Linear Code Implies Public-Key Traitor Tracing

Kaoru Kurosawa[1] and Takuya Yoshida[2]

[1]Department of Computer and Information Sciences,
Ibaraki University, Japan
kurosawa@cis.ibaraki.ac.jp
[2]Department of Communications and Integrated Systems,
Tokyo Institute of Technology, Japan
takuya@crypt.ss.titech.ac.jp

Abstract. In this paper, we first show that three public-key (k, n)-traceability schemes can be derived from an $[n, u, d]$-linear code $\mathcal{C}$ such that $d \geq 2k + 1$. The previous schemes are obtained as special cases. This observation gives a more freedom and a new insight to this field. For example, we show that Boneh-Franklin scheme is equivalent to a slight modification of the corrected Kurosawa-Desmedt scheme. This means that BF scheme is redundant or overdesigned because the modified KD scheme is much simpler. It is also shown that the corrected KD scheme is the best among them.

1 Introduction

In such applications as pay TV, CD-ROM distribution and online databases, data should only be available to authorized users. To prevent unauthorized users from accessing data, the data supplier will encrypt data and provide only the authorized users with personal keys to decrypt it. However, some authorized users (*traitors*) may create a pirate decoder.

A (k, n)-traceability scheme is a scheme in which at least one traitor is detected from a confiscated pirate decoder if there are at most k traitors among n authorized users. Chor, Fiat and Naor [4] introduced the first (k, n)-traceability scheme. Their scheme is, however, non-constructive. Stinson and Wei showed some explicit constructions by using combinatorial designs [10]. In the above two schemes, a private-key encryption scheme is used to encrypt a session key.

On the other hand, the first public-key (k, n)-traceability scheme was shown by Kurosawa and Desmedt [6, Sec.5]. That is, anyone can broadcast encrypted data to authorized users. Although Shamir's $(k + 1, n)$-threshold secret sharing scheme was used in their original scheme, we should use Shamir's $(2k - 1, n)$-threshold secret sharing scheme to avoid a linear attack given by [10]. We call such a corrected scheme the corrected KD scheme.

After that, Boneh and Franklin presented another public-key (k, n)-traceability scheme [2]. Only the above two schemes are known as public-key (k, n)-traceability schemes currently.

In this paper, we first show that three public-key (k, n)-traceability schemes can be derived from an $[n, u, d]$-linear code $\mathcal{C}$ such that $d \geq 2k + 1$. We call them linear coded KD scheme (LC-KD scheme), linear coded BF scheme (LC-BF scheme) and linear coded KD' scheme (LC-KD' scheme), respectively. The previous schemes are obtained as special cases. This observation gives a more freedom and a new insight to the study of this field.

For example, we show that Boneh-Franklin scheme (BF scheme) is equivalent to a slight modification of the corrected KD scheme. (We call it modified KD scheme. It will be given in Sec.5.3.) This means that BF scheme is redundant or overdesigned because modified KD scheme is much simpler. Indeed, BF scheme must use a public code matrix Γ and $2k$ additional secret random numbers $\beta_1, \cdots, \beta_{2k}$ which modified KD scheme does not require. More generally, we prove the equivalence between LC-BF scheme and LC-KD' scheme.

We also show that LC-KD scheme is better than LC-KD' scheme from a view point of key generation. This implies that the corrected KD scheme is better than modified KD scheme from a view point of key generation. Since modified KD scheme is better than BF scheme as shown above, we see that the corrected KD scheme is the best among them.

We finally prove the secrecy and the black box traceability of LC-KD scheme under the decision Deffie-Hellman assumption. Those of LC-KD' scheme and LC-BF scheme are proved similarly. The tracing algorithm of BF scheme for any pirate decoder is obtained as a special case. (It is not written clearly in the original paper [2]. It is not written at all in their latest version [3].)

Generalized Scheme	Original Scheme
LC-KD scheme	corrected KD scheme
LC-KD' scheme	modified KD scheme
LC-BF scheme	BF scheme

2 Preliminaries

2.1 Notation

An $[n, u, d]$-linear code is a linear code of length n, dimension u and the minimum Hamming distance d.

Let $q > n$ be a prime. Let G_q be a group of prime order q. Let $g \in G_q$ be a generator of G_q. For example, G_q is a subgroup of Z_p^* of order q, where $q \mid p-1$. Alternatively, we can use an elliptic curve over a finite field.

$\cdot$ denotes the inner product of two vectors over $GF(q)$.

2.2 DDH Assumption

The decision Diffie-Hellman assumption (DDH) assumption says that no polynomial statistical test can distinguish with non negligible advantage between the two distributions $\mathbf{D} = (g, g^r, y, y^r)$ and $\mathbf{R} = (g, g^r, y, v)$, where g, y, v are chosen at random from G_q and r is chosen at random in Z_q.

2.3 Model of Traitor Tracing

In the model of traceability schemes, there are a data supplier T, a set of n authorized users and a pirate user. Some authorized users are malicious and they are called *traitors*. The traitors create a pirate key e_p. The pirate key is used in a pirate decoder.

Suppose that there are at most k traitors. Then a (k, n)-traceability scheme is a scheme such that at least one traitor is detected from a confiscated pirate decoder. A (k, n)-traceability scheme has four components.

Key Generation: The key generation algorithm $\mathcal{K}$ is a probabilistic polynomial time algorithm that outputs $(e_T, e_1, \cdots, e_n)$ on input 1^l, where l is the security parameter. e_T is the broadcast encryption key of the data supplier T and e_i is the personal decryption key of authorized user i.

T runs $\mathcal{K}$ and sends e_i to authorized user i secretly.

Encryption: The encryption algorithm $\mathcal{E}$ is a probabilistic polynomial time algorithm that takes an encryption key e_T and a session key s to return a header h; we write

$$h \stackrel{R}{\leftarrow} e_T(s).$$

The data m is encrypted by using a secure symmetric encryption function E with the session key s as $E_s(m)$. Finally, T broadcasts $(h, E_s(m))$.

Decryption: Then decryption algorithm $\mathcal{D}$ is a deterministic algorithm that takes the personal decryption key e_i and a header h to return the session key s; we write

$$s \leftarrow e_i(h).$$

Each authorized user i can recover s from h by using his personal key e_i and then decrypt $E_s(m)$ to obtain the data m.

Tracing: T can detect at least one traitor from a pirate key e_p by using a tracing algorithm. We have *black box* traceability if the pirate decoder can only be used as an oracle. That is, the tracing algorithm cannot examine the pirate key e_p. For black box tracing, we shall assume that the pirate decoder is resettable to its initial state, as in [5].

In what follows, a session key s is chosen from G_q.

3 Previous Public-Key (k, n) Traceability Schemes

3.1 Corrected Kurosawa-Desmedt Scheme

Key Generation: The data supplier T chooses a uniformly random polynomial $f(x) = a_0 + a_1 x + \cdots + a_{2k-1} x^{2k-1}$ over $GF(q)$. Then T gives to each authorized user i the personal decryption key $e_i = f(i)$, where $i = 1, 2, \ldots, n$. He next publishes g and $y_0 = g^{a_0}, y_1 = g^{a_1}, \ldots, y_{2k-1} = g^{a_{2k-1}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (g^r, sy_0^r, y_1^r, \ldots, y_{2k-1}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using $f(i)$.

$$s = U/(g^r)^{f(i)}, \text{ where } U = sy_0^r \prod_{j=1}^{2k-1} (y_j^r)^{i^j}.$$

3.2 Boneh-Franklin Scheme

BF scheme makes use of a public code matrix Γ defined as follows. Consider the following $(n-2k) \times n$ matrix G:

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ 1^2 & 2^2 & 3^2 & \cdots & n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1^{n-2k-1} & 2^{n-2k-1} & 3^{n-2k-1} & \cdots & n^{n-2k-1} \end{pmatrix} \pmod{q}$$

Let $w_1, \ldots, w_{2k}$ be a basis of the linear space of vectors satisfying

$$G\mathbf{x} = 0 \bmod q. \tag{1}$$

Viewing these $2k$ vectors as the columns of a matrix, we obtain an $n \times 2k$ matrix Γ:

$$\Gamma = \begin{pmatrix} | & | & | & & | \\ w_1 & w_2 & w_3 & \cdots & w_{2k} \\ | & | & | & & | \end{pmatrix}$$

Define the code as the set of rows of the matrix Γ. Hence, it consists of n codewords each of length $2k$.

Key Generation: For $i = 1, \ldots, 2k$, the data supplier chooses a random $a_i \in Z_q$ and compute $y_i = g^{a_i}$. Then T computes $z = \prod_{i=1}^{2k} y_i^{\beta_i}$ for random $\beta_1, \ldots, \beta_{2k} \in Z_q$ and publishes $z, y_1, \ldots, y_{2k}$ as the public key. The personal decryption key of user i is computed as

$$\theta_i = (\sum_{j=1}^{2k} a_j \beta_j)/(\sum_{j=1}^{2k} a_j \gamma_j) \pmod{q},$$

where $\gamma^{(i)} = (\gamma_1, \ldots, \gamma_{2k}) \in \Gamma$ is the i'th codeword of Γ.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (sz^r, y_1^r, \ldots, y_{2k}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using θ_i.

$$s = sz^r/U^{\theta_i}, \text{ where } U = \prod_{j=1}^{2k} (y_j^r)^{\gamma_j}.$$

Remark 3.1. In the key generation, $a_1, \cdots, a_{2k}$ must be chosen so that $\sum_{j=1}^{2k} a_j \gamma_j \neq 0 \pmod{q}$ for $i = 1, \cdots, n$. This was overlooked in [2].

4 Linear Code Implies Public-Key Traitor Tracing

This section shows that if there exists an $[n, u, d]$-linear code $\mathcal{C}$ such that $d \geq 2k + 1$, then three public-key (k, n)-traceability schemes are derived. We call them linear coded KD scheme (LC-KD scheme), linear coded BF scheme (LC-BF scheme) and linear coded KD' scheme (LC-KD' scheme), respectively. The corrected KD scheme and the original BF scheme are obtained as special cases.

Let H be a parity check matrix of an $[n, u, d]$-linear code over $GF(q)$ such that $d \geq 2k + 1$. Any $2k$ columns of H are linearly independent because $d \geq 2k + 1$. This property plays a central role in the proof of traceability of our schemes.

We assume that H is publicly known. Note that H is an $(n - u) \times n$ matrix over $GF(q)$. Let the ith column of H be $\mathbf{b}_i = (b_{1,i}, b_{2,i}, \cdots, b_{n-u,i})^T$.

4.1 LC-KD Scheme

Assume that the first row of H is $(1, \cdots, 1)$.

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random Let $(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H$. T gives e_i to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_1 = g^{a_1}, y_2 = g^{a_2}, \ldots, y_{n-u} = g^{a_{n-u}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (g^r, sy_1^r, y_2^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using e_i.

$$s = U/(g^r)^{e_i}, \text{ where } U = sy_1^r \prod_{j=2}^{n-u} (y_j^r)^{i^j}. \tag{2}$$

The tracing algorithm will be given in Sec.7.

4.2 LC-BF Scheme

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random in such a way that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ for $i = 1, \cdots, n$. The personal decryption key of user i is computed as

$$\theta_i = (a_1, \ldots, a_{n-u}) \cdot (\beta_1, \ldots, \beta_{n-u})/(a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i. \tag{3}$$

Next let $y_i = g^{a_i}$. Then T computes $z = \prod_{i=1}^{n-u} y_i^{\beta_i}$ for random $\beta_1, \ldots, \beta_{n-u} \in Z_q$ and publishes $z, y_1, \ldots, y_{n-u}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (sz^r, y_1^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using θ_i.

$$s = sz^r/U^{\theta_i}, \text{ where } U = \prod_{j=1}^{n-u} (y_j^r)^{b_{j,i}}. \tag{4}$$

4.3 LC-KD' Scheme

This is a slight modification of LC-KD scheme.

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random in such a way that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ for $i = 1, \cdots, n$. Let $(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H$. (Note that $e_i \neq 0$ for $i = 1, \cdots, n$.) T gives e_i to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_1 = g^{a_1}, y_2 = g^{a_2}, \ldots, y_{n-u} = g^{a_{n-u}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (sg^r, y_1^r, y_2^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows.

$$s = sg^r/U^{1/e_i}, \text{ where } U = \prod_{j=1}^{n-u} (y_j^r)^{b_{j,i}}. \tag{5}$$

Remark 4.1. In h, s is multiplied to g^r in LC-KD' scheme while it is multiplied to y_1^r in LC-KD scheme.

5 Relationship with the Original Schemes

5.1 Corrected KD Scheme

Let $\mathcal{C}$ be an $[n, n-2k, d]$-Reed Solomon code over $GF(q)$, where $d = 2k+1$. Then it is clear that the corrected KD scheme is obtained from LC-KD scheme as a special case.

5.2 BF Scheme

In BF scheme, note that G (shown in Sec.3.2) is a generator matrix of an $[n, n-2k, d]$ Reed-Solomon code over $GF(q)$. Further we see that $G \cdot \Gamma = \mathcal{O}$ from eq.(1). Hence Γ^T is a parity check matrix of the Reed-Solomon code $\mathcal{C}$. This implies that the original BF scheme is obtained from LC-BF scheme as a special case.

5.3 Modified KD Scheme

In LC-KD' scheme, let $\mathcal{C}$ be an $[n, n-2k, d]$-Reed Solomon code over $GF(q)$, where $d = 2k+1$. Then the following scheme is obtained. We call it modified KD scheme because it is a slight modification of the corrected KD scheme.

Key Generation: The data supplier T chooses a uniformly random polynomial $f(x) = a_0 + a_1 x + \cdots + a_{2k-1} x^{2k-1}$ over $GF(q)$ such that $f(i) \neq 0$ for $i = 1, \cdots, n$. Then T gives $f(i)$ to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_0 = g^{a_0}, y_1 = g^{a_1}, \ldots, y_{2k-1} = g^{a_{2k-1}}$.

Encryption: For a session key $s \in G_q$, T computes a header as
$h = (sg^r, y_0^r, y_1^r, \ldots, y_{2k-1}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using $f(i)$.

$$s = sg^r / U^{1/f(i)}, \text{ where } U = \prod_{j=0}^{2k-1} (y_j^r)^{i^j}. \tag{6}$$

Remark 5.1. In h, s is multiplied to g^r in modified KD scheme while it is multiplied to y_1^r in the corrected KD scheme.

6 Equivalence

6.1 LC-BF Scheme = LC-KD' Scheme

LC-BF scheme is more complicated than LC-KD' scheme because it uses secret random numbers $\beta_1, \cdots, \beta_{n-u}$ which LC-KD' scheme does not use. Nevertheless, we show that they are equivalent. This means that LC-BF scheme is redundant or overdesigned.

Public-key equivalence: In the key generation of LC-BF scheme, let

$$c = \sum_{i=1}^{n-u} a_i \beta_i.$$

For any fixed $(a_1, \cdots, a_{n-u})$, it is easy to see that $\Pr(c \neq 0) = 1 - (1/q)$. Therefore, we assume that $c \neq 0$ in what follows.

The public key of LC-BF scheme is $pk = (z, y_1, \ldots, y_{n-u})$. First since q is a prime and $z \in G_q$, z is a generator of G_q. Next note that

$$z = \prod_{i=1}^{n-u} y_i^{\beta_i} = \prod_{i=1}^{n-u} g^{a_i \beta_i} = z^c.$$

Let $a'_i = a_i/c$. Then we have

$$y_i = g^{a_i} = z^{a_i/c} = z^{a'_i}.$$

Now it is clear that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ if and only if $(a'_1, \cdots, a'_{n-u}) \cdot \mathbf{b}_i \neq 0$, where $i = 1, \cdots, n$. Therefore, the public key pk of LC-BF scheme is equivalent to that of LC-KD' scheme.

Header equivalence: Clear.

Decryption equivalence: In LC-BF scheme, from eq.(3), we obtain that

$$1/\theta_i = (a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i / c = (a'_1, \cdots, a'_{n-u}) \mathbf{b}_i.$$

On the other hand, in LC-KD' scheme,

$$e_i = (a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i.$$

Therefore, $1/\theta_i$ of LC-BF scheme is equivalent to e_i of LC-KD' scheme.

Secrecy equivalence: The same public key and the same header are used in both schemes. Therefore, the secrecy of LC-BF scheme against outside enemies is equivalent to that of LC-KD' scheme.

Traceability equivalence: Suppose that there exists a pirate decoder M_0 for LC-BF scheme which is not (black box) traceable. Then we show that there exists a pirate decoder M_1 for LC-KD' scheme which is not (black box) traceable. Let k traitors be $i_1, \cdots, i_k$ in both schemes.

Consider LC-KD' scheme in which a public key is $pk = (g, y_1, \cdots, y_{n-u})$ and the private key of user i is e_i. From the above equivalence, the same pk is used and the private key of user i is $\theta_i = 1/e_i$ in LC-BF scheme.

From our assumption, there exists an algorithm B which creates an untraceable pirate decoder M_0 from pk and $\theta_{i_1}, \cdots, \theta_{i_k}$ for LC-BF scheme.

Now in LC-KD' scheme, our traitors first create M_0 by running B on input pk and $1/e_{i_1}, \cdots, 1/e_{i_k}$. They then use M_0 as their pirate decoder M_1.

Finally it is easy to show that if there is a tracing algorithm which detects some traitor from M_1, then M_0 is also traceable. This contradicts our assumption. Hence, M_1 is not traceable.

The converse part is proved similarly.

Now we have proved the following theorem.

Theorem 6.1. *LC-BF scheme is equivalent to LC-KD' scheme.*

6.2 BF Scheme = Modified KD Scheme

From Theorem 6.1, we have the following equivalence.

Corollary 6.1. *BF scheme is equivalent to modified KD scheme.*

However, BF scheme is more complicated than the modified KD scheme because it must use a public code matrix Γ and $2k$ additional secret random numbers $\beta_1, \cdots, \beta_{2k}$. This means that BF scheme is redundant or overdesigned.

6.3 Comparison

We compare three schemes, LC-KD scheme, LC-BF scheme and LC-KD' scheme. We have seen that LC-BF scheme is equivalent to LC-KD' scheme, and hence redundant.

Now in LC-KD' scheme, $a_1, \cdots, a_{n-u}$ must be chosen in such a way that $e_i \neq 0$ for $i = 1, \cdots, n$, which LC-KD scheme does not require. This check is very inefficient if n is large. Therefore. LC-KD scheme is better than LC-KD' scheme from a view point of key generation.

Similarly, the corrected KD scheme is better than modified KD scheme from a view point of key generation. Further, modified KD scheme is better than BF scheme as shown in Sec.6.2. As a conclusion, we see that the corrected KD scheme is the best among them.

7 Secrecy and Traceability

In this section, we prove the secrecy and the traceability of LC-KD scheme, LC-KD' scheme and LC-BF scheme.

Note that any $2k$ columns of H are linearly independent because $d \geq 2k+1$.

7.1 Secrecy of LC-KD Scheme

Theorem 7.1. *LC-KD scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption.*

Proof. Similarly to the proof of [6, Theorem 14], we can show that the secrecy of LC-KD scheme is reduced to that of ElGamal encryption scheme. It is well known that ElGamal encryption scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption. The details will be given in the final paper.

7.2 Black Box Tracing Algorithm for LC-KD Scheme

Let BAD be the set of at most k traitors who created a confiscated pirate decoder. Let A be a subset of at most k users. We first describe a procedure TEST which checks whether $A \cap BAD \neq \emptyset$.

Suppose that $(e_T, e_1, \cdots, e_n)$ is being used as the key. For a random encryption key $e'_T = (a'_1, \cdots, a'_{n-m})$, let the corresponding private decryption keys be $(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H$. We say that e'_T matches with A if $e'_i = e_i$ for all $i \in A$.

TEST(A)
Step 1. T chooses e'_T which matches A randomly. (We can do this because any $2k$ columns of H are linearly independent.) He chooses a random session key s' and computes an *illegal* header $h' \stackrel{R}{\leftarrow} e'_T(s')$.
Step 2. T gives h' to the pirate decoder. Let the output of the pirate decoder be s_A.
Output:

$$TEST(A) = \begin{cases} 1 & \text{if } s_A = s' \\ 0 & \text{otherwise} \end{cases}$$

We next describe a procedure $TEST2(A, m)$ which runs $TEST(A)$ m times independently, where m is a sufficiently large positive integer.

TEST2(A, m)
Set *counter* $:= 0$. For $i = 1, 2, \ldots, m$, do
Step 1. Run $TEST(A)$ randomly.
Step 2. Let *counter* $:=$ *counter* $+ TEST(A)$. Reset the pirate decoder.
Output: $TEST2(A, m)$, the final value of *counter*.

We say that a set of users A is *marked* if $TEST2(A, m) = m$. We now present our tracing algorithm.

Black box tracing algorithm
Find a marked set $A = \{i_1, i_2, \ldots, i_k\}$ by exhaustive search. Suppose that $i_1 < i_2 < \cdots < i_k$. For $j = 1, 2, \ldots, k$, do:
Step 1. Let $B := A \setminus \{i_1, i_2, \ldots, i_j\}$. Run $TEST2(B, m)$.
Step 2. Let $m_j = TEST2(B, m)$.
Output: i_j such that $m_{j-1} - m_j$ is the maximum. (If there are more than one such j, choose one of them arbitrarily.) User i_j is a traitor.

Remark 7.1. By testing all the permutations on $\{i_1, i_2, \ldots, i_k\}$ (instead of $i_1 < i_2 < \cdots < i_k$), we can detect all traitors who are active in A. All active traitors are found by applying this process to all marked sets A.

7.3 Validity of Our Tracing Algorithm

We can show the validity of our tracing algorithm by using the following three *test conditions.*

(1) If $A \supseteq BAD$, then $\Pr(TEST(A) = 1)$ is overwhelming.

(2) If $A \cap BAD = \emptyset$, then $\Pr(TEST(A) = 1)$ is negligible.

(3) If $A \cap BAD \neq \emptyset$ and $A \setminus BAD \neq \emptyset$, then for any $i \in A \setminus BAD$,

$$|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{i\}) = 1)|$$

is negligible.

Theorem 7.2. *If the above three conditions are satisfied, then our black box tracing algorithm succeeds with overwhelming probability. That is user i_j is a traitor.*

Proof. If $A \supseteq BAD$, then $TEST2(A, m) = m$ with overwhelming probability from (1). Therefore, there exists at least one marked A. On the other hand, from (2), if $A \cap BAD = \emptyset$, then $TEST2(A, m) \ll m$. This means that if A is marked, then $A \cap BAD \neq \emptyset$.

Now suppose that A is marked. Let $m_0 = m$. It is easy to see that $m_k = 0$. If $m_{j-1} - m_j$ is the maximum, then $m_{j-1} - m_j \geq m/k$. On the other hand, if $j \in A \setminus BAD$, then $m_{j-1} - m_j \ll m/k$ from (3). Therefore, if $m_{j-1} - m_j$ is the maximum, then $i_j \in BAD$.

We finally show that LC-KD scheme satisfies the above three test conditions under the DDH assumption. We assume that a pirate decoder decrypts valid headers with overwhelming probability.

Theorem 7.3 (Test Condition (1)). *In LC-KD scheme, if $A \supseteq BAD$, then $\Pr(TEST(A) = 1)$ is overwhelming under the DDH assumption.*

Theorem 7.4 (Test Condition (2)). *In LC-KD scheme, if* $A \cap BAD = \emptyset$*, then* $\Pr(TEST(A) = 1)$ *is negligible under the DDH assumption.*

Theorem 7.5 (Test Condition (3)). *In LC-KD scheme, if* $A \cap BAD \neq \emptyset$ *and* $A \setminus BAD \neq \emptyset$*, then for any* $i \in A \setminus BAD$*,*

$$|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{i\}) = 1)|$$

is negligible under the DDH assumption.

The proofs will be given in Appendix.

7.4 Secrecy and Traceability of LC-KD' Scheme

The secrecy and the traceability of LC-KD' Scheme are proved similarly.

Theorem 7.6. *LC-KD' scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption.*

Theorem 7.7 (Test Condition (1)). *In LC-KD' scheme, if* $A \supseteq BAD$*, then* $\Pr(TEST(A) = 1)$ *is overwhelming under the DDH assumption.*

Theorem 7.8 (Test Condition (2)). *In LC-KD' scheme, if* $A \cap BAD = \emptyset$*, then* $\Pr(TEST(A) = 1)$ *is negligible under the DDH assumption.*

Theorem 7.9 (Test Condition (3)). *In LC-KD' scheme, if* $A \cap BAD \neq \emptyset$ *and* $A \setminus BAD \neq \emptyset$*, then for any* $i \in A \setminus BAD$*,*

$$|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{i\}) = 1)|$$

is negligible under the DDH assumption.

We show the proof of Theorem 7.8 in Appendix. The other theorems are proved similarly to those of LC-KD scheme.

7.5 Secrecy and Traceability of LC-BF Scheme

The secrecy and the traceability of LC-BF Scheme are equivalent to those of LC-KD' scheme as shown in Sec.6.1.

References

1. M. Bellare, A. Boldyreva and S. Micali, "Public-key encryption in a multi-user setting: Security proofs and improvements," *Proceedings of EUROCRYPT 2000*, LNCS 1807, Springer Verlag, pp.259–274, 2000.
2. D. Boneh and M. Franklin, "An efficient public key traitor tracing scheme (Extended Abstract)," *Proceedings of CRYPTO '99*, LNCS 1666, Springer Verlag, pp.338–353, 1999.

3. D. Boneh and M. Franklin, "An efficient public key traitor tracing scheme *(full-version of [2])*," `http://crypto.stanford.edu/~dabo/`, 2001.
4. B. Chor, A. Fiat, and M. Naor, "Tracing traitors," *Proceedings of CRYPTO '94*, LNCS 839, Springer Verlag, pages 257–270, 1994.
5. B. Chor, A. Fiat, and M. Naor, B. Pinkas, "Tracing traitors," *IEEE Transactions on Information Theory*, Vol.46, No.3, pp.893–910, 2000.
6. K. Kurosawa and Y. Desmedt, "Optimum traitor tracing and asymmetric schemes with arbiter," *Proceedings of EUROCRYPT '98*, LNCS 1403, Springer Verlag, pp.145–157, 1999.
7. M. Naor and O. Reingold, "Number theoretic constructions of efficient pseudo-random functions," *Proceedings of 38th IEEE Symposium on Foundations of Computer Science*, pp.458–467, 1997.
8. M. Stadler, "Publicly verifiable secret sharing," *Proceedings of EUROCRYPT '96*, LNCS 1070, Springer Verlag, pp.190–199, 1996.
9. D. Stinson and R. Wei, "Combinatorial properties and constructions of traceability schemes and frameproof codes," *SIAM Journal on Discrete Mathematics*, Vol.11, No.1, pp.41–53, 1998.
10. D. Stinson and R. Wei, "Key preassigned traceability schemes for broadcast encryption," *Proceedings of SAC'98*, LNCS 1556, Springer Verlag, pp.144–156, 1998.

A Proof of Theorem 7.3

By extending the result of Stadler [8, in the proof of Proposition 1] and Naor and Reingold [7, lemma 3.2], Bellare et al. proved the following proposition [1].

Proposition A.1. *[1] There is a probabilistic algorithm Σ such that on input g^a, g^b, g^c, Σ outputs $g^{b'}, g^{c'}$, where b' is random and*

$$c' = \begin{cases} ab' \bmod p & \text{if } c = ab \bmod p \\ \text{random} & \text{if } c \neq ab \bmod p \end{cases}$$

Σ runs in $O(T^{exp})$ time, where T^{exp} is the time needed to perform an exponentiation.

Now we show that

$$p_0 = |\Pr(P \text{ decrypts valid headers correctly}) - \Pr(TEST(A) = 1)|$$

is negligible for any pirate decoder P.

Suppose that $p_0 \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder such that $p_0 \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2k})$ and the private keys of BAD.

Now on input $d = (g, g'y, y')$, M works as follows.

1. Choose e_i at random for each $i \in A$ and let $e'_i = e_i$ for each $i \in A$.

2. Let $OUT = \{i_1, i_2, \ldots, i_k\}$ be a k-subset of users such that $OUT \cap A = \emptyset$.
3. For $j = 1, 2, \ldots, k$, M runs Σ of Proposition A.1 k times independently on input $d = (g, g', y, y')$. Let the output of Σ be $g^{e_{i_j}}, (g')^{e'_{i_j}}$.
4. Compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$ from $\{g^{e_i} \mid i \in OUT \cup A\}$, where
$$(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H.$$
Each a_i is written as a linear combination of $\{e_i \mid i \in A \cup BAD\}$ because any $2k$ columns of H are linearly independent and $|A \cup BAD| \leq 2k$. Therefore, we can do this.
5. Compute $(g')^{a'_1}, (g')^{a'_2}, \ldots, (g')^{a'_{n-u}}$ from $\{(g')^{e'_i} \mid i \in OUT \cup A\}$, where
$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$
6. Select a random session key s' and compute h' as follows.
$$h' = (g', s'(g')^{a'_1}, (g')^{a'_2}, \ldots, (g')^{a'_{n-u}}).$$
7. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.
8. Give h' to the pirate decoder P. Let the output of P be s_A.
9. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

For $OUT = \{i_1, i_2, \ldots, i_k\}$, it holds that
$$e'_{i_j} = \begin{cases} e_{i_j} \bmod p & \text{if } d \leftarrow \mathbf{D} \\ \text{random} & \text{if } d \leftarrow \mathbf{R}. \end{cases}$$
from Proposition A.1. Therefore, if d is chosen from $\mathbf{D}$, h' is a legal header. On the other hand, if d is chosen from $\mathbf{R}$, h' is an illegal header used in $TEST(A)$. Hence, we have
$$\begin{aligned} &|\Pr(M(d) = 1 \mid d \in \mathbf{D}) - \Pr(M_k(d) = 1 \mid d \in \mathbf{R})| \\ &= p_0 \\ &\geq \epsilon. \end{aligned}$$
from our assumption.

This means that M can distinguish $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

B Proof of Theorem 7.4

Suppose that $\Pr(TEST(A) = 1) \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder P such that $\Pr(TEST(A) = 1) \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2k})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose $a'_2, \ldots, a'_{n-u}$ at random. Let a'_1 be such that $g^{a'_1} = y$.
2. Select a random session key s' and compute h' as follows.
$$h' = (g', s'y', y^{a'_2}, y^{a'_3}, \ldots, y^{a'_{n-u}}).$$
3. Compute $g^{e'_1}, g^{e'_2}, \ldots, g^{e'_n}$ from $g^{a'_1}, g^{a'_2}, \ldots, g^{a'_{n-u}}$, where
$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$
4. Choose e_i at random for each $i \in BAD$. Let $g^{e_i} = g^{e'_i}$ for each $i \in A$.
5. From $\{g^{e_i} \mid i \in A \cup BAD\}$, compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$, where
$$(e_1, \ldots, e_n) = (a_1, \cdots, a_{n-u}) \cdot H$$
Each a_i is written as a linear combination of $\{e_i \mid i \in A \cup BAD\}$ because any $2k$ columns of H are linearly independent and $|A \cup BAD| \leq 2k$. Therefore, we can do this.
6. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$. Give h' to the pirate decoder P. Let the output of P be s_A.
7. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

Then we obtain that
$$\begin{aligned}&|\Pr(M(d) = 1 \mid d \leftarrow \mathbf{D}) - \Pr(M(d) = 1 \mid d \leftarrow \mathbf{R})|\\ &\quad = |\Pr(s_A = s' \mid d \leftarrow \mathbf{D}) - \Pr(s_A = s' \mid d \leftarrow \mathbf{R})|\end{aligned}$$

First we see that $\Pr(s' = s \mid d \leftarrow \mathbf{R})$ is negligible because y' is random. Next it is easy to see that if d is chosen from $\mathbf{D}$, then h' is a testing header used in $TEST(A)$. Therefore,
$$\Pr(s_A = s' \mid d \leftarrow \mathbf{D}) = Pr(TEST(A) = 1) \geq \epsilon$$
from our assumption.

This means that M can distinguishes $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

C Proof of Theorem 7.5

Suppose that
$$|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{\tilde{i}\}) = 1)| \geq \epsilon$$
for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^r, y, y^r)$ and $\mathbf{R} = (g, g^r, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and r is chosen at random from Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder P such that
$$|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{i\}) = 1)| \geq \epsilon$$
from a public key $pk = (g, y_1, \ldots, y_{n-u})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose e_i for each $i \in BAD \cup (A \setminus \{\tilde{i}\})$. Let $e_{\tilde{i}}$ be such that $g^{e_{\tilde{i}}} = y$.
2. Compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$ from $\{g^{e_i} \mid i \in BAD \cup A\}$, where

$$(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H.$$

3. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.
4. Next let $e'_i = e_i$ for each $i \in A \setminus \{\tilde{i}\}$ and $e'_{\tilde{i}}$ be such that $y^{e'_{\tilde{i}}} = y'$.
5. Compute $y^{a'_1}, y^{a'_2}, \ldots, y^{a'_{n-u}}$ from $\{y^{e'_i} \mid i \in BAD \cup A\}$, where

$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$

6. Select a random session key s' and compute h' as follows.

$$h' = (g', s'y^{a'_1}, y^{a'_2}, \ldots, y^{a'_{n-u}}).$$

   Give h' to the pirate decoder P. Let the output of P be s_A.
7. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

It is easy to see that if d is chosen from $\mathbf{D}$, then h' is an illegal header used in $TEST(A)$. On the other hand, if d is chosen from $\mathbf{R}$, then h' is an illegal header used in $TEST(A \setminus \{\tilde{i}\})$.

Therefore,

$$\begin{aligned}|\Pr(M(d) = 1 \mid d \leftarrow \mathbf{D}) - \Pr(M(d) = 1 \mid d \leftarrow \mathbf{R})| \\ &= \left|\Pr(TEST(A) = 1) - \Pr(TEST(A \setminus \{\tilde{i}\}) = 1)\right| \\ &\geq \epsilon\end{aligned}$$

from our assumption.

This means that M can distinguish $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

D Proof of Theorem 7.8

Suppose that $\Pr(TEST(A) = 1) \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder such that $\Pr(TEST(A) = 1) \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2k})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose e_i at random for each $i \in BAD$.
2. For each $i \in A$, choose t_i at random and compute y^{t_i}. Define e_i as $g^{e_i} = y^{t_i}$.

3. From $\{g^{e_i} \mid i \in A \cup BAD\}$, compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$, where
$$(e_1, \ldots, e_n) = (a_1, \cdots, a_{n-u}) \cdot H.$$
4. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.
5. For each $i \in A$, compute $\beta_i = (y')^{t_i}$. For each $i \in BAD$, choose a random element β_i.
6. Suppose that $y' = y^r$. Define e'_i as $\beta_i = g^{re'_i}$ for each $i \in (A \cup BAD)$.
7. From $\{\beta_i \mid i \in A \cup BAD\}$, compute $g^{ra'_1}, g^{ra'_2}, \ldots, g^{ra'_{n-u}}$, where $\beta_i = g^{re'_i}$ and
$$(e'_1, \ldots, e'_n) = (a'_1, \cdots, a'_{n-u}) \cdot H.$$
8. Select a random session key s' and compute h' as follows.
$$h' = (s'g', g^{ra'_1}, g^{ra'_2}, \ldots, g^{ra'_{n-u}}).$$
Give h' to the pirate decoder P. Let the output of P be s_A.
9. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

Then we obtain that
$$\begin{aligned}&|\Pr(M(d) = 1 \mid d \leftarrow \mathbf{D}) - \Pr(M(d) = 1 \mid d \leftarrow \mathbf{R})| \\ &\quad = |\Pr(s_A = s' \mid d \leftarrow \mathbf{D}) - \Pr(s_A = s' \mid d \leftarrow \mathbf{R})|\end{aligned}$$

First we see that $\Pr(s' = s \mid d \leftarrow \mathbf{R})$ is negligible because g' is random.

Next we will show that if d is chosen from $\mathbf{D}$, then h' is an illegal header used in $TEST(A)$. In this case, $y' = y^r$ and $g' = g^r$ for some r. We need to show that $e'_i = e_i$ for each $i \in A$. Assume that $y = g^x$. Then

1. $e_i = xt_i$ since $y^{t_i} = g^{e_i}$.
2. On the other hand,
$$g^{re'_i} = \beta_i = (y')^{t_i} = y^{rt_i} = g^{xrt_i}.$$
Therefore, $e'_i = xt_i$.

Hence, $e'_i = e_i$. Therefore, h' is an illegal header used in $TEST(A)$. Consequently,
$$\Pr(s_A = s' \mid d \leftarrow \mathbf{D}) = Pr(TEST(A) = 1) \geq \epsilon$$
from our assumption.

This means that M can distinguishes $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.