ to $(i, f'(c_i))$. Thus, it is revoked permanently.

The total messages in the bulletin board is of length $O(mz)$.

4.5. Speedup of Tracing

Since the runtime of the traitor tracing algorithm is $O(C_k^n)$, when n or k is large, the algorithm is not efficient. In practice, we would like to have a more efficient traitor tracing algorithm.

A practical solution to this problem is to group receivers into classes $C_1, C_2, \ldots, C_r$. Each class C_i consists of a reasonable number of receivers. For each class C_i, the sender uses a different polynomial $f_i(x)$ as the secret key. A receiver j in class C_i is given the share $(j, f_i(j))$. The sender encrypts message M using the secret key $f_i(x)$. The decryption and tracing algorithms are the same as the original ones except that the keys are different for different classes.

Grouping receivers can make our revocation mechanism more practical. It will be less frequent to revoke the receivers in a class since a class consists of less receivers. Even if the sender wants to revoke more than z receivers in a class, only the private keys of the non-revoked receivers in the class have to be updated.

5. Security Analysis

We consider both semantic security and security against the z-coalition attack, in which any coalition of z or less legal receivers cannot compute a legal private key for decryption.

Before we proceed, we first address the framing problem [3]. We show that it is not possible for two disjoint sets of k receivers to construct the same "new" share by linear combination. Therefore, framing is not possible by linear combination of shares in our scheme.

LEMMA 5.1. *Let $C = \{c_1, c_2, \ldots, c_k\}$ and $D = \{d_1, d_2, \ldots, d_k\}$ be two disjoint receiver sets. All linear combination of shares of C and those of D are different except the zero point.*

Proof. We can represent a share i as a $z+2$-dimensional vector

$$v_i = (1, i, i^2, \ldots, i^z, f(i)).$$

Since it is a point of a degree-z polynomial, any $z+1$ different shares are linearly independent. If one can use the shares of C and the shares of D to construct the same non-zero share by linear combination, we have

$$\sum_{i=1}^{k} a_i v_{c_i} = \sum_{i=1}^{k} b_i v_{d_i} \neq \mathbf{0}.$$

Therefore, we have

$$\sum_{i=1}^{k} a_i v_{c_i} - \sum_{i=1}^{k} b_i v_{d_i} = \mathbf{0}.$$

This is a contradiction since not all a_i's and b_i's are zero and $C \cup D$ is linearly independent. ■

The encryption algorithm of our scheme is semantically secure against a passive adversary if the DDH problem in G_q is hard (or computationally infeasible). Recall that $D = \langle g_1, g_2, g_1^r, g_2^r \rangle$ and $R = \langle g_1, g_2, g_1^a, g_2^b \rangle$, where g_1, g_2 are generators and a, b and r are randomly chosen over Z_q.

THEOREM 5.2. (Semantic security) *Assume that the DDH problem is hard. The encryption algorithm of our traitor tracing scheme is semantically secure against the passive adversary.*

Proof. Suppose that our encryption algorithm is not semantically secure against the passive adversary. We show that there is a probabilistic polynomial-time algorithm $\mathscr{B}$ that distinguishes between D and R with a non-negligible advantage ε.

Assume that adversary $\mathscr{A}$ attacks our encryption algorithm successfully in terms of semantic security. $\mathscr{A}$ has two procedures A_1 and A_2. Given the public key $\langle g, g^{a_0}, g^{f(1)}, \ldots, g^{f(z)} \rangle$ of the sender, A_1 finds two session keys s_0 and s_1 in G_q such that A_2 can distinguish them by observing the enabling block.

Let $\langle g_1, g_2, u_1, u_2 \rangle$ be an input of the DDH problem. The following algorithm $\mathscr{B}$ shall decide whether $\langle g_1, g_2, u_1, u_2 \rangle$ is from D or R.

1. Randomly choose $a_i \in Z_q$, $1 \leq i \leq z$, and let $f'(x) = \sum_{t=1}^{z} a_t x^t$. Let $g = g_1$, $g^{a_0} = g_2$, $g^{f(1)} = g_2 g_1^{f'(1)}, \ldots, g^{f(z)} = g_2 g_1^{f'(z)}$, where $f(x) = f'(x) + a_0$. Note that we don't know a_0.
2. Feed the public key $\langle g, g^{a_0}, g^{f(1)}, \ldots, g^{f(z)} \rangle$ to A_1. A_1 returns s_0 and s_1 in G_q.
3. Randomly select $d \in \{0, 1\}$ and encrypt s_d as

   $$C = \langle s_d u_2, u_1, (j_1, u_2 u_1^{f'(j_1)}), \ldots, (j_z, u_2 u_1^{f'(j_z)}) \rangle$$

   where $j_1, j_2, \ldots, j_z$ are randomly chosen.
4. Feed C to A_2 and get a return d'. Then, the algorithm outputs 1 if and only if $d = d'$.

If $\langle g_1, g_2, u_1, u_2 \rangle$ is from D, $g = g_1$, $g_2 = g^{a_0}$, $u_1 = g^r$, $u_2 = g_2^r = g^{r a_0}$ and $u_2 u_1^{f'(j_i)} = g^{r f(j_i)}$ for $1 \leq i \leq z$. Thus, C is the encryption of s_d and $\Pr[\mathscr{B}(g_1, g_2, u_1, u_2) = 1] =$

$\Pr[A_2(C)=d]=1/2+\varepsilon$. Otherwise, since $u_1=g_1^a$ and $u_2=g_2^b$, the distribution of C is the same for $d=0$ and $d=1$. Thus, $\Pr[\mathscr{B}(g_1, g_2, u_1, u_2)=1]=\Pr[A_2(C)=d]=1/2$. Therefore, $\mathscr{B}$ distinguishes D from R with a non-negligible advantage ε. ■

The encryption algorithm of our scheme is secure against z-coalition assuming that computing the discrete logarithm is hard.

THEOREM 5.3. *Assume that computing the discrete logarithm over G_q is hard. No coalition of z or less legal receivers can compute the private key of another legal receiver with a non-negligible probability.*

Proof. Assume that the probabilistic polynomial-time algorithm $\mathscr{A}$ can compute a new share (private key) $(x_u, f(x_u))$ from the given public key $\langle g, g^{a_0}, g^{f(1)}, g^{f(2)}, \ldots, g^{f(z)}\rangle$ and z shares $(x_1, f(x_1)),\ldots, (x_z, f(x_z))$ with a non-negligible probability ε. We construct another probabilistic polynomial-time algorithm $\mathscr{B}$ to compute the discrete logarithm over G_q with an overwhelming probability.

Let (p, g, y) be the input of the discrete logarithm problem. The following algorithm $\mathscr{B}'$ computes $\log_g y \pmod p$ with a non-negligible probability. Let $y=g^{a_0}$ and $f(x)$ be the degree-z polynomial passing $(0, a_0)$ and $(x_i, f(x_i)), 1\le i\le z$. Note that we don't a_0 yet. By Lagrange's interpolation method, we can compute $g^{f(i)}, 1\le i\le z$, from g^{a_0} and $g^{f(x_i)}$, $1\le i\le z$. We feed the public key $\langle g, g^{a_0}, g^{f(1)}, g^{f(2)},\ldots, g^{f(z)}\rangle$ and z shares $(x_1, f(x_1)),\ldots, (x_z, f(x_z))$ to $\mathscr{A}$ and shall get a new share $(x_u, f(x_u))$ with a non-negligible probability. With the given z shares and $(x_u, f(x_u))$, we can compute $f(0)=a_0$.

By applying the randomized technique to $\mathscr{B}'$ for a polynomial number of times, we get $\mathscr{B}$. ■

6. Discussion

We can drop the sender's public key from our traitor tracing schemes if verification of private keys by receivers is not necessary. This is indeed the case for practicality. Thus, only the sender can send messages to the receivers. Since the enabling blocks are computationally indistinguishable from each other due to the DDH assumption, our scheme should be more secure.

For practicality, we can set $z=k$. In this case, there may be framing problem. The probability that a set of k receivers can frame a specific set of k receivers is $1/q$. Assume that there are $m=10,000,000$ receivers and k is set as 20. Then, the probability that a set of k receivers can frame some set of k receivers is $\le C_k^m/q\approx m^k/q\approx 1/(10)^{168}$, for q being 1024-bit long.

7. Conclusion

In this work we have proposed a new public-key traitor tracing scheme with revocation capability using dynamic shares. Its distinct feature of revoking private keys

makes the protocol highly practical. The scheme's traitor tracing algorithm is fully k-resilient and conceptually simple. The size of the enabling block is independent of the number of receivers.

Our scheme is semantically secure against the passive adversary assuming that the DDH problem is hard. We also present a variant scheme that is semantically secure against the adaptive chosen ciphertext attack assuming that the DDH problem is hard.

Acknowledgement

Research supported in part by the National Science Council grant NSC-89-2213-E-009-180 and by the Ministry of Education grant 89-E-FA04-1-4, Taiwan, ROC.

Notes

1. A preliminary version appeared in PKC 2001 [20].

References

1. M. Abdalla, Y. Shavitt and A. Wool, Key management for restricted multicast using broadcast encryption, In *Proc. of Financial Cryptology 99*, Lecture Notes in Computer Science 1648, Springer-Verlag (1999).
2. J. Anzai, N. Matsuzaki and T. Matsumoto, A quick group key distribution scheme with "entity revocation", In *Proc. of Advances in Cryptology - Asiacrypt 99*, Lecture Notes in Computer Science 1716, Springer-Verlag (1999) pp. 333–347.
3. D. Boneh and M. Franklin, An efficient public key traitor tracing scheme, *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, Springer-Verlag (1999) pp. 338–353.
4. D. Boneh and J. Shaw, Collusion-secure fingerprinting for digital data, *IEEE Transaction on Information Theory* 44(5), 1998. In *Proc. of Advances in Cryptology - Crypto 95*, Lecture Notes in Computer Science 963, pp. 452–465, Springer-Verlag (1995) pp. 1897–1905.
5. R. Canetti, T. Malkin and K. Nissim, Efficient communication-storage tradeoffs for multicast encryption, In *Proc. of Advances in Cryptology - Eurocrypt 99*, Lecture Notes in Computer Science 1592 (1999) pp. 459–474.
6. B. Chor, A. Fiat and M. Naor, Tracing traitors, In *Proc. of Advances in Cryptology - Crypto 94*, Lecture Notes in Computer Science 839, Springer-Verlag (1994) pp. 257–270.
7. T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory* 31(4) (1985) pp. 469–472.
8. A. Fiat and T. Tassa, Dynamic traitor tracing, In *Proc. of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, Springer-Verlag (1999) pp. 354–371.
9. A. Fiat and M. Naor, Broadcast encryption, In *Proc. of Advances in Cryptology - Crypto 93*, Lecture Notes in Computer Science 773, Springer-Verlag (1993) pp. 480–491.
10. E. Gafni, J. Staddon and Y. L. Yin, Efficient methods for integrating traceability and broadcast encryption, In *Proc. of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, Springer-Verlag (1999) pp. 372–387.
11. R. Kumar, S. Rajagopalan and A. Sahai, Coding constructions for blacklisting problems without computational assumptions, In *Proc. of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, Springer-Verlag (1999) pp. 609–623.

12. K. Kurosawa and Y. Desmedt, Optimum traitor tracing and asymmetric schemes, In *Proc. of Advances in Cryptology - Eurocrypt 98*, Lecture Notes in Computer Science 1403, Springer-Verlag (1998) pp. 145–157.
13. K. Kurosawa and Y. Yoshida, Linear code implies public-key traitor tracing, In *Proc. of the 5th International Workshop on Practive and Theory in Public Key Cryptosystems (PKC 02)*, Lecture Notes in Computer Science 2274, Springer-Verlag (2002) pp. 172–187.
14. M. Luby and J. Staddon, Combinatorial bounds for braodcast encryption, In *Proc. of Advances of Cryptology - Eurocrypt 98*, Lecture Notes in Compouter Science 1403, Springer-Verlag (1998) pp. 512–526.
15. M. Naor and B. Pinkas, Threshold traitor tracing, In *Proc. of Advances in Cryptology - Crypto 98*, Lecture Notes in Computer Science 1462, Springer-Verlag (1998) pp. 502–517.
16. M. Naor and B. Pinkas, Efficient trace and revoke schemes, In *Proc. of Financial Cryptography* 00 (2000).
17. B. Pfitzmann, Trials of traced traitors, In *Proc. of Workshop on Information Hiding*, Lecture Notes in Computer Science 1174, Springer-Verlag (1996) pp. 49–64.
18. B. Pfitzmann and M. Waidner, Asymmetric fingerprinting for large collusions, In *Proc. of ACM Conference on Computer and Communication Security*, (1997) pp. 151–160.
19. D. R. Stinson and R. Wei, Combinatorial properties and constructions of traceability schemes and frameproof codes, *SIAM J. on Discrete Math* 11(1) (1998) pp. 41–53.
20. W.-G. Tzeng and Z.-J. Tzeng, A public-key traitor tracing scheme with revocation using dyanmic shares, In *Proc. of the 4th International Workshop on Practive and Theory in Public Key Cryptosystems (PKC 01)*, Lecture Notes in Computer Science 1992, Springer-Verlag (2001) pp. 207–224.
21. C. K. Wong, M. Gouda and S. Lam, Secure group communications using key graphs, In *Proc. of ACM SIGCOMM '98* (1998) pp. 68–79.
22. M. Yoshida and T. Fujiwara, An efficient traitor tracing scheme for broadcast encryption, In *Proc. of 2000 IEEE International Symposium on Information Theory* (2000) pp. 463.