A Public-Key Black-Box Traitor Tracing Scheme with Sublinear Ciphertext Size against Self-Defensive Pirates

Tatsuyuki MATSUSHITA[1,2] and Hideki IMAI[2]

[1] TOSHIBA Corporate Research & Development Center
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
tatsuyuki.matsushita@toshiba.co.jp
[2] Institute of Industrial Science, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, Japan
imai@iis.u-tokyo.ac.jp

Abstract. We propose a public-key traitor tracing scheme in which (1) the size of a ciphertext is sublinear in the number of receivers and (2) black-box tracing is efficiently achieved against self-defensive pirate decoders. When assuming that a pirate decoder can take some self-defensive reaction (e.g., erasing all of the internal keys and shutting down) to escape from tracing if it detects tracing, it has been an open question to construct a sublinear black-box traitor tracing scheme that can detect efficiently at least one traitor (who builds the pirate decoder) with overwhelming probability, although a tracing algorithm that works successfully against self-defensive pirate decoders itself is known. In this paper, we answer affirmatively the above question by presenting a concrete construction of a public-key black-box tracing scheme in which the known tracing algorithm can be used while keeping the size of a ciphertext sublinear.
Key words: Public-key traitor tracing, Black-box tracing, Self-defensive pirates

1 Introduction

Consider content distribution (e.g., pay-TV) in which digital contents should be available only to subscribers. A data supplier broadcasts an encrypted version of the digital contents (e.g., a movie) to subscribers, and only subscribers can decrypt them with their decryption keys given in advance. In this application, malicious subscribers may redistribute their decryption keys to non-subscribers. This piracy is serious since it allows the non-subscribers to have illegal access to the contents.

To prevent the piracy, *traitor tracing* [3] has been studied extensively. In traitor tracing, each subscriber is given a distinct decryption key (*personal key*) which is contained in a decryption device (*decoder*), and the data supplier broadcasts both the contents encrypted with a *session key* and the encrypted session key (*header*). The subscribers can obtain the session key (and consequently the contents) by inputting the received header to their decoders. In this scenario, malicious subscribers (*traitors*) may give away their personal keys to a pirated version of a decoder (*pirate decoder*). Once the pirate decoder is found, at least one of the traitors who join the piracy can be identified from it. A traitor tracing scheme discourages traitors from committing the piracy since the confiscated pirate decoder can be traced back to its producers.

Among traitor tracing schemes, our interest is in a black-box tracing scheme in the public-key setting. In black-box tracing, a tracer does not break open the pirate decoder but uses it as a black box. Briefly, the tracer chooses a set of suspects and tests whether traitors are included in it only by observing the behavior of the pirate decoder on chosen inputs. Since traitors can be identified no matter how the pirate decoder is implemented, it is desirable to support black-box tracing. In the public-key setting, there are one or more public keys and subscribers can decrypt the header by using their personal keys. Since no secret information is needed to build the header and to execute the tracing algorithm, anyone can work as a data supplier and/or a tracer. This property is desirable as well because of the following two reasons: (1) it enhances the sender-scalability in the sense that plural data suppliers can use the same system and (2) it provides public verifiability of the tracing result, which is a stronger deterrent to the piracy.

As a public-key black-box tracing scheme, the schemes of [7, 2] are known.[1] While these are efficient in the sense that the size of a personal key is constant and that of a header is linear only in the maximum number of traitors in a coalition, the running time of the tracing algorithm is exponential in the maximum coalition size, hence impractical. The convergence time for the tracing algorithm is improved to be practical in the schemes of [5, 9] by integrating the mechanism of revocation of any number of subscribers into black-box tracing. However, if it is assumed that a pirate decoder can take measures (e.g., it erases all of the internal keys and shuts down once it detects tracing) that might escape from tracing, tracing is impossible since the identities of suspects are revealed in the inputs for black-box tracing. In this paper, we consider this type of pirate decoders.

1.1 Our result

We explain our contribution by comparing previous schemes against self-defensive pirate decoders with ours. (See Table 1.) As mentioned above, in the scheme of [2, 8] the tracer can only do *black-box confirmation* in which the number of suspects examined in one test has to be limited to k, where k is the maximum coalition size. Therefore, the black-box confirmation algorithm needs to be executed on all of the possible $\binom{n}{k}$ sets of suspects in the worst case, where n is the total number of subscribers. This results in an impractical tracing algorithm. Note that there is a trade-off between the running time of the tracing algorithm and the transmission overhead. For instance, if we set $k = n-1$, then the number of sets of suspects required for tracing is reduced to n, but the size of a header

[1] The scheme of [7] is improved in [10, 8].

Table 1. Summary of our result (n: the total number of subscribers, k: the maximum coalition size, c: a constant ($0 < c < 1$))

	Personal-key size	Header size	# of sets of suspects required for tracing	Type of tracing
[2, 8]	$O(1)$	$O(k)$	$\binom{n}{k}$	Black-box confirmation
[6] ($c = 1/2$)	$O(1)$	$O(\sqrt{n})$	$\sqrt{n}$	Black-box list-tracing
Ours ($k = \sqrt{n/8}$)	$O(1)$	$O(\sqrt{n})$	n	Black-box tracing

is linear in n, hence inefficient. It has been an open question to obtain a traitor tracing scheme with both practical black-box tracing and sublinear header size, as pointed out in [2].

In [6], a partial solution to this question is presented by introducing a relaxation idea called as *list-tracing* in which the output of the tracing algorithm is a set of suspects, i.e., a suspect list, and it is guaranteed that at least one traitor is included in it. The scheme of [6] is based on that of [4] and achieves both practical black-box list-tracing and sublinear header size. Unfortunately, this approach incurs another trade-off between the size of the suspect list and that of a header. In order to reduce the header size the suspect list needs to be larger, but the probability that the tracer detects a traitor correctly is in inverse proportion to the size of the suspect list, if the tracer attempts to identify the traitor only from the suspect list.

In this paper, we solve the open question without the list-tracing approach. By applying the key-generation method of [9] to the scheme of [8], a sublinear public-key black-box tracing scheme against self-defensive pirate decoders can be obtained. Note that the improvement we achieve is not in the tracing algorithm itself but in a concrete construction of a public-key black-box tracing scheme in which the known tracing algorithm that can identify at least one traitor with overwhelming probability from the self-defensive pirate decoder can be used while keeping the size of a header sublinear.

The rest of the paper is organized as follows. In Sect. 2, the assumptions on the pirate decoder are described. We propose a sublinear public-key black-box tracing scheme in Sect. 3. The proposed scheme is analyzed in terms of security and efficiency in Sect. 4 and Sect. 5, respectively. We present our conclusions in Sect. 6.

2 Assumptions on pirate decoders

Let a *valid input* denote a header for the normal broadcast and an *invalid input* denote a header for black-box tracing. In this paper, we make two assumptions on the pirate decoder.

Assumption 1 The pirate decoder can take measures that might escape from tracing if it detects tracing.

In Assumption 1 the pirate decoder outputs the correct plaintext only when it gets a valid input or an invalid input which is indistinguishable from a valid one. If the pirate decoder detects that it is examined in the tracing process, it will evade tracing by, e.g., erasing all of the internal keys and shutting down. As well as such self-defensive reaction, the pirate decoder can take aggressive countermeasures (e.g., crashing the host system or releasing a virus) as described in [6]. Note that (1) for simplicity we assume that the reaction is triggered deterministically, i.e., it is activated once the pirate decoder detects tracing and (2) our scheme can be easily extended to the general probabilistic case. In order to identify efficiently traitors from the pirate decoder with the reaction mechanism, it is necessary that a tracing algorithm can decide at least one traitor immediately when the reaction is triggered.

Assumption 2 The tracer can reset the pirate decoder to its initial state each time the tracer gives the input to it.

Assumption 2 means that each test during black-box tracing can be done independently. We do not consider the pirate decoder that records the previous inputs submitted by the tracer and reacts based on its record.

The pirate decoder assumed in the paper can be viewed as a *type-2* pirate decoder categorized in [6].

3 Proposed Scheme

First, we describe an outline of the proposed scheme. Secondly, an explicit construction of our scheme is shown.

3.1 Outline

Our scheme consists of the four phases.
Key generation: A trusted party generates and secretly gives every subscriber a distinct personal key. The personal key is stored in the decoder.
Encryption: The data supplier encrypts (1) the contents with the session key and (2) the session key itself as a header. Then, the data supplier broadcasts the encrypted contents and the header. To avoid complication, we assume that (1) a symmetric encryption algorithm used for encryption of the contents is secure and publicly known and (2) a broadcast channel is reliable in the sense that the received information is not altered.
Decryption: When receiving the header, subscribers compute the session key (and consequently the contents) by inputting it to their decoders.
Black-box tracing: Suppose that the pirate decoder is confiscated. In the jth test, the tracer chooses a subscriber, u_j, and builds the header in which the subscribers, $u_1, \ldots, u_j$, are revoked and the others are not, where $u_1, \ldots, u_{j-1}$ has been selected in the $(j-1)$th test. The tracer inputs this header to the pirate decoder and observes whether it decrypts correctly or not. If its output is (1)

correct on the input where a set of revoked subscribers is $\mathcal{X}$ and (2) incorrect on the input where a set of revoked ones is $\mathcal{X} \cup \{u\}$, then the tracer decides that the subscriber, u, is a traitor.

3.2 Protocol

Let n be the total number of subscribers and k be the maximum number of traitors in a coalition. Let p, q be primes s.t. $q|p-1$ and $q \geq n + 2k - 1$. Let g be a qth root of unity over $\mathbb{Z}_p^*$ and G_q be a subgroup of $\mathbb{Z}_p^*$ of order q. Let $\mathcal{U}$ be a set of subscribers ($\mathcal{U} \subseteq \mathbb{Z}_q \backslash \{0\}$). All of the participants agree on p, q, and g. The calculations are done over $\mathbb{Z}_p^*$ unless otherwise specified.

Key generation: The key-generation method is similar to that of [9]. Split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. These subsets are publicly known. Choose $a_0, \ldots, a_{2k-1}, b_0, \ldots, b_{\ell-1} \in_{\mathrm{R}} \mathbb{Z}_q$. Then, compute the public key e as follows:

$$\begin{aligned} e &= (g, y_{0,0}, \ldots, y_{0,2k-1}, y_{1,0}, \ldots, y_{1,\ell-1}) \\ &= (g, g^{a_0}, \ldots, g^{a_{2k-1}}, g^{b_0}, \ldots, g^{b_{\ell-1}}). \end{aligned}$$

Suppose that $u \in \mathcal{U}_i$. The subscriber u's personal key is $(u, i, f_i(u))$ where

$$f_i(u) = \sum_{j=0}^{2k-1} a_{i,j} u^j \bmod q,$$

$$a_{i,j} = \begin{cases} a_j & (j \neq i \bmod 2k), \\ b_i & (j = i \bmod 2k). \end{cases}$$

Encryption: Select the session key $s \in_{\mathrm{R}} G_q$ and random numbers $R_0, R_1 \in_{\mathrm{R}} \mathbb{Z}_q$. Build the header $H = (H_0, \ldots, H_{\ell-1})$ by repeating the following procedure for $0 \leq i \leq \ell - 1$.

- Set $r_i = R_0$ or R_1, and compute H_i as follows.

$$H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1}),$$

$$\hat{h}_i = g^{r_i},$$

$$h_{i,j} = \begin{cases} y_{0,j}^{r_i} & (j \neq i \bmod 2k), \\ s y_{1,i}^{r_i} & (j = i \bmod 2k). \end{cases}$$

  Note that all of the subscribers in $\mathcal{U}_i$ can be revoked by replacing $s y_{1,i}^{r_i}$ with g^{z_i} where $z_i \in_{\mathrm{R}} \mathbb{Z}_q$ is a random number.

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

$$\begin{aligned} &\left\{ \left(h_{i,0} \times h_{i,1}^{u} \times \cdots \times h_{i,2k-1}^{u^{2k-1}} \right) \Big/ \hat{h}_i^{f_i(u)} \right\}^{1/u^{i \bmod 2k}} \\ &\quad = \left\{ \left(y_{0,0}^{r_i} \times y_{0,1}^{r_i u} \times \cdots \times y_{1,i}^{r_i u^{i \bmod 2k}} \times \cdots \times y_{0,2k-1}^{r_i u^{2k-1}} \right) \Big/ g^{r_i f_i(u)} \right\}^{1/u^{i \bmod 2k}} \\ &\quad = \left\{ s^{u^{i \bmod 2k}} g^{r_i \sum_{j=0}^{2k-1} a_{i,j} u^j} \Big/ g^{r_i f_i(u)} \right\}^{1/u^{i \bmod 2k}} \\ &\quad = s. \end{aligned}$$

Black-box tracing: The black-box tracing algorithm is based on that of [8]. The difference is that while in [8] suspects must be narrowed down to k subscribers before the execution of black-box confirmation, in ours no such preprocessing, which runs in exponential time, is needed. The inputs of the tracing algorithm are $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$ and the pirate decoder, and the output is a traitor's ID.

For simplicity, we assume that $|\mathcal{U}_0| = \cdots = |\mathcal{U}_{\ell-1}| = 2k,\ \ell = n/2k,\ 2k|n$. Label all of the elements in $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$ as follows.

$$\begin{aligned}\mathcal{U}_0 &= \{u_1, \ldots, u_{2k}\},\\ \mathcal{U}_1 &= \{u_{2k+1}, \ldots, u_{4k}\},\\ &\vdots\\ \mathcal{U}_{\ell-1} &= \{u_{n-2k+1}, \ldots, u_n\}.\end{aligned}$$

For $1 \leq j \leq n$, repeat the following procedure.

- Set $ctr_j = 0$ and then repeat the following test m times. In each test, the session key s and random numbers R_0, R_1 are chosen randomly.
  1. Set $\mathcal{X} = \{u_1, \ldots, u_j\}$ and build the header $H = (H_0, \ldots, H_{\ell-1})$ by repeating the following procedure for $0 \leq i \leq \ell - 1$. The same notations are used as in the encryption phase and a random number $z_i \in_{\mathrm{R}} \mathbb{Z}_q$ is selected randomly in each time.
     - If there exists a subset $\mathcal{U}_t$ $(0 \leq t \leq \ell - 1)$ s.t. $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$, then first, suppose that $\mathcal{U}_t \setminus \mathcal{X} = \{x_1, \ldots, x_w\}$ and choose $2k - w - 1$ distinct elements $x_{w+1}, \ldots, x_{2k-1} \in_{\mathrm{R}} \mathbb{Z}_q \setminus (\mathcal{U} \cup \{0\})$ when $2k - w - 1 > 0$. Secondly, find $c_0, \ldots, c_{2k-1} \in \mathbb{Z}_q$ s.t. $\sum_{j=0}^{2k-1} c_j x_\alpha^j = 0 \bmod q$ for $1 \leq \alpha \leq 2k - 1$. Finally, compute H_t as follows.
       $$\begin{aligned}\hat{h}_t &= g^{R_1},\\ h_{t,j} &= \begin{cases} g^{c_j} y_{0,j}^{R_1} & (j \neq t \bmod 2k),\\ s g^{c_j} y_{1,t}^{R_1} & (j = t \bmod 2k).\end{cases}\end{aligned}$$
       For $i \neq t$, set $r_i = R_0$ if $\mathcal{X} \cap \mathcal{U}_i = \emptyset$. Otherwise ($\mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i$), set $r_i = R_0$ or R_1. Then, compute H_i as follows.
       $$\begin{aligned}\hat{h}_i &= g^{r_i},\\ h_{i,j} &= \begin{cases} y_{0,j}^{R_0} & (j \neq i \bmod 2k, r_i = R_0),\\ g^{c_j} y_{0,j}^{R_1} & (j \neq i \bmod 2k, r_i = R_1),\\ s y_{1,i}^{R_0} & (j = i \bmod 2k, \mathcal{X} \cap \mathcal{U}_i = \emptyset),\\ g^{z_i} & (j = i \bmod 2k, \mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i).\end{cases}\end{aligned}$$
     - Otherwise ($\mathcal{X} \cap \mathcal{U}_i = \emptyset$ or $\mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i$ for any i), H_i is the same as in the encryption phase.
  2. Give H to the pirate decoder and observe its output.
  3. If it decrypts correctly, then increment ctr_j by one. (If a self-defensive reaction is triggered, then decide that the subscriber u_j is a traitor.)

Finally, find an integer $j \in \{1, \ldots, n\}$ s.t. $ctr_{j-1} - ctr_j$ is the maximum and then decide that the subscriber u_j is a traitor, where $ctr_0 = m$.

4 Security

The security of our scheme is based on the difficulty of the Decision Diffie-Hellman problem (DDH) [1]. Informally, the assumption that DDH in G_q is intractable means that no probabilistic polynomial-time (p.p.t. for short) algorithm can distinguish with non-negligible advantage between the two distributions $\langle g_1, g_2, g_1^a, g_2^a\rangle$ and $\langle g_1, g_2, g_1^a, g_2^b\rangle$ where $g_1,\ g_2 \in_{\mathrm{R}} G_q$ and $a, b \in_{\mathrm{R}} \mathbb{Z}_q$. We call a 4-tuple coming from the former distribution as a Diffie-Hellman tuple. Let $\mathcal{M}^{\mathrm{DDH}}$ be a p.p.t. algorithm which solves DDH in G_q. For two p.p.t. algorithms $\mathcal{M}_0, \mathcal{M}_1$, we mean by $\mathcal{M}_0 \Rightarrow \mathcal{M}_1$ that the existence of $\mathcal{M}_0$ implies that of $\mathcal{M}_1$ and by $\mathcal{M}_0 \Leftrightarrow \mathcal{M}_1$ that $\mathcal{M}_0 \Rightarrow \mathcal{M}_1$ and $\mathcal{M}_1 \Rightarrow \mathcal{M}_0$.

4.1 Indistinguishability of a session key

Theorem 1 (Indistinguishability of a session key) *When given a header, the computational complexity for the non-subscribers to distinguish the session key corresponding to the header from a random element in G_q is as difficult as DDH in G_q.*

Proof Let $\mathcal{M}_{\bar{\mathcal{U}}}^{\mathrm{dist}}$ be a p.p.t. algorithm the non-subscribers use to distinguish between the session key corresponding to the header and a random element in G_q. We prove that $\mathcal{M}_{\bar{\mathcal{U}}}^{\mathrm{dist}} \Leftrightarrow \mathcal{M}^{\mathrm{DDH}}$. First, it is clear that $\mathcal{M}^{\mathrm{DDH}} \Rightarrow \mathcal{M}_{\bar{\mathcal{U}}}^{\mathrm{dist}}$. Secondly, we show that $\mathcal{M}_{\bar{\mathcal{U}}}^{\mathrm{dist}} \Rightarrow \mathcal{M}^{\mathrm{DDH}}$ by constructing $\mathcal{M}^{\mathrm{DDH}}$ using $\mathcal{M}_{\bar{\mathcal{U}}}^{\mathrm{dist}}$ as a subroutine. The construction of $\mathcal{M}^{\mathrm{DDH}}$ is as follows.

Algorithm 1 (P.p.t. algorithm $\boldsymbol{\mathcal{M}^{\mathrm{DDH}}}$)
Input: a challenge 4-tuple, (g_1, g_2, g_3, g_4).
Output: "Diffie-Hellman tuple" or "Random tuple."

Step 1. Choose a set of subscribers $\mathcal{U}$ ($\subseteq \mathbb{Z}_q \backslash \{0\}$) and split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. For $0 \le i \le \ell-1,\ 0 \le j \le 2k-1$, choose random numbers $\mu, \lambda_i, a_j \in_{\mathrm{R}} \mathbb{Z}_q$ and compute the public key $e = (g_1, g_1^{a_0}, \ldots, g_1^{a_{2k-1}}, g_1^{b_0}, \ldots, g_1^{b_{\ell-1}})$ where $g_1^{b_i} = g_1^{\lambda_i} g_2^{\mu}$.

Step 2. Select the session key $s \in_{\mathrm{R}} G_q$ and a random number $r \in_{\mathrm{R}} \mathbb{Z}_q$. Compute the header $H = (H_0, \ldots, H_{\ell-1})$ by repeating the following procedure for $0 \le i \le \ell - 1$.
- Set $B_i = 0$ or 1 and compute H_i as follows.

$$H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1}),$$
$$\hat{h}_i = g_1^{B_i r} g_3,$$
$$h_{i,j} = \begin{cases} \left(g_1^{B_i r} g_3\right)^{a_j} & (j \ne i \bmod 2k), \\ s\left(g_1^{B_i r} g_3\right)^{\lambda_i} \left(g_2^{B_i r} g_4\right)^{\mu} & (j = i \bmod 2k). \end{cases}$$

Observe that if the challenge 4-tuple is a Diffie-Hellman tuple, the session key corresponding to the header is s. Otherwise, it is a random element in G_q.

Step 3. Give s, H, e to $\mathcal{M}_{\mathcal{U}}^{\text{dist}}$. If $\mathcal{M}_{\mathcal{U}}^{\text{dist}}$ decides that s is the session key corresponding to H, then output "Diffie-Hellman tuple." Otherwise, output "Random tuple." Since $\mathcal{M}_{\mathcal{U}}^{\text{dist}}$ behaves differently for session keys and random elements in G_q, $\mathcal{M}^{\text{DDH}}$ can solve the given DDH challenge. This completes the proof. □

4.2 Black-box traceability

Recall that valid and invalid inputs denote headers for the normal broadcast and those for black-box tracing respectively. In our tracing algorithm subscribers in $\mathcal{X}$ are revoked in invalid inputs. The following three lemmas are used to prove black-box traceability of our scheme.

Lemma 1 (Indistinguishability of an input) *The computational complexity for any coalition of k non-revoked subscribers to distinguish a valid input from an invalid one is as difficult as DDH in G_q.*

Proof Let $\mathcal{C}$ be a set of k non-revoked subscribers in a coalition and $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ be a p.p.t. algorithm the coalition $\mathcal{C}$ uses to distinguish a valid input from an invalid one. We prove that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Leftrightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{X} \cap \mathcal{C} = \emptyset,\ |\mathcal{C}| = k$. First, it is clear that $\mathcal{M}^{\text{DDH}} \Rightarrow \mathcal{D}_{\mathcal{C}}^{\text{dist}}$ for any $\mathcal{C}$ with $\mathcal{X} \cap \mathcal{C} = \emptyset,\ |\mathcal{C}| = k$. Secondly, we show that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{X} \cap \mathcal{C} = \emptyset,\ |\mathcal{C}| = k$ by constructing $\mathcal{M}^{\text{DDH}}$ using $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ as a subroutine. The construction of $\mathcal{M}^{\text{DDH}}$ is as follows.

Algorithm 2 (P.p.t. algorithm $\mathcal{M}^{\text{DDH}}$)
Input: a challenge 4-tuple, (g_1, g_2, g_3, g_4).
Output: "Diffie-Hellman tuple" or "Random tuple."

Step 1. Choose a set of subscribers $\mathcal{U}$ $(\subseteq \mathbb{Z}_q \backslash \{0\})$ and split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. Select a set of revoked subscribers $\mathcal{X}$ $(\subseteq \mathcal{U})$ with a condition that there is at most one subset $\mathcal{U}_i$ $(0 \le i \le \ell - 1)$ s.t. $\mathcal{U}_i \cap \mathcal{X} \neq \emptyset$, $\mathcal{U}_i \cap \mathcal{X} \neq \mathcal{U}_i$. Then, choose a set of k colluders $\mathcal{C}$ s.t. $\mathcal{X} \cap \mathcal{C} = \emptyset$.

Step 2. Suppose that $\mathcal{C} = \{x_1, \ldots, x_k\}$. Choose $k-1$ distinct elements $x_{k+1}, \ldots, x_{2k-1} \in_{\text{R}} \mathbb{Z}_q \backslash \mathcal{C}$ and random numbers $\beta_1, \ldots, \beta_k, \lambda, \mu, \psi_t, \omega_t \in_{\text{R}} \mathbb{Z}_q$ for $k+1 \le t \le 2k-1$. Then, there exists a unique polynomial $\alpha(x) = \sum_{m=0}^{2k-1} \alpha_m x^m \bmod q$ s.t. $g_1^{\alpha_0} = g_1^{\lambda} g_2^{\mu}$ and

$$\begin{aligned}(\alpha(x_1), \ldots, \alpha(x_{2k-1}))^{\text{T}} &= (\beta_1, \ldots, \beta_{2k-1})^{\text{T}} \\ &= (\alpha_0, \ldots, \alpha_0)^{\text{T}} + V(\alpha_1, \ldots, \alpha_{2k-1})^{\text{T}} \bmod q, \\ g_1^{\beta_t} &= g_1^{\psi_t} g_2^{\omega_t} \qquad (k+1 \le t \le 2k-1),\end{aligned}$$

where

$$V = \begin{pmatrix} x_1 & \ldots & x_1^{2k-1} \\ \vdots & \ddots & \vdots \\ x_{2k-1} & \ldots & x_{2k-1}^{2k-1} \end{pmatrix} \bmod q.$$

Since V is the Vandermonde matrix, we obtain

$$(\alpha_1,\ldots,\alpha_{2k-1})^{\mathrm{T}} = V^{-1}(\beta_1-\alpha_0,\ldots,\beta_{2k-1}-\alpha_0)^{\mathrm{T}} \bmod q.$$

Let $(v_{m,1},\ldots,v_{m,2k-1})$ be the mth row of V^{-1}. For $1 \le m \le 2k-1$, α_m is represented as follows.

$$\begin{aligned}\alpha_m &= v_{m,1}(\beta_1-\alpha_0)+\cdots+v_{m,2k-1}(\beta_{2k-1}-\alpha_0)\\ &= v_{m,1}\beta_1+\cdots+v_{m,2k-1}\beta_{2k-1}-\alpha_0(v_{m,1}+\cdots+v_{m,2k-1}) \bmod q.\end{aligned}$$

Therefore, $g_1^{\alpha_m}$ is calculated as follows.

$$g_1^{\alpha_m} = g_1^{v_{m,1}\beta_1+\cdots+v_{m,2k-1}\beta_{2k-1}} \Big/ \left(g_1^{\lambda} g_2^{\mu}\right)^{v_{m,1}+\cdots+v_{m,2k-1}}.$$

Suppose that $x_j \in \mathcal{U}_{i_j}$ $(1 \le j \le k,\ i_j \in \{0,\ldots,\ell-1\})$ and define $\mathcal{J} = \{i_j | 1 \le j \le k,\ x_j \in \mathcal{U}_{i_j}\}$. Choose random numbers $\lambda_i, \mu_i \in_{\mathrm{R}} \mathbb{Z}_q$ for $0 \le i \le \ell-1$ and $\delta_{i_j} \in_{\mathrm{R}} \mathbb{Z}_q$ for all i_j's in $\mathcal{J}$. Then, there exists a unique element $\gamma_{i_j} \in \mathbb{Z}_q$ for each $i_j \in \mathcal{J}$ s.t.

$$\begin{aligned}\delta_{i_j} &= b_{i_j} + \gamma_{i_j} - \alpha_{i_j \bmod 2k} \qquad (i_j \in \mathcal{J}),\\ g_1^{b_i} &= g_1^{\lambda_i} g_2^{\mu_i} \qquad (0 \le i \le \ell-1).\end{aligned}$$

We plan to compute the subscriber x_j's personal key (x_j, i_j, d_j) as follows.

$$\begin{aligned}d_j &= \alpha(x_j) + \delta_{i_j} x_j^{i_j \bmod 2k}\\ &= \alpha_0 + \alpha_1 x_j + \cdots + b_{i_j} x_j^{i_j \bmod 2k} + \cdots + \alpha_{2k-1} x_j^{2k-1} + \gamma_{i_j} x_j^{i_j \bmod 2k}.\end{aligned}$$

To satisfy $d_j = f_{i_j}(x_j)$ where f is the key-generation function defined in 3.2, the coefficients $a_0,\ldots,a_{2k-1}$ are represented as follows. There are at least k elements in $\{0,\ldots,2k-1\} \setminus \{i_j \bmod 2k | i_j \in \mathcal{J}\}$ and we can select k such elements $\theta_1,\ldots,\theta_k$. Then, compute $g_1^{\alpha'_{\theta_1}},\ldots,g_1^{\alpha'_{\theta_k}}$ s.t.

$$\begin{aligned}g_1^{\sum_{\tau\in\{\theta_1,\ldots,\theta_k\}} \alpha'_\tau x_j^\tau} &= g_1^{\gamma_{i_j} x_j^{i_j \bmod 2k}}\\ &= \left(g_1^{\delta_{i_j}} g_1^{\alpha_{i_j \bmod 2k}} \Big/ g_1^{b_{i_j}}\right)^{x_j^{i_j \bmod 2k}} \qquad (1 \le j \le k).\end{aligned}$$

Finally, compute $g_1^{a_m}$ $(0 \le m \le 2k-1)$ and build the public key e.

$$\begin{aligned}g_1^{a_m} &= \begin{cases} g_1^{\alpha_m} & (m \notin \{\theta_1,\ldots,\theta_k\}),\\ g_1^{\alpha_m} g_1^{\alpha'_m} & (m \in \{\theta_1,\ldots,\theta_k\}),\end{cases}\\ e &= (g_1, g_1^{a_0},\ldots,g_1^{a_{2k-1}}, g_1^{b_0},\ldots,g_1^{b_{\ell-1}}).\end{aligned}$$

Step 3. Select the session key $s \in_{\mathrm{R}} G_q$ and a random number $r \in_{\mathrm{R}} \mathbb{Z}_q$. Build the header $H = (H_0,\ldots,H_{\ell-1})$ by repeating the following procedure for $0 \le i \le \ell-1$.

– If $\mathcal{U}_i \cap \mathcal{X} = \emptyset$, set $B_i = 0$. If $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$, set $B_i = 0$ or 1. Otherwise ($\mathcal{U}_i \cap \mathcal{X} \neq \emptyset$, $\mathcal{U}_i \cap \mathcal{X} \neq \mathcal{U}_i$), set $B_i = 1$. Then, compute H_i as follows.

$$
\begin{aligned}
H_i &= (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1}),\\
\hat{h}_i &= \begin{cases} g_1^r & (B_i = 0),\\ g_3 & (B_i = 1),\end{cases}\\
h_{i,j} &= \begin{cases} g_1^{a_j r} & (j \neq i \bmod 2k, B_i = 0),\\ g_3^{a_j} & (j \neq i \bmod 2k, B_i = 1),\\ s g_1^{b_i r} & (j = i \bmod 2k, B_i = 0),\\ s g_3^{\lambda_i} g_4^{\mu_i} & (j = i \bmod 2k, B_i = 1),\end{cases}\\
g_3^{a_j} &= \begin{cases} g_3^{\alpha_j} & (j \notin \{\theta_1, \ldots, \theta_k\}),\\ g_3^{\alpha_j} g_3^{\alpha'_j} & (j \in \{\theta_1, \ldots, \theta_k\}),\end{cases}\\
g_3^{\alpha_j} &= g_3^{v_{j,1}\beta_1 + \cdots + v_{j,k}\beta_k} \prod_{t=k+1}^{2k-1} \left(g_3^{\psi_t} g_4^{\omega_t}\right)^{v_{j,t}} \Big/ \left(g_3^{\lambda} g_4^{\mu}\right)^{v_{j,1} + \cdots + v_{j,2k-1}},
\end{aligned}
$$

where $g_3^{\alpha'_{\theta_1}}, \ldots, g_3^{\alpha'_{\theta_k}}$ are computed from the following system of equations.

$$
g_3^{\sum_{\tau \in \{\theta_1, \ldots, \theta_k\}} \alpha'_\tau x_z^\tau} = \left(g_3^{\delta_{i_z}} g_3^{\alpha_{i_z \bmod 2k}} \Big/ g_3^{\lambda_{i_z}} g_4^{\mu_{i_z}}\right)^{x_z^{i_z \bmod 2k}} \qquad (1 \leq z \leq k).
$$

Observe that if the challenge 4-tuple is a Diffie-Hellman tuple, H is a valid input. Otherwise, it is an invalid one in which the k colluders in $\mathcal{C}$ are not revoked.

Step 4. Give $H, e, (x_1, i_1, d_1), \ldots, (x_k, i_k, d_k)$ to $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$. If $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ decides that H is a valid input, then output "Diffie-Hellman tuple." Otherwise output "Random tuple." Since $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ behaves differently for valid inputs and invalid ones, $\mathcal{M}^{\text{DDH}}$ can solve the given DDH challenge.

Since $\mathcal{C}$ with $\mathcal{X} \cap \mathcal{C} = \emptyset$, $|\mathcal{C}| = k$ can be chosen arbitrarily in Step 1, it holds that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{X} \cap \mathcal{C} = \emptyset$, $|\mathcal{C}| = k$. This completes the proof. □

Lemma 2 (Secrecy of a session key in an invalid input) *When given an invalid input, the computational complexity for any coalition of k subscribers revoked in the invalid input to compute the session key corresponding to the input is at least as difficult as DDH in G_q.*

Proof Let $\mathcal{C}$ be a set of k colluders revoked in the invalid input and $\mathcal{M}_{\mathcal{C}}^{\text{comp}}$ be a p.p.t. algorithm the coalition $\mathcal{C}$ uses to compute the session key corresponding to the input. Let $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$ be a p.p.t. algorithm the coalition $\mathcal{C}$ uses to distinguish the session key corresponding to the input from a random element in G_q. We prove that $\mathcal{M}_{\mathcal{C}}^{\text{comp}} \Rightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{X}$, $|\mathcal{C}| = k$.

Since it is clear that $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$ can be constructed by using $\mathcal{M}_{\mathcal{C}}^{\text{comp}}$ as a subroutine, it holds that $\mathcal{M}_{\mathcal{C}}^{\text{comp}} \Rightarrow \mathcal{M}_{\mathcal{C}}^{\text{dist}}$ for any $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{X}$, $|\mathcal{C}| = k$. Therefore, we show that $\mathcal{M}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{X}$, $|\mathcal{C}| = k$ by constructing $\mathcal{M}^{\text{DDH}}$ using $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$ as a subroutine. The construction of $\mathcal{M}^{\text{DDH}}$ is as follows.

Algorithm 3 (P.p.t. algorithm $\mathcal{M}^{\mathbf{DDH}}$)
Input: a challenge 4-tuple, (g_1, g_2, g_3, g_4).
Output: “Diffie-Hellman tuple” or “Random tuple.”

Step 1. Choose a set of subscribers $\mathcal{U}$ ($\subseteq \mathbb{Z}_q \backslash \{0\}$) and split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. Select a set of revoked subscribers $\mathcal{X}$ ($\subseteq \mathcal{U}$) with a condition that there is at most one subset $\mathcal{U}_i$ $(0 \le i \le \ell-1)$ s.t. $\mathcal{U}_i \cap \mathcal{X} \neq \emptyset$, $\mathcal{U}_i \cap \mathcal{X} \neq \mathcal{U}_i$. Then, choose a set of k colluders $\mathcal{C}$ s.t. $\mathcal{C} \subseteq \mathcal{X}$.

Step 2. Suppose that $\mathcal{C} = \{x_1, \ldots, x_k\}$. Construct the personal key (x_j, i_j, d_j) given to the subscriber, $x_j \in \mathcal{U}_{i_j}$, and the public key $e = (g_1, g_1^{a_0}, \ldots, g_1^{a_{2k-1}}, g_1^{b_0}, \ldots, g_1^{b_{\ell-1}})$ by executing the same procedure as in Step 2 of Algorithm 2.

Step 3. Select the session key $s \in_{\mathrm{R}} G_q$ and random numbers $r, x, y \in_{\mathrm{R}} \mathbb{Z}_q$. Build the header $H = (H_0, \ldots, H_{\ell-1})$ by repeating the following procedure to compute $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ for $0 \le i \le \ell-1$.

– If $\mathcal{X} \cap \mathcal{U}_i = \emptyset$, then compute H_i as follows.

$$\hat{h}_i = g_3^r,$$
$$h_{i,j} = \begin{cases} g_3^{a_j r} & (j \neq i \bmod 2k), \\ s\left(g_3^{\lambda_i} g_4^{\mu_i}\right)^r & (j = i \bmod 2k). \end{cases}$$

– If $\mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i$, then set $B_i = 0$ or 1 and compute H_i as follows. In each time, a random number $z_i \in_{\mathrm{R}} \mathbb{Z}_q$ is selected randomly.

$$\hat{h}_i = \begin{cases} g_3^r & (B_i = 0), \\ g_1^x g_3^y & (B_i = 1), \end{cases}$$
$$h_{i,j} = \begin{cases} h'_{i,j} & (j \neq i \bmod 2k), \\ g_1^{z_i} & (j = i \bmod 2k), \end{cases}$$

where if there exists a subset $\mathcal{U}_t$ $(0 \le t \le \ell-1)$ s.t. $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$, then

$$h'_{i,j} = \begin{cases} g_3^{a_j r} & (B_i = 0), \\ g_1^{c_j} (g_1^x g_3^y)^{a_j} & (B_i = 1). \end{cases}$$

Otherwise ($\mathcal{X} \cap \mathcal{U}_i = \emptyset$ or $\mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i$ for any i),

$$h'_{i,j} = \begin{cases} g_3^{a_j r} & (B_i = 0), \\ (g_1^x g_3^y)^{a_j} & (B_i = 1), \end{cases}$$

– If $\mathcal{X} \cap \mathcal{U}_i \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_i \neq \mathcal{U}_i$, then first, suppose that $\mathcal{U}_i \setminus \mathcal{X} = \{u_1, \ldots, u_w\}$ and choose $2k-w-1$ distinct elements $u_{w+1}, \ldots, u_{2k-1} \in_{\mathrm{R}} \mathbb{Z}_q \setminus (\mathcal{U} \cup \{0\})$ when $2k-w-1 > 0$. Secondly, find $c_0, \ldots, c_{2k-1} \in_{\mathrm{R}} \mathbb{Z}_q$ s.t. $\sum_{j=0}^{2k-1} c_j u_\alpha^j = 0 \bmod q$ for $1 \le \alpha \le 2k-1$. Finally, compute H_i as follows.

$$\hat{h}_i = g_1^x g_3^y,$$
$$h_{i,j} = \begin{cases} g_1^{c_j} (g_1^x g_3^y)^{a_j} & (j \neq i \bmod 2k), \\ s g_1^{c_j} (g_1^x g_3^y)^{\lambda_i} (g_2^x g_4^y)^{\mu_i} & (j = i \bmod 2k). \end{cases}$$

In this procedure, $g_3^{a_j}$ is computed as in Step 3 of Algorithm 2. Observe that if the challenge 4-tuple is a Diffie-Hellman tuple, s is the session key corresponding to H. Otherwise, it is not.

Step 4. Give $s, H, e, (x_1, i_1, d_1), \ldots, (x_k, i_k, d_k)$ to $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$. If $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$ decides that s is the session key corresponding to H, then output "Diffie-Hellman tuple." Otherwise output "Random tuple." Since $\mathcal{M}_{\mathcal{C}}^{\text{dist}}$ behaves differently for session keys and random elements in G_q, $\mathcal{M}^{\text{DDH}}$ can solve the given DDH challenge.

Since $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{X},\ |\mathcal{C}| = k$ can be chosen arbitrarily in Step 1, it holds that $\mathcal{M}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{\text{DDH}}$ for any $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{X},\ |\mathcal{C}| = k$. This completes the proof. □

Lemma 3 (Indistinguishability of a suspect) *The computational complexity for any coalition of k subscribers to distinguish (1) an invalid input in which a given subscriber other than the k ones is not revoked from (2) an invalid one in which the subscriber is revoked is as difficult as DDH in G_q.*

Sketch of Proof Due to space limitation, we describe a sketch of the proof. Let $\mathcal{C}$ be a set of k colluders. Let $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$ be a p.p.t. algorithm the coalition $\mathcal{C}$ uses to distinguish an invalid input in which the given subscriber is not revoked from an invalid one in which the subscriber is revoked. Similarly in the proofs of the other lemmas, we construct $\mathcal{M}^{\text{DDH}}$ using $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$ as a subroutine.

Algorithm 4 (P.p.t. algorithm $\mathcal{M}^{\text{DDH}}$)
Input: a challenge 4-tuple, (g_1, g_2, g_3, g_4).
Output: "Diffie-Hellman tuple" or "Random tuple."

Step 1. Choose a set of subscribers $\mathcal{U}$ $(\subseteq \mathbb{Z}_q \backslash \{0\})$ and split $\mathcal{U}$ into ℓ disjoint subsets $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. Select a set of k colluders $\mathcal{C}$ and one subscriber $u \in_{\text{R}} \mathcal{U} \setminus \mathcal{C}$. Suppose that $u \in \mathcal{U}_t$, $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ for $0 \le i \le t-1$, and $\mathcal{U}_i \cap \mathcal{X} = \emptyset$ for $t+1 \le i \le \ell - 1$. There are three possible relations between $\mathcal{U}_t$ and $\mathcal{X}$: (1) $\mathcal{U}_t \cap \mathcal{X} \ne \mathcal{U}_t$, $\mathcal{U}_t \cap \mathcal{X} \ne \emptyset$ both when $u \notin \mathcal{X}$ and $u \in \mathcal{X}$, (2) $\mathcal{U}_t \cap \mathcal{X} = \emptyset$ when $u \notin \mathcal{X}$, and $\mathcal{U}_t \cap \mathcal{X} = \{u\}$ when $u \in \mathcal{X}$, (3) $\mathcal{U}_t \cap \mathcal{X} = \mathcal{U}_t \setminus \{u\}$ when $u \notin \mathcal{X}$, and $\mathcal{U}_t \cap \mathcal{X} = \mathcal{U}_t$ when $u \in \mathcal{X}$.

Step 2. Suppose that $\mathcal{C} = \{x_1, \ldots, x_k\}$. Construct the personal key (x_j, i_j, d_j) given to the subscriber, $x_j \in \mathcal{U}_{i_j}$, and the public key $e = (g_1, g_1^{a_0}, \ldots, g_1^{a_{2k-1}}, g_1^{b_0}, \ldots, g_1^{b_{\ell-1}})$ by executing the same procedure as in Step 2 of Algorithm 2.

Step 3. Build the header H in which (1) if the challenge 4-tuple is a Diffie-Hellman tuple, the subscriber u is not revoked and (2) otherwise, the subscriber u is revoked, in each case. The construction of H is similar to that in Step 3 of Algorithm 3.

Step 4. Give $u, H, e, (x_1, i_1, d_1), \ldots, (x_k, i_k, d_k)$ to $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$. Since $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$ behaves differently for invalid inputs in which the subscriber u is not revoked and invalid ones in which the subscriber u is revoked, $\mathcal{M}^{\text{DDH}}$ can solve the given DDH challenge. □

From Lemma 1, Lemma 2, and Lemma 3, it follows that the next theorem holds.

Theorem 2 (Black-box traceability) *In the proposed scheme, from the pirate decoder constructed by a coalition of at most k traitors, at least one of them can be identified with probability $1-\varepsilon$ where ε is negligible.*

Proof Recall that ctr_j ($0 \leq ctr_j \leq m$) denotes the number of times of observing that the pirate decoder decrypts correctly the input in which $\mathcal{X} = \{u_1, \ldots, u_j\}$, i.e., the subscribers $u_1, \ldots, u_j$ are revoked. Define $j = 0$ if $\mathcal{X} = \emptyset$, i.e., the input is valid. It is clear that $ctr_0 = m$. From Lemma 2, it holds that $ctr_n = 0$ with overwhelming probability. From the triangular inequality, it follows that there exists an integer $j \in \{1, \ldots, n\}$ s.t. $ctr_{j-1} - ctr_j \geq m/n$. If the subscriber u_j is not a traitor, $ctr_{j-1} - ctr_j \ll m/n$ since it follows from Lemma 3 that the pirate decoder cannot distinguish an invalid input in which $\mathcal{X} = \{u_1, \ldots, u_{j-1}\}$ from an invalid one in which $\mathcal{X} = \{u_1, \ldots, u_j\}$ with non-negligible advantage. Therefore, the subscriber u_j is a traitor with overwhelming probability if $ctr_{j-1} - ctr_j$ is the maximum.

Next, consider the case where the reaction mechanism is activated. From Lemma 1, no such reaction is triggered as long as $\mathcal{X} \cap \mathcal{C} = \emptyset$ where $\mathcal{C}$ denotes a set of the colluders. Therefore, if the reaction is triggered in the input in which $\mathcal{X} = \{u_1, \ldots, u_j\}$, it holds that $\{u_1, \ldots, u_j\} \cap \mathcal{C} \neq \emptyset$. In this case, if the subscriber u_j is not a traitor, the pirate decoder must have taken the reaction in the previous input in which $\mathcal{X} = \{u_1, \ldots, u_{j-1}\}$ since it follows from Lemma 3 that the pirate decoder cannot distinguish an invalid input in which $\mathcal{X} = \{u_1, \ldots, u_j\}$ from an invalid one in which $\mathcal{X} = \{u_1, \ldots, u_{j-1}\}$ with non-negligible advantage. Hence, if the reaction is triggered in the input in which $\mathcal{X} = \{u_1, \ldots, u_j\}$, it holds that the subscriber u_j is a traitor with overwhelming probability. □

Note that our scheme can be easily applied to the case where the pirate decoder takes the reaction in a probabilistic way.

5 Efficiency

In Table 2, the previous schemes and ours are compared from the viewpoints of each subscriber's storage, the transmission overhead, the number of sets of suspects required for tracing, the detection probability, and the computational cost for decryption. The scheme of [2] is omitted since its efficiency is almost the same as that of [8] in the above criteria. We suppose that the standard ElGamal encryption scheme is straightforwardly used in the scheme of [6]

In the scheme of [6], the size of a personal key is determined by a constant c ($0 < c < 1$) selected when initializing the system. In the other schemes, the size of a personal key is constant. In the scheme of [8], the efficient transmission overhead which is linear only in k is achieved where k is the maximum coalition size. However, the scheme of [8] can only support black-box confirmation in which

Table 2. Efficiency comparison ($\mathcal{P}$, $\mathcal{S}$, $\mathcal{H}$: sets of possible personal keys, session keys, and headers respectively, n: the total number of subscribers, k: the maximum coalition size, c: a constant $(0 < c < 1)$, ε: negligible probability)

	Each subscriber's storage $(\log\lvert\mathcal{P}\rvert/\log\lvert\mathcal{S}\rvert)$	Transmission overhead $(\log\lvert\mathcal{H}\rvert/\log\lvert\mathcal{S}\rvert)$	# of sets of suspects for tracing	Detection probability	# of exp.'s for decryption
[8]	1	$2k+1$	$\binom{n}{k}$	$1-\varepsilon$	$O(k)$
[6]	$(1-c)^{-1}$	$(1-c)^{-1}n^{1-c}$	n^{1-c}	n^{-c}	$O((1-c)^{-1})$
[6] $(c=1/2)$	2	$2\sqrt{n}$	$\sqrt{n}$	$1/\sqrt{n}$	$O(1)$
Ours	1	$4k+n/2k+2$	n	$1-\varepsilon$	$O(k)$
Ours $(k=\sqrt{n/8})$	1	$2\sqrt{2n}+2$	n	$1-\varepsilon$	$O(\sqrt{n})$

only k suspects can be tested in one confirmation. Therefore, the tracer needs to execute the confirmation algorithm on all of the possible $\binom{n}{k}$ sets of suspects at the worst case, where n is the total number of subscribers. Since the number of sets of suspects required for tracing is directly affected to the running time of the tracing algorithm, the scheme of [8] is impractical from this viewpoint. On the other hand, in the scheme of [6] and ours the number of sets of suspects required for tracing is drastically reduced and hence the practical convergence time for tracing is achieved.

In the scheme of [6], the output of the tracing algorithm is the list of suspects in which at least one traitor is included with overwhelming probability. If the tracer attempts to identify the traitor only from the suspect list, the probability that the tracer correctly detect the traitor is n^{-c}, since the list size is n^c. Due to its combinatorial construction, there is a trade-off between the transmission overhead and the detection probability in the scheme of [6]. The value of c which gives the smallest header size and detection probability at the same time is $c = 1/2$ and in this case the header size is $O(\sqrt{n})$ and the detection probability is $1/\sqrt{n}$. Although the sublinear header size is achieved in the scheme, its detection probability becomes smaller as n gets larger.

In our scheme, efficient black-box tracing is achieved without the above list-tracing approach, i.e., there is no such trade-off. The header size is linear in k and the number of subsets of subscribers. Especially, if we set $k = \sqrt{n/8}$, the header size is $O(\sqrt{n})$, where we assume that the size of each subset is $2k$. The tracer can identify at least one traitor with overwhelming probability, regardless of n. By applying the key-generation method of [9] to the scheme of [8], our scheme enables the tracer to make it impossible for the revoked subscribers to compute the session key by substituting a random value for the element used only by the subscribers in one of the ℓ disjoint subsets if all of them in the subset are revoked. This helps to remove the restriction of the number of suspects in the previous schemes with black-box confirmation and hence efficient black-box tracing without sacrificing the detection probability is achieved. On the value of m, which is the number of repetition times of the test in the tracing algorithm, it is shown in [6] that at least one traitor can be identified with overwhelming probability if $m = O(n^2 \log^2 n)$. By using this result, it can be said that the running time of the tracing algorithm is $O(n^3 \log^2 n)$.

The main differences between the scheme of [6] and ours are the detection probability and the computational cost for decryption. While in the scheme of [6] the detection probability gets smaller as the value of n increases, in our scheme it is independent of n and always overwhelming. On the other hand, the scheme of [6] is efficient from the viewpoint of the computational cost for decryption. In the previous scheme, only a few exponentiations are needed, while the number of exponentiations required for decryption is $O(k)$ in ours. This can be alleviated by using a technique of vector-addition chain exponentiation [11, p.622].

6 Conclusions

In this paper, we have proposed a sublinear public-key black-box tracing scheme against self-defensive pirate decoders. This can be viewed as a solution to the open question to build a sublinear traitor tracing scheme that supports efficient black-box tracing against self-defensive pirate decoders with negligible probability of error.

References

1. D. Boneh: “The Decision Diffie-Hellman Problem”, In *Proc. of the Third Algorithmic Number Theory Symposium*, LNCS 1423, Springer-Verlag, pp. 48–63, 1998.
2. D. Boneh and M. Franklin: “An Efficient Public Key Traitor Tracing Scheme”, In *Proc. of CRYPTO ’99*, LNCS 1666, Springer-Verlag, pp. 338–353, 1999.
3. B. Chor, A. Fiat, and M. Naor: “Tracing Traitors”, In *Proc. of CRYPTO ’94*, LNCS 839, Springer-Verlag, pp. 257–270, 1994.
4. B. Chor, A. Fiat, M. Naor, and B. Pinkas: “Tracing Traitors”, *IEEE Transactions on Information Theory*, Vol. 46, No. 3, pp. 893–910, 2000.
5. Y. Dodis and N. Fazio: “Public Key Broadcast Encryption for Stateless Receivers”, *ACM Workshop on Digital Rights Management (DRM ’02)*, 2002.
6. A. Kiayias and M. Yung: “On Crafty Pirates and Foxy Tracers”, In *Proc of Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management (SPDRM ’01)*, LNCS 2320, Springer-Verlag, pp. 22–39, 2002.
7. K. Kurosawa and Y. Desmedt: “Optimum Traitor Tracing and Asymmetric Schemes”, In *Proc. of EUROCRYPT ’98*, LNCS 1403, Springer-Verlag, pp. 145–157, 1998.
8. K. Kurosawa and T. Yoshida: “Linear Code Implies Public-Key Traitor Tracing”, In *Proc. of PKC ’02*, LNCS 2274, Springer-Verlag, pp. 172–187, 2002.
9. T. Matsushita: “A Flexibly Revocable Key-Distribution Scheme for Efficient Black-Box Tracing”, In *Proc. of ICICS ’02*, LNCS 2513, Springer-Verlag, pp. 197–208, 2002.
10. T. Matsushita and H. Imai: “Black-box Traitor Tracing against Arbitrary Pirate Decoders”, In *Proc. of the 1st Workshop on Information Security Applications (WISA ’00)*, pp. 265–274, 2000.
11. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.