Breaking Two k-Resilient Traitor Tracing Schemes with Sublinear Ciphertext Size*

MoonShik Lee, Daegun Ma, and MinJae Seo

Department of Mathematical Sciences and ISaC-RIM,
Seoul National University, Seoul, 151-747, Korea
{kafa04,madgun7,morion81}@snu.ac.kr

Abstract. In 2004, Matsushita and Imai proposed a k-resilient public-key traitor tracing scheme which has sublinear ciphertext size $4k + 2 + (n/2k)$ with efficient black-box tracing against self-defensive pirates, where n, k are the total number of subscribers and the maximum number of colluders. After that, in 2006, they presented a hierarchical key assignment method to reduce the ciphertext size into $4k + 5 + \log(n/2k)$ by combining a complete binary tree with the former scheme.

In this paper, we show that the proposed schemes are vulnerable to our attack which makes pirate keys able to avoid the black-box tracing. Their schemes are based on multiple polynomials and our attack use a combination between different polynomials. The latter scheme can be broken by other attacks which use secret values of the key generation polynomial or use partial keys.

Keywords: cryptanalysis, public-key traitor tracing, black-box tracing, self-defensive pirates, linear attack.

1 Introduction

In modern times, enormous digital contents are transmitted over the various media through encryption and they can be decrypted only by the legitimate subscribers. But some of the subscribers may collude to make pirate decoders and distribute them. In the traitor tracing schemes, at least one of the traitors can be traced and this traceability is required in various contents delivery systems such as satellite broadcast, DMB, pay-TV, DVD, online database and so on.

The first traitor tracing scheme was introduced by Chor et al. [3] in 1994, which was inefficient and only the system manager could encrypt a message. In 1998, Kurosawa and Desmedt [5] proposed a polynomial based *public-key* tracing scheme, where any data supplier could encrypt a message. In 1999, Boneh and Franklin [1] proposed another public-key traitor tracing scheme with *black-box tracing* algorithm, where the tracer could reveal at least one of the traitors without opening the pirate decoder by using it as a black box.

* This research are partially supported by BK21 project and Korea Research Council of Fundamental Science & Technology.

M. Abdalla et al. (Eds.): ACNS 2009, LNCS 5536, pp. 238–252, 2009.

If a pirate decoder is able to detect a tracing, then it may take some reactions such as erasing the key. In 2001, Kiayias and Yung [4] considered crafty pirates and categorized them into four types, from type-0 to type-3, according to their capabilities(resettable vs. history recording, available vs. abrupt). They also proposed a generic tracing technique of hybrid colorings and schemes of type-2 and type-3 tracing.

In 2004, Matsushita and Imai [6] proposed a k-resilient public-key traitor tracing scheme with efficient black-box tracing against the type-2(resettable and abrupt) pirate decoder. In this scheme, subscribers are split into $\ell(= n/2k)$ sets and a broadcaster transmits a header H with $4k + \ell + 2$ elements, from which each subscriber in the i-th subscriber set extracts a corresponding header H_i for decrypting a session key. Although it has a limit of the collusion size k, they argued that the scheme achieved the efficient black-box tracing with sublinear ciphertext size.

In 2006, the same authors suggested another new idea of combining a tree structure with the scheme of [6] to reduce the ciphertext size [7]. They positioned each subscriber set in the leaf node of a complete binary tree and assigned several keys to a subscriber. They considered three methods of constructing the key generation polynomial, and by the last method which is called a hierarchical key assignment method, they could reduce the ciphertext size from $4k + \ell + 2$ to $4k + 5 + \log \ell$. Comparing with the fully resilient scheme [2] with $6\sqrt{n}$ header size, this scheme is more efficient when k is not so large.

Our contribution. In this paper, we break the two k-resilient schemes of [6] and [7], which have been the most efficient public-key black-box traitor-tracing schemes. An important characteristic of their schemes is that they use multiple polynomials which are interrelated. We newly introduce a variant of linear attack of combining two polynomials to construct pirate keys which cannot be traced. Furthermore, the latter scheme has another problem in adapting a tree structure, which leads to untraceable piracy through the computation of secret values or use of partial keys.

Organization. This paper is organized as follows. In Section 2, we briefly describe the schemes in [6] and [7]. In Section 3, we show our attack on the scheme of [6]. In section 4, we point out that the three methods of [7] are broken by our attack. Finally, in section 5, we conclude our paper.

2 Preliminaries

In this Section, we review the categorization of pirate decoders [4] and the schemes of [6] and [7], which are the targets of our analyses.

2.1 Models of Pirate Decoders

In 2001, Kiayias and Yung [4] categorized pirate decoders by the following two criterion.

Resettable vs. history recording : Resettable decoders can be reset to their initial state by the tracer after each test. History recording pirate decoders remember the previous inputs and may use that information to detect the tracing.

Available vs. abrupt(self-defensive) : Abrupt or self-defensive pirate decoders can take some counter-actions against the tracing if the decoder detects it. Available pirate decoder is a device that does not take such reactions.

From this, they suggested the following 4 types of pirate decoders.

type 0: Available and resettable.
type 1: Available and history recording.
type 2: Abrupt and resettable.
type 3: Abrupt and history recording.

In the schemes of [6] and [7], they considered black-box traceability against type-2 decoders, which would be simply expressed by *type-2 tracing*. Although they didn't state it explicitly, they also assumed that a tracer knew the reaction mechanism, saying that the reaction was triggered once when a tracing was detected.

2.2 Common Parameters of the Schemes

Let n, k denote the total number of subscribers and the maximum number of colluders in a coalition, respectively. Let p, q be the primes s.t. $q \mid p-1$ and $q \geq n+2k$. Let g be a qth root of unity over $\mathbb{Z}_p^*$ and G_q be the subgroup of $\mathbb{Z}_p^*$ of order q. Let $\mathcal{U}$ be a set of subscribers ($\mathcal{U} \subseteq \mathbb{Z}_q^*$). All of the participants agree on p, q and g. Split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. For notational simplicity, we assume that $|\mathcal{U}_i| = 2k$, $\mathcal{U}_i = \{u_t | 2ki+1 \leq t \leq 2k(i+1)\}$ for $0 \leq i \leq \ell-1$ and $n = 2k\ell$.

2.3 The Scheme of [6]

In 2004, Matsushita and Imai proposed an efficient type-2 black-box tracing scheme with sublinear ciphertext size[6]. We briefly summarize this scheme in different form with minor corrections. Let a *valid input* denote a header for the normal broadcast and an *invalid input* denote a header for the black-box tracing.

Key generation. Choose $a_0, \ldots, a_{2k-1}, c_0, \ldots, c_{\ell-1} \in_R \mathbb{Z}_q$ and compute the public key e by

$$e = (g, g^{a_0}, \ldots, g^{a_{2k-1}}, g^{c_0}, \ldots, g^{c_{\ell-1}}).$$

The private key for a subscriber $u \in \mathcal{U}_i$ is generated as $(u, i, f_i(u))$ where

$$f_i(u) = \sum_{j=0}^{2k-1} a_{i,j} u^j \bmod q, \quad a_{i,j} = \begin{cases} a_j & (j \neq i \bmod 2k), \\ c_i & (j = i \bmod 2k). \end{cases} \tag{1}$$

Encryption. Select a session key $s \in_R G_q$ and random numbers $R_0, R_1 \in_R \mathbb{Z}_q$. Build a header $H = (H_0, \ldots, H_{\ell-1})$ by repeating the following procedure for $0 \le i \le \ell - 1$. Set $r_i \in_R \{R_0, R_1\}$, and compute $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ where

$$\hat{h}_i = g^{r_i}, \quad h_{i,j} = \begin{cases} g^{a_j r_i} & (j \neq i \bmod 2k), \\ s g^{c_i r_i} & (j = i \bmod 2k). \end{cases} \tag{2}$$

Decryption. Suppose that $u \in \mathcal{U}_i$. The subscriber u can compute the session key s from H_i as follows,

$$\left(\frac{\prod_{j=0}^{2k-1} {h_{i,j}}^{u^j}}{{\hat{h}_i}^{f_i(u)}} \right)^{\frac{1}{u^i}} = \left(s^{u^i} \frac{\prod_{j=0}^{2k-1} g^{a_{i,j} u^j r_i}}{g^{f_i(u) r_i}} \right)^{\frac{1}{u^i}} = s.$$

Black-box tracing. For $1 \le t \le n$, repeat the following procedure.
Set $\mathcal{X} := \{u_1, \ldots, u_t\}$ as a set of revoked subscribers and $ctr_t = 0$. Find integers t_1, t_2 s.t. $t = 2kt_1 + t_2$ where $0 \le t_1 \le \ell - 1$ and $1 \le t_2 \le 2k$. Repeat the following test m times. In each test, the session key s and R_0, R_1 are chosen randomly.

1. Build the header $H = (H_0, \ldots, H_{\ell-1})$ through the following procedure.
   A: If $t_2 = 2k$, then choose $r_i \in_R \{R_0, R_1\}$ for $0 \le i \le \ell - 1$.
   A-a: For each $0 \le i \le t_1$, select a random number $z_i \in_R \mathbb{Z}_q$, compute $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ where

$$\hat{h}_i = g^{r_i}, \quad h_{i,j} = \begin{cases} g^{a_j r_i} & (j \neq i \bmod 2k), \\ g^{z_i} & (j = i \bmod 2k). \end{cases} \tag{3}$$

   A-b: For each $t_1 < i \le \ell - 1$, compute H_i in the same way as (2).

   B: If $t_2 \neq 2k$, then choose $r_i \in_R \{R_0, R_1\}$ for $0 \le i < t_1$ and set $r_{t_1} = R_1$, $r_i = R_0$ for $t_1 < i \le \ell - 1$.
   B-a: For each $0 \le i < t_1$, compute H_i in the same way as (3).
   B-b: For each $t_1 < i \le \ell - 1$, compute H_i with $r_i = R_0$ in the same way as (2).
   B-c: For $i = t_1$, let $x_1 := u_{t+1}, \cdots, x_{2k-t_2} := u_{2k(t_1+1)}$ and choose distinct random numbers $x_j \in \mathbb{Z}_q^* \setminus \mathcal{U}$ for $2k - t_2 < j \le 2k - 1$. Find $d_0, \ldots, d_{2k-1}$ s.t. $\sum_{j=0}^{2k-1} d_j x_\alpha^j = 0$ for all $1 \le \alpha \le 2k - 1$ and compute $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ where

$$\hat{h}_i = g^{R_1}, \quad h_{i,j} = \begin{cases} g^{d_j + a_j R_1} & (j \neq i \bmod 2k), \\ s g^{d_i + c_i R_1} & (j = i \bmod 2k). \end{cases} \tag{4}$$

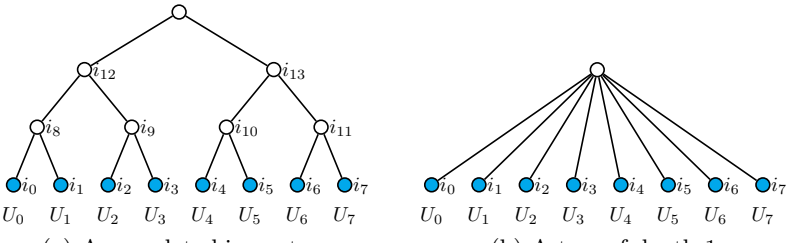
   We call this step of **B-c** the *fine revocation*.

2. Give H to the pirate decoder and observe its output.
3. If it decrypts correctly, then increment ctr_t by one. If a self-defensive reaction is triggered, then decide that the subscriber u_t is a traitor.

Finally, find an integer t s.t. $ctr_{t-1} - ctr_t$ is the maximum and then decide that the subscriber u_t is a traitor, where $ctr_0 = m$.

2.4 The Schemes of [7]

They applied the complete subtree method [8] to reduce the ciphertext size of [6]. By considering a tree T with ℓ leaves they interpreted the scheme of [6] as a depth-1 case (Fig. 1(b)) and generalized it (Fig. 1(a)).



(a) A complete binary tree (b) A tree of depth 1

Fig. 1. Structure of T ($\ell = 8$)

Let T be a complete binary tree with ℓ leaves, $\mathcal{N}_T$ be a set of the all nodes except the root and $\mathcal{L}_T (\subset \mathcal{N}_T)$ be a set of leaf nodes. For a leaf node $v \in \mathcal{L}_T$, $\mathcal{U}_v$ is a corresponding subscriber set and for a non-leaf node $v \in \mathcal{N}_T \setminus \mathcal{L}_T$, we define $\mathcal{U}_v = \bigcup \mathcal{U}_w$ where w's are the leaves of the subtree rooted at v. For a subscriber u, let $V(u)$ be a set of nodes corresponding to subscriber sets including u i.e., $V(u) = \{v \in \mathcal{N}_T | u \in \mathcal{U}_v\}$. This definition can be naturally extended to the list of subscribers such a way of $V(u, v) = V(u) \cup V(v)$. For a node v, we represent the depth by $\delta(v)$ or just δ.

For notational simplicity, we number the nodes from 0 to $2\ell - 3$ and identify αth node with the index itself, i.e. we use the notation $\mathcal{U}_\alpha := \mathcal{U}_{i_\alpha}$ where $i_\alpha \in \mathcal{N}_T$. For example, in Fig. 1(a), T has $\ell = 8$ leaves and $\mathcal{N}_T = \{0, \ldots, 13\}$, $\mathcal{L}_T = \{0, \ldots, 7\}$, $\mathcal{U}_8 = \mathcal{U}_0 \cup \mathcal{U}_1$, for $u_1 \in \mathcal{U}_0$ and $u_{8k} \in \mathcal{U}_3$, $V(u_1) = \{0, 8, 12\}$, $V(u_{8k}) = \{3, 9, 12\}$ and $V(u_1, u_{8k}) = \{0, 3, 8, 9, 12\}$.

They considered two simple extension methods and proposed one resulting method of constructing key generation polynomials. To indicate each method we use the following notation.

Method 1. The first method of the simple extension,
Method 2. The second method of the simple extension,
Method 3. The third method of their resulting suggestion.

In the following, we summarize their methods. The differences between them are originated from the way of defining the key generation polynomials. Since they stated the full scheme only for the resulting method, so do we.

Key generation

Method 1. Polynomials are generated from a single system, i.e. for $0 \leq v \leq 2\ell - 3$, $f_v(x)$ is defined by the equation (1) in the same way as [6] as well as the public key e. The private keys of a subscriber u are

represented as $K_u = \{(u, v, f_v(u))|v \in V(u)\} = \{(u, v, f_v(u)) \mid v \in \mathcal{N}_T \text{ s.t. } u \in \mathcal{U}_v\}$. For example, in Fig. 1(a), if $u \in \mathcal{U}_0$ then $K_u = \{(u, 0, f_0(u)), (u, 8, f_8(u)), (u, 12, f_{12}(u))\}$.

Method 2. Polynomials are generated from plural systems according to each level, i.e., a $f_v^{(\delta)}(x)$ is generated from the δth system where v is a node at depth δ. Let $e^{(\delta)}$ be a public key corresponding to $f_v^{(\delta)}(x)$. The private keys of a subscriber u are represented as $K_u = \{(u, v, f_v^{(\delta)}(u)) \mid v \in V(u)\}$.

Method 3. Choose $a_i, b_i, c_v, \lambda_v \in_R \mathbb{Z}_q$ for $0 \le i \le 2k-1$ and $0 \le v \le 2\ell-3$. Compute a public key e as follows,

$$e = (g, g^{a_0}, \ldots, g^{a_{2k-1}}, g^{c_0}, \ldots, g^{c_{2\ell-3}}, g^{\lambda_0}, \ldots, g^{\lambda_{2\ell-3}}).$$

Define key generation polynomials, $A_v(x), B(x)$ as follows,

$$A_v(x) = \sum_{i=0}^{2k-1} (a_{v,i} - \lambda_v b_i) x^i \bmod q, \quad B(x) = \sum_{i=0}^{2k-1} b_i x^i \bmod q, \qquad (5)$$

where $a_{v,i}$ is the same with the equation (1). The private key K_u of a subscriber u is represented as $K_u = \{(u, v, A_v(u), B(u)) \mid v \in V(u)\}$.

Encryption. Using the following Sel procedure, a broadcaster selects $\log \ell + 1$ nodes and executes Encryption in the previous Subsection for the selected nodes. For $H = (H_{v_1}, \ldots, H_{v_{\log \ell+1}})$, we denote the selected nodes by $V(H)$. For Method 2 and Method 3, it is required additional relations.

Sel. Select $\log \ell + 1$ nodes $v_1, \ldots, v_{\log \ell+1}$ including two leaves which satisfy the condition that $\bigsqcup_{i=1}^{\log \ell+1} \mathcal{U}_{v_i} = \mathcal{U}$, where $\bigsqcup$ means the disjoint union. Note that only one node is selected for each level except the leaf nodes and $|V(u) \cup V(H)| = 1$ for any subscriber u and any header H.

Method 2. $e^{(\delta)}$ is used as a public key when computing H_v where v is a node at depth δ.

Method 3. For each v, $\bar{h}_v = g^{\lambda_v r_v}$ is additionally included in H_v.

Decryption. This is described for only Method 3.

For a subscriber u and a header H, suppose that $V(u) \cap V(H) = v$. Then u computes the session key s by using $(u, v, A_v(u), B(u)) \in K_u$ from H_v as follows,

$$\left(\frac{\prod_{j=0}^{2k-1} h_{v,j}^{\ u^j}}{\hat{h}_v^{A_v(u)} \bar{h}_v^{B(u)}} \right)^{\frac{1}{u^v}} = \left(s^{u^v} \frac{\prod_{j=0}^{2k-1} g^{a_{v,j} u^j r_v}}{g^{(A_v(u) + \lambda_v B(u)) r_v}} \right)^{\frac{1}{u^v}} = s.$$

Black-box tracing. The procedure is similar to the Black-box tracing in the previous Subsection except that $\log \ell + 1$ nodes are chosen in Sel procedure and the fine revocation is executed only on a leaf node $v \in V(H) \cap \mathcal{L}_T$. We omit the detailed description.

3 A Flaw on the Scheme of [6]

In [6], authors asserted that the black-box tracing algorithm would work for any type-2 pirate decoder. In this Section we will show that this is not true by our attack which is a variant of the linear attack. In particular, in the Theorem 2 they followed the logic:

> If the subscriber u_t is not a traitor, then a pirate decoder cannot distinguish an invalid input of $\mathcal{X} = \{u_1, \ldots, u_{t-1}\}$ from an invalid input of $\mathcal{X} = \{u_1, \ldots, u_t\}$ and therefore $ctr_{t-1} - ctr_t \ll m/n$, where m is the number of tests for each $\mathcal{X}$.

However, this logic has a flaw. This means that although u_t is not a traitor, $ctr_{t-1} - ctr_t > m/n$ may happen so that the black-box tracing outputs the innocent subscriber u_t as a traitor.

In the following Subsection, we show our attack as a variant of linear attack on the proposed scheme. The linear attack was considered in [1,9] for k-resilient schemes based on a polynomial of degree k. Suppose that colluders $x_1, \ldots, x_k$ have private keys $(x_1, f(x_1)), \ldots, (x_k, f(x_k))$ for a polynomial $f(x)$ of degree k. Then a linearly combined vector

$$(\delta_0, \ldots, \delta_k, \Delta) := (\sum_{j=1}^{t} \mu_j, \sum_{j=1}^{t} \mu_j x_j, \ldots, \sum_{j=1}^{t} \mu_j x_j^k, \sum_{j=1}^{t} \mu_j f(x_j))$$

can be used as a key which is not traced, where $\mu_1, \ldots, \mu_t \in \mathbb{Z}_q$. To resist against the linear attack, the schemes based on single polynomial have raised the degree $\geq 2k-1$. However in the scheme of [6], there are multiple polynomials to be used so that the circumstance is somewhat different. As this peculiar structure affects the black-box tracing, a combination of the keys from different polynomials can be used as a key which cannot be traced.

3.1 A Variant of Linear Attack

Suppose that 2 colluders $x_1 \in \mathcal{U}_i, x_2 \in \mathcal{U}_j$ $(i < j)$ collude to make a *pirate key* K_p. Given two private keys $(x_1, f_i(x_1)), (x_2, f_j(x_2))$, they compute a pirate key

$$K_p = \{K_{i,j}\} := \{(x_1, x_2, f_i(x_1) + f_j(x_2))\}.$$

Note that

$$f_i(x_1) + f_j(x_2) = \sum_{\substack{t=0 \\ t \neq i,j}}^{2k-1} a_t(x_1^t + x_2^t) + (c_i x_1^i + a_i x_2^i) + (a_j x_1^j + c_j x_2^j).$$

In fact, each subscriber's key $(u, f_i(u))$ can be used as a vector $(u, u^2, \ldots, u^{2k-1}, f_i(u))$ in decryption phase, therefore we can also regard the pirate key $K_{i,j}$ as a vector

$$K_{i,j} = (x_1 + x_2, x_1^2 + x_2^2, \ldots, x_1^{2k-1} + x_2^{2k-1}, x_1^i, x_2^j, f_i(x_1) + f_j(x_2)).$$

Proposition 1. *For a given valid input, the pirate decoder with K_p of the above form can compute a session key with probability $\frac{1}{2}$.*

Proof. For a given valid input of $H = (H_0, \ldots, H_{\ell-1})$, from the H_i, H_j, the pirate decoder first computes[1]

$$\begin{aligned}\gamma &:= h_{i,i}^{x_1^i}\, h_{j,i}^{x_2^i}\, h_{i,j}^{x_1^j}\, h_{j,j}^{x_2^j} \\ &= (sg^{c_i r_i})^{x_1^i} (g^{a_i r_j})^{x_2^i} (g^{a_j r_i})^{x_1^j} (sg^{c_j r_j})^{x_2^j} \\ &= s^{x_1^i + x_2^j} \cdot g^{c_i x_1^i r_i + a_i x_2^i r_j + a_j x_1^j r_i + c_j x_2^j r_j}.\end{aligned} \tag{6}$$

If $r_i = r_j$, then it can compute the session key by

$$\begin{aligned}&\left(\gamma \prod_{\substack{t=0 \\ t\neq i,j}}^{2k-1} h_{i,t}^{x_1^t + x_2^t} \Big/ \hat{h}_i^{f_i(x_1)+f_j(x_2)}\right)^{\frac{1}{x_1^i + x_2^j}} \\ &= \left(\gamma \prod_{\substack{t=0 \\ t\neq i,j}}^{2k-1} (g^{a_t r_i})^{x_1^t + x_2^t} \Big/ g^{(f_i(x_1)+f_j(x_2))r_i}\right)^{\frac{1}{x_1^i + x_2^j}} = s.\end{aligned} \tag{7}$$

Since each r_i is chosen at random from $\{R_0, R_1\}$ for valid inputs, the probability that $r_i = r_j$ is $\frac{1}{2}$ for any $i \neq j$. □

However, the pirate decoder with the decryption probability $\frac{1}{2}$ may not be used in many applications. For these cases we can also consider a pirate decoder with the probability 1 which can be made by 3 colluders. Suppose that 3 colluders $x_1 \in \mathcal{U}_i, x_2 \in \mathcal{U}_j, x_3 \in \mathcal{U}_k (i < j < k)$ with a pirate key $K_p = \{K_{i,j}, K_{i,k}, K_{j,k}\}$. Since a broadcaster selects r_i, r_j, r_k randomly from $\{R_0, R_1\}$ to build a valid input, at least two values of them coincide, therefore the pirate decoder with this key can compute a session key with probability 1 according to the Proposition 1.

3.2 Probability of Untraceability

We now consider the probability that the pirate decoder with a pirate key $K_{i,j}$ $(i < j)$ can decrypt the invalid inputs for the given revocation set $\mathcal{X}$. Note that in the black-box tracing algorithm, the tracer uses this probability to decide whether a traitor is included in the $\mathcal{X}$ or not. Since the black-box algorithm requires the gradual increment of the revoked subscriber set $\mathcal{X}$, we assume that $\mathcal{X} = \{u_1, \ldots, u_t\}$[2], where $t = 2kt_1 + t_2$ for $0 \leq t_1 \leq \ell - 1$ and $1 \leq t_2 \leq 2k$.

Let Pr_t denote the probability that the pirate decoder computes a session key correctly from the invalid inputs H for $\mathcal{X} = \{u_1, \ldots, u_t\}$. We consider a pirate key $K_{i,j}$ made by two colluders $x_1 \in \mathcal{U}_i$, $x_2 \in \mathcal{U}_j$ $(i < j)$ at first. Note that the

[1] For easy reading, we drop the 'mod $2k$' in the second subscript of h.

[2] In the black-box tracing phase, a tracer can determine the order of subscriber set to be revoked as well as the order of subscriber to be revoked in a given subscriber set.

[**A-b**] and the [**B-b**] steps are the same as the valid inputs. Then, according to the case in the tracing algorithm, the probability Pr_t for each $1 \leq t \leq n$ becomes:

1. If $\mathcal{X} \cap \mathcal{U}_i = \varnothing$ and $t_2 \neq 2k$, then $Pr_t = 1$,
   since H_i and H_j are computed according to [**B-b**] and $r_i = r_j = R_0$.

2. If $\mathcal{X} \cap \mathcal{U}_i = \varnothing$ and $t_2 = 2k$, then $Pr_t = \frac{1}{2}$,
   since H_i and H_j are computed according to [**A-b**] and r_i, r_j are chosen from $\{R_0, R_1\}$ randomly.

3. If $\varnothing \neq \mathcal{X} \cap \mathcal{U}_i \neq \mathcal{U}_i$ and $\mathcal{X} \cap \mathcal{U}_j = \varnothing$ then $Pr_t = 0$,
   since $H_i = (g^{R_1}, g^{d_0+a_0R_1}, \ldots, sg^{d_i+c_iR_1}, \ldots, g^{d_{2k-1}+a_{2k-1}R_1})$ is computed according to [**B-c**] and H_j is computed according to [**B-b**] with $r_j = R_0$, it becomes $r_i \neq r_j$.

4. If $\mathcal{X} \supset \mathcal{U}_i$ then $Pr_t = 0$,
   since H_i is computed according to [**A-a**] or [**B-a**] and $h_{i,i} = g^{z_i}$, so the pirate decoder cannot decrypt the session key.

Observe that the Pr_t does not decrease so that their black-box algorithm does not work. However, since the probability vanishes from $t = 2ki+1$, a tracer can know that the first traitor x_1 is positioned in $\mathcal{U}_i$, but cannot know who the traitor is exactly.

Theorem 1. *For the pirate decoder with a pirate key $K_{i,j}$ by two colluders $x_1 \in \mathcal{U}_i$ and $x_2 \in \mathcal{U}_j$ $(i < j)$, they cannot be traced with the probability $1 - \frac{1}{2k}$.*

Proof. It is clear from the above argument. Since there are $2k$ subscribers in $\mathcal{U}_i$, the probability that x_1 can be pointed out as a traitor is $\frac{1}{2k}$. □

We now consider a pirate decoder from 3 colluders, $x_1 \in \mathcal{U}_i, x_2 \in \mathcal{U}_j, x_3 \in \mathcal{U}_k$ where $i < j < k$ with 3 combined keys $K_p = \{K_{i,j}, K_{i,k}, K_{j,k}\}$. Then since $r_i, r_j, r_k \in_R \{R_0, R_1\}$, there must be at least one pair that the random numbers coincide. So, this pirate decoder can always decrypt the valid inputs correctly. The probability that this pirate decoder can decrypt the invalid inputs correctly will be:

1. If $\mathcal{X} \cap \mathcal{U}_i = \varnothing$, then $Pr_t = 1$, since
   (a) If $t_2 \neq 2k$ then H_i, H_j, H_k are computed according to [**B-b**] and $r_i = r_j = r_k = R_0$, so it decrypts all such H correctly.
   (b) If $t_2 = 2k$ then according to [**A-b**] there is one pair that the random numbers coincide, so it decrypts all such H correctly.

2. If $\mathcal{X} \cap \mathcal{U}_i \neq \varnothing$ and $\mathcal{X} \cap \mathcal{U}_j = \varnothing$, then
   (a) If $t_2 \neq 2k$ then $Pr_t = 1$, since H_j, H_k are computed according to [**B-b**] and $r_j = r_k = R_0$.

(b) If $t_2 = 2k$ then $Pr_t = \frac{1}{2}$, since $h_{i,i} = g^{z_i}$ according to **[A-a]**, and H_j, H_k are computed according to **[A-b]** and the probability that $r_j = r_k$ is $\frac{1}{2}$.

3. If $\varnothing \neq \mathcal{X} \cap \mathcal{U}_j \neq \mathcal{U}_j$ and $\mathcal{X} \cap \mathcal{U}_k = \varnothing$, then $Pr_t = 0$,
since according to **[B-a]**, **[B-b]** and **[B-c]**, $h_{i,i} = g^{z_i}$ in H_i and H_j, H_k are computed by $r_j = R_1$ and $r_k = R_0$ respectively.

4. If $\mathcal{X} \supset \mathcal{U}_j$, then $Pr_t = 0$,
since according to **[B-a]**, $h_{i,i} = g^{z_i}$ in H_i and $h_{j,j} = g^{z_j}$ in H_j.

Observe that a tracer can know in which subscriber set $\mathcal{U}_j$ the second traitor x_2 is positioned, but cannot know who the traitor is exactly. From this we can obtain the similar result that this pirate decoder from 3 colluders cannot be traced with the probability $1 - \frac{1}{2k}$. We skip the explanations on the cases that more than 3 colluders attend the piracy.

4 Flaws on the Scheme of [7]

In [7], to decrease the ciphertext size, the authors considered a complete binary tree and bound each subscriber set $\mathcal{U}_i$ to a leaf node for $0 \leq i \leq \ell - 1$. Then they considered three methods of defining key generation polynomials for each node. They argued that the first simple extension method was insecure against collusion attack through solving linear equations[3], the second simple extension method was just inefficient from the viewpoint of the header size and the last hierarchical key assignment method was efficient and secure. However, all of their arguments were flawed.

In this Section, we will show what flaws they have; (1) they are also vulnerable to our attack as shown in Section 3, (2) there is an easy way of extracting secret values which can be used for decryption and (3) there is a structural flaw of easy construction of a non-traceable pirate key.

4.1 A Variant of Linear Attack on the Hierarchical Key Assignment Methods

The first and third methods are also vulnerable to our attack as shown in Section 3. Now consider a case of 2 colluders x_1 and x_2. Suppose that x_1 and x_2 are positioned at the left child and right child node of the root, respectively, or equivalently $V(x_1) \cap V(x_2) = \varnothing$. For the case of $V(x_1) \cap V(x_2) \neq \varnothing$, we will introduce different types of attacks in the following Subsections. To avoid unnecessary repetitions, we describe our attack just on the third method.

[3] But this argument is incorrect. They missed the fact that there had to be sufficiently many *linearly independent* equations, not just equations. By considering the rank of the corresponding matrix, this can be shown easily.

Each subscriber has $\log \ell$ keys, which are on the path between a leaf node and the child node of the root. The colluders x_1 and x_2 have their keys,

$$\{(x_1, i, A_i(x_1), B(x_1))|i \in V(x_1)\},$$
$$\{(x_2, j, A_j(x_2), B(x_2))|j \in V(x_2)\}.$$

To apply our attack, we assume that the pirate decoder computes all the possible combinations of $K_{i,j}$ into the pirate key as follows.

$$K_p = \{K_{i,j}|i \in V(x_1), j \in V(x_2)\}.$$

However, only some parts are necessary. For example, in Fig. 1(a), if $x_1 \in \mathcal{U}_2$ and $x_2 \in \mathcal{U}_5$ then they need only 4 keys of $\{K_{2,13}, K_{9,13}, K_{12,5}, K_{12,10}\}$, since the other combinations of i, j cannot be included in $V(H)$ at once.

Proposition 2. *For a given valid input, the pirate decoder with K_p of the above form can compute a session key with probability $\frac{1}{2}$.*

Proof. This is similar to the Proposition 1. For a given header H, let $V(H) \cap V(x_1) = i$ and $V(H) \cap V(x_2) = j$. Since $V(x_1) \cap V(x_2) = \varnothing$ implies $i \neq j$, the combined key for this case, $K_{i,j}$ can be obtained as follows,

$$K_{i,j} := (x_1+x_2, x_1^2+x_2^2, \ldots, x_1^{2k-1}+x_2^{2k-1}, x_1^i, x_2^j, A_i(x_1)+A_j(x_2), B(x_1), B(x_2)).$$

Note that

$$\begin{aligned} A_i(x_1) + A_j(x_2) = &\sum_{\substack{t=0 \\ t\neq i,j}}^{2k-1} a_t(x_1+x_2)^t + (c_i x_1^i + a_i x_2^i) + (a_j x_1^j + c_j x_2^j) \\ &- \lambda_i B(x_1) - \lambda_j B(x_2). \end{aligned} \tag{8}$$

The pirate decoder computes

$$\begin{aligned} \gamma = &\ h_{i,i}^{x_1^i} h_{j,i}^{x_2^i} h_{i,j}^{x_1^j} h_{j,j}^{x_2^j} \\ = &\ s^{x_1^i + x_2^j} \cdot g^{c_i x_1^i r_i + a_i x_2^i r_j + a_j x_1^j r_i + c_j x_2^j r_j}. \end{aligned}$$

If $r_i = r_j$, then it can compute the session key by

$$\begin{aligned} &\left(\gamma \prod_{\substack{t=0 \\ t\neq i,j}}^{2k-1} h_{i,t}^{x_1^t + x_2^t} \Big/ \hat{h}_i^{A_i(x_1)+A_j(x_2)} \bar{h}_i^{B(x_1)} \bar{h}_j^{B(x_2)} \right)^{\frac{1}{x_1^i + x_2^j}} \\ &= \left(\gamma \prod_{\substack{t=0 \\ t\neq i,j}}^{2k-1} (g^{a_t r_i})^{x_1^t + x_2^t} \Big/ g^{(A_i(x_1)+A_j(x_2)+\lambda_i B(x_1)+\lambda_j B(x_2)) r_i} \right)^{\frac{1}{x_1^i + x_2^j}} = s. \end{aligned} \tag{9}$$

For valid inputs, since each r_i is chosen from $\{R_0, R_1\}$ at random, the probability that $r_i = r_j$ is $\frac{1}{2}$ for any $i \neq j$. □

Similarly to the Theorem 1, this pirate key cannot be traced with high probability. Note that we still consider two colluders x_1 and x_2 who are in the left and right child of the root, respectively.

Theorem 2. *For the pirate decoder with a pirate key K_p of the above form by two colluders, they cannot be traced with the probability $1 - \frac{1}{2k}$.*

Proof. It is similar to the proof of Theorem 1, but since the key which is used for decryption varies for each header H, the precise description is somewhat complicated. Let $V(x_1) \cap \mathcal{L}_T = i$ and $V(x_2) \cap \mathcal{L}_T = j$, then $i < j$. The probability Pr_t for each $1 \leq t \leq n$ becomes: Let $v := V(H) \cap V(x_1)$ and $w := V(H) \cap V(x_2)$.

1. If $\mathcal{X} \cap \mathcal{U}_i = \varnothing$ and $t_2 \neq 2k$, then $Pr_t = 1$.
   Given a header H for each $\mathcal{X}$, since $r_v = r_w = R_0$, it can decrypt the H using $K_{v,w}$.
2. If $\mathcal{X} \cap \mathcal{U}_i = \varnothing$ and $t_2 = 2k$, then $Pr_t = \frac{1}{2}$.
   For the nodes v, w, since r_v and r_w are randomly chosen from $\{R_0, R_1\}$, the probability of $r_v = r_w$ is $\frac{1}{2}$, thus $Pr_t = \frac{1}{2}$.
3. If $\varnothing \neq \mathcal{X} \cap \mathcal{U}_i \neq \mathcal{U}_i$ and $\mathcal{X} \cap \mathcal{U}_j = \varnothing$ then $Pr_t = 0$.
   Since $\mathcal{U}_i$ where i is the leaf node is being finely revoked, $r_i = R_1$ and $r_w = R_0$, thus $Pr_t = 0$.
4. If $\mathcal{X} \supset \mathcal{U}_i$ then $Pr_t = 0$.
   Since $h_{v,v} = g^{z_v}$, $Pr_t = 0$.

Note that their black-box tracing algorithm also doesn't work for this pirate decoder. But by checking the probability of zero, a tracer can know the $\mathcal{U}_i$ which includes x_1, but cannot know who a traitor is exactly. From this we can conclude that the colluders of the pirate decoder cannot be traced with the probability $1 - \frac{1}{2k}$. □

Similar to the case of [6], we can still raise the probability of decryption on the valid inputs to 1 by 3 colluders. But this collusion are also related to the vulnerabilities which are argued in the next Subsections and from them the decryption probability can be 1 only by 2 colluders, we don't need to describe such collusion attacks.

4.2 Extraction of Secret Values

In this Subsection, we consider another attack which is applicable to the first and third methods by two and three colluders, respectively. The vulnerabilities also come from the structural problem that a subscriber has several keys from similar polynomials.

More precisely, a subscriber receives $\log \ell$ keys, but each couple of the key generation polynomials have the same coefficient except for two or three terms. By subtracting the polynomials with each other, 2 or 3 colluders can compute some secret values which should be kept securely. Since the procedure of extracting the secret values is similar in the first and third methods, we describe the procedure only on the third method.

Consider that three colluders x_1, x_2 and x_3 are in the same subscriber set corresponding to a leaf node. They are commonly included in the $\log \ell$ sets. For each $\mathcal{U}_i$ and $\mathcal{U}_j$ s.t. $i, j \in V(x_1) = V(x_2) = V(x_3)$ and $i < j$, they can set up the following system of equations.

$$\begin{cases} A_i(x_1) - A_j(x_1) = (c_i - a_i)x_1^i + (a_j - c_j)x_1^j - (\lambda_i - \lambda_j)B(x_1) \\ A_i(x_2) - A_j(x_2) = (c_i - a_i)x_2^i + (a_j - c_j)x_2^j - (\lambda_i - \lambda_j)B(x_2) \\ A_i(x_3) - A_j(x_3) = (c_i - a_i)x_3^i + (a_j - c_j)x_3^j - (\lambda_i - \lambda_j)B(x_3) \end{cases} \tag{10}$$

By solving this system of equations, they can compute $c_i - a_i$ for all i in $V(x_1)$. For a valid input H and the node i s.t. $i = V(H) \cap V(x_1)$, if there is another node $t \in V(H)$ s.t. $r_i = r_t$, they can obtain the session key by computing

$$\frac{h_{i,i}/h_{t,i}}{\hat{h}_i^{c_i - a_i}} = \frac{s \cdot g^{c_i r_i}/g^{a_i r_t}}{(g^{r_i})^{c_i - a_i}} = s, \tag{11}$$

where $h_{i,i}, \hat{h}_i$ are from H_i and $h_{t,i}$ is from H_t.

Note that the probability that there is such an additional node $t \in V(H)$ is $1 - \frac{2}{2^{\log \ell + 1}} = 1 - \frac{1}{\ell}$. This is because that a broadcaster selects $\log \ell + 1$ nodes and each r_v for $v \in V(H)$ is chosen in $\{R_0, R_1\}$ at random. So, the pirate decoder with the following secret values

$$K_p = \{c_i - a_i | i \in V(x_1)\}$$

can compute a session key correctly with high probability.

For example, in Fig. 1(a), let $x_1, x_2, x_3 \in \mathcal{U}_0$ be colluders, they also belong to $\mathcal{U}_8$,$\mathcal{U}_{12}$, then $\{0, 8, 12 \in V(x_1) = V(x_2) = V(x_3)\}$, they can solve the above system of equations and obtain $c_i - a_i$ for $i \in \{0, 8, 12\}$.

In this attack, since these values do not contain any information of the traitors, the tracer cannot find the identities of the traitors.

4.3 Constructing a Non-traceable Key Using Partial Keys

Although a tree-based tracing scheme has many good properties, the approach of combining the scheme of [6] with a tree structure has a critical vulnerability. This defect is related to the requirement of the Sel procedure of Encryption phase, so that it is applicable to all methods.

In the black-box tracing algorithm, in order to identify a traitor, one leaf node should be selected for the fine revocation. But this requirement has a flaw that if the colluders make a pirate key only with their keys corresponding to non-leaf nodes, then they can evade from the tracing algorithm. Furthermore, to achieve the minimum ciphertext size, the scheme requires a special form of node selection in the Sel procedure. If we follow the original Sel procedure, a tracer must take two subscriber sets corresponding to sibling leaf nodes of size $2k$, one set of size of $4k$, ..., one set of size $\frac{n}{2}$ corresponding to a child of the root node.

Note that the nodes of $i_{2\ell-4}$ and $i_{2\ell-3}$ are left and right child of the root node. Consider two colluders $x_1 \in \mathcal{U}_{2\ell-4}$ and $x_2 \in \mathcal{U}_{2\ell-3}$, i.e. they are in the left and

right side of the tree T, respectively. They construct a pirate key K_p by collecting their keys corresponding to the nodes of depth 1. Then, for a valid input H, since there should be one node of depth 1 in the $V(H)$, the pirate decoder can always decrypt it using one of two keys. Since there are $\frac{n}{2}$ subscribers in each $\mathcal{U}_{2\ell-4}$ or $\mathcal{U}_{2\ell-3}$, they can be (almost) perfectly untraceable.

This simple but powerful attack is possible since the Sel always select a node of depth 1. One way of evading from this attack is to increase the number of selected nodes in the Sel procedure at the cost of efficiency. But this trial also fails by the following observations.

Since a subscriber should be able to decrypt valid inputs all the time and the height of the tree T should not be 1, we can find essential requirements of the Sel procedure as follows.

$$\bigcup_{v \in V(H)} \mathcal{U}_v = \mathcal{U}, \quad V(H) \setminus \mathcal{L}_T \neq \varnothing. \tag{12}$$

We consider two colluders x_1 and x_2 who are not in the sibling leaf nodes. They construct a pirate key K_p by collecting all their keys corresponding to non-leaf nodes. For example, in Fig. 1(a), suppose that two colluders $x_1 \in \mathcal{U}_1, x_2 \in \mathcal{U}_2$ for the third method, then a pirate key can be

$$K_p = \{(x_1, 8, A_8(x_1), B(x_1)), (x_1, 12, A_{12}(x_1), B(x_1)), (x_2, 9, A_9(x_2), B(x_2))\}.$$

Proposition 3. *For any valid input H through the Sel procedure satisfying the conditions (12), the pirate decoder with K_p of the above form can always compute a session key.*

Proof. Note that if the decoder has a key corresponding to the node v included in $V(H)$ then it can decrypt the H. For a subscriber u, the first condition of (12) can be rewritten as $V(H) \cap V(u) \neq \varnothing$. If we denote the nodes corresponding to the K_p by $V(K_p)$, then it is $V(K_p) = (V(x_1) \cup V(x_2)) \setminus \mathcal{L}_T$. Therefore,

$$\begin{aligned} V(H) \cap V(K_p) = &\quad ((V(H) \cap V(x_1)) \cup (V(H) \cap V(x_1))) \setminus \mathcal{L}_T \\ &= (V(H) \cap V(x_1) \setminus \mathcal{L}_T) \cup (V(H) \cap V(x_2) \setminus \mathcal{L}_T)\,. \end{aligned}$$

This becomes empty only when $V(H)$ intersects at leaves with $V(x_1)$ and $V(x_2)$ at once. But since the two leaf nodes corresponding x_1 and x_2 are not siblings and the leaf nodes of $V(H)$ should be siblings, this set cannot be empty. From this, it is straightforward to decrypt a valid input all the time with the K_p. □

Since the keys corresponding to leaves are removed from the K_p, the untraceable probability is at least $1 - \frac{1}{4k}$. From the point of view of the pirate decoder, it is better to construct the pirate key K_p using the keys corresponding to the nodes of small depths. The above proposition shows that for all the Sel procedures which satisfy the conditions of (12), it is possible to construct a pirate decoder which can decrypt valid inputs with probability 1 but cannot be traced with the probability at least $1 - \frac{1}{4k}$.

5 Conclusion

In 2004, Matsushita and Imai proposed an efficient k-resilient public-key black-box traitor tracing scheme against self-defensive pirates, and in 2006, the same authors proposed a hierarchical key assignment method to reduce the ciphertext size by combining a complete binary tree with the former scheme. In this paper, we showed that the proposed schemes were vulnerable to our attack and it was possible to make an untraceable pirate decoder. Their schemes were based on multiple polynomials and our attack used a combination between different polynomials. The latter scheme also had a structural problem that we could exploit to deduce different types of attacks.

Although this paper is concentrated on breaking these schemes, we would like to remark that their contributions are still significant. In their schemes, there are many features to be discussed more, which were not explained in detail even in the original papers. In fact, we started this research from studying the work of [6]. After all, we happened to know that there were many delicate problems and they resolved some of them by novel ideas. Including modifying their schemes to resist our attacks, there still remain some problems to be resolved. We believe that much understanding of their schemes will lead us to some significant advance in the public-key black-box traitor tracing area.

References

1. Boneh, D., Franklin, M.: An Efficient Public Key Traitor Tracing Scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999)
2. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
3. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)
4. Kiayias, A., Yung, M.: On Crafty Pirates and Foxy Tracers. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 450–465. Springer, Heidelberg (2002)
5. Kurosawa, K., Desmedt, Y.: Optimum Traitor Tracing and Asymmetric Scheme. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 145–157. Springer, Heidelberg (1998)
6. Matsushita, T., Imai, H.: A Public-Key Black-Box Traitor Tracing Scheme with Sublinear Ciphertext Size against Self-Defensive Pirates. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 260–275. Springer, Heidelberg (2004)
7. Matsushita, T., Imai, H.: Hierarchical Key Assignment for Black-Box Tracing with Efficient Ciphertext Size. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 92–111. Springer, Heidelberg (2006)
8. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
9. Stinson, D., Wei, R.: Key Preassigned Traceability Schemes for Broadcast Encryption. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 144–156. Springer, Heidelberg (1999)