An Improved Public-key Tracing Scheme with Sublinear Ciphertext Size

Chiara Valentina Schiavo and Andrea Visconti
Dipartimento di Informatica e Comunicazione, Universitá degli Studi di Milano, via Comelico 39/41, 20135 Milano, Italy

Keywords: Traitor Tracing Schemes, Piracy, Digital Content Distribution Systems, Pirate Decoders, Traitors.

Abstract: To overcome the piracy problem in digital content distribution systems, a number of traitor tracing schemes have been suggested by researchers. The goal of these schemes is to enable the tracer to identify at least one of the traitors. In this context, Matsushita and Imai (2004) proposed a black-box tracing scheme with sublinear header size that is able to perform tracing of self-defensive pirate decoders. Kiayias and Pehlivanoglu (2009) proved that this scheme is vulnerable to an attack which allows an illicit decoder to recognize normal ciphertext to tracing ones and distinguish two consecutive tracing ciphertexts. For making the scheme no more susceptible to such attack, authors modified the encryption phase and assumed that traitors belong to the same user group. In this paper, we present a solution that has no traitors restrictions, repairing the scheme totally. In particular, we modified the tracing scheme proving that (a) a pirate decoder is not able to recognize normal ciphertext to tracing ones with sufficiently high probability, and (b) the statistical distance between two consecutive tracing operations is negligible under Decision Diffie Hellman assumption.

1 INTRODUCTION

Secure distribution of digital contents plays a key role in many applications such as Pay-TV systems, streaming media distributions, copyrighted material, etc. in which only authorized users should be able to use them. Since the main model for digital content distribution is virtual and not physical, malicious users may decrypt and redistribute digital content, disclose their personal key to unauthorized users, or build a pirate decoder. Therefore, the piracy problem needs to be addressed and traitor tracing can help us to mitigate this unwanted behavior. In a first high-level scenario, a broadcaster, or data supplier, encrypts the digital contents using a session key, blinds such key into the header, and then sent encrypted contents and headers to users. Authorized users, the subscribers, by means of a decoder, can retrieve the session key and subsequently decrypt the digital contents. On the other hand, malicious subscribers, the traitors, may build a pirate decoder with their own personal keys, allowing unauthorized users, also called pirates, to illegally decrypt the copyrighted material. In order to identify users involved in constructing a pirate decoder, a number of traitor tracing schemes have been suggested (Dodis and Fazio, 2002), (Kiayias and Yung, 2001), (Naor and Pinkas, 2010), (Naor and Pinkas, 1998), (Chor et al., 2000), (Kiayias and Pehlivanoglu, 2011). All such schemes enable a broadcaster to trace at least one traitor of the coalition. In 1994 Chor, Fiat and Naor (1994) introduced the concept of traitor tracing schemes to prevent the piracy. Boneh and Franklin (1999) suggested a deterministic public key traitor tracing scheme, applying error correcting techniques, while Kurosawa and Desmedt (1998) described a multiple-use traceability scheme which use small keys and short ciphertext. Taking into account memory capabilities and the ability of triggering self-defense mechanisms, Kiayias and Yung (2001) classified pirate decoders into four non-disjoint categories — i.e. resettable, history-recording, abrupt, and available. Moreover, authors introduced the concept of list-tracing, and presented a traitor tracing scheme that is successful against abrupt and resettable decoders. Dodis and Fazio (2002) described a public key broadcast encryption scheme for stateless receivers which reduces the public key size and user's storage. Unfortunately, such scheme is not effective against pirate decoder which are able to trigger a self-defense mechanism. Dwork, Lotspiech and Naor (1996), suggested the notion of self-enforcement for combating leakage of keys and deterring users from revealing sensitive information. An efficient revocation scheme based on secret sharing enhanced with traitor tracing and self-enforcement properties has been suggested by Naor and Pinkas (2010). In (Matsushita and Imai, 2004), the idea of Matsushita (2002) of the key generation method

Valentina Schiavo C. and Visconti A..
An Improved Public-key Tracing Scheme with Sublinear Ciphertext Size.
DOI: 10.5220/0004074003020307
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 302-307
ISBN: 978-989-8565-24-2

has been applied to (Kurosawa and Yoshida, 2002). They suggested an efficient black-box tracing scheme against abrupt pirate decoders, keeping the size of the header sublinear in the number of receivers. In a subsequent work Matsushita and Imai (2006) extended their previous scheme presented in (Matsushita and Imai, 2004) in order to reduce the header size. Kiayias and Pehlivanoglu (2009) showed that the traitor tracing scheme of Matsushita and Imai (2004) is susceptible to an attack that allows an illicit decoder to avoid tracing and accuse an innocent user. In this paper, we analyze the attack described by Kiayias and Pehlivanoglu (2009) which (a) exploits the distance between normal ciphertext from tracing ones and (b) is able to distinguish two consecutive tracing ciphertext with non-negligible probability. We improve the black-box tracing algorithm described by Matsushita and Imai (2004), showing that the restriction on the geometry of traitors suggested in (Kiayias and Pehlivanoglu, 2009) can be omitted. In particular, we suggest a way to repair the black-box tracing algorithm (Matsushita and Imai, 2004) in order to reduce the distance between normal and tracing ciphertext and moreover close the gap between two consecutive tracing ciphertext, making the scheme no more susceptible to the attack.

The paper is organized as follows. In Section 2 we recall the attack proposed by Kiayias and Pehlivanoglu (2009) on the Matsushita and Imai protocol. In Section 3, we suggest a new solution that repairs the scheme totally. Finally, in Section 4, a security proof of the protocol is presented.

2 THE ATTACK

Due to space limitations, the authors do not describe the Matsushita and Imai protocol (Matsushita and Imai, 2004). However, we briefly introduce the main parameters used in such protocol Let n be the number of users and k be the maximum number of traitors in a coalition. Let p and q be two primes such that $q \mid p-1$ and $q \geq n+2k-1$. G_q is a subgroup of $\mathbb{Z}_p^*$ of order q, g is a generator of subgroup G_q and $\mathcal{U} = \{u_1, \ldots, u_n\}$ is the set of all user where $\mathcal{U} \subseteq \mathbb{Z}_q \backslash \{0\}$. Let ctr_j be a counter used in the tracing phase in order to decide if the considered user u_j is a traitor or not. For generating the public key and users' private keys, the protocol splits the set of user $\mathcal{U}$ in ℓ disjoint subset $\mathcal{U}_0, \mathcal{U}_1, \ldots, \mathcal{U}_{\ell-1}$, $|\mathcal{U}_i| = 2k$ with $i = 0, \ldots, \ell-1$ and chooses $a_0, a_1, \ldots, a_{2k-1}, b_0, b_1, \ldots, b_{\ell-1} \in_R \mathbb{Z}_q$. The public key will be $e = (p, q, g, g^{a_0}, \ldots, g^{a_{2k-1}}, g^{b_0}, \ldots, g^{b_{\ell-1}}) = (p, q, g, y_{0,0}, \ldots, y_{0,2k-1}, y_{1,0}, \ldots, y_{1,l-1})$. The private key of user $u \in \mathcal{U}_i$, with $0 \leq i \leq \ell-1$, is $(u, i, f_i(u))$, where $f_i(u) = \sum_{j=0}^{2k-1} a_{i,j} u^j \bmod q$ with $a_{i,j} = a_j$ if $j \neq i \bmod 2k$ or $a_{i,j} = b_i$ otherwise. The encrypted headers sent to users can be represented as $H = (H_0, H_1, \ldots, H_{\ell-1})$. Each group $\mathcal{U}_i$ receives the header $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ where $\hat{h}_i = g^{r_i}$ with $r_i \in \{R_0, R_1\}$ where $R_0, R_1 \in_R \mathbb{Z}_q$ are random numbers. It is important to note that the header H_i can contain either the blinded session key $s \in_R G_q$ —chosen by the data supplier— or a revoking value —computed using a random value $z_i \in_R \mathbb{Z}_q$ —.

In (Kiayias and Pehlivanoglu, 2009), authors showed that the public-key black-box traitor tracing scheme in (Matsushita and Imai, 2004) is vulnerable to self-defense mechanism. The attack (Kiayias and Pehlivanoglu, 2009) relies on the possibility to distinguish normal ciphertext from tracing ones, monitoring the headers $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ sent to a coalition of k non-revoked traitors that belong to different subgroups $\mathcal{U}_i$, $i > t$. When tracing is disabled, each subgroup of users $\mathcal{U}_i$ receives $\hat{h}_i = g^{r_i}$ — recall that $r_i \in \{R_0, R_1\}$ uniformly at random. On the other hand, when tracing is enabled, these subgroups of users receive $\hat{h}_i = g^{R_0}$. Therefore, the probability that k traitors receive the same $\hat{h}_i$ is $1/2^k$ when normal ciphertext is sent, while is 1 when tracing. The statistical distance between these probability distribution converges to 1 when the number of traitors grows (see (Kiayias and Pehlivanoglu, 2009), Theorem 1). Monitoring header H_i, a pirate decoder is able to distinguish these distributions with a non-negligible probability and trigger a self-defensive mechanism. Moreover, the pirate decoder is able to distinguish the gap between two consecutive tracing ciphertexts $CTrace(e, j-1, s)$ and $CTrace(e, j, s)$ when $j = 1 \bmod 2k$. In the first case, i.e. $CTrace(e, j-1, s)$, all subgroups $\mathcal{U}_0, \ldots \mathcal{U}_{\ell-1}$ will receive either $r_i = R_0$ or $r_i = R_1$ at random. In the second case, i.e. $CTrace(e, j, s)$, there exists a subgroup $\mathcal{U}_t$, which contains u_j, such that $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$. Hence, subgroup $\mathcal{U}_t$ will receive $r_i = R_1$, subgroups $\mathcal{U}_0 \ldots \mathcal{U}_{t-1}$ receive $r_i = R_0$ or $r_i = R_1$ at random, and finally, subgroups $\mathcal{U}_{t+1} \ldots \mathcal{U}_{\ell-1}$ receive $r_i = R_0$. Exploiting the gap between $CTrace(e, j-1, s)$ and $CTrace(e, j, s)$, a pirate decoder is able to recognize two consecutive tracing ciphertexts and trigger a self-defense mechanism. As consequences of such mechanism, counter ctr_j of traitor u_j will be not increased, the probability that the difference $ctr_{j-1} - ctr_j$ gets the maximum value is dramatically reduced, tracing is avoided, and an innocent user is in fact accused with a non-negligible probability (see (Kiayias and Pehlivanoglu, 2009), Corollary 1). The main problem of the Matsushita and Imai

protocol is the possibility to recognize normal and tracing operations, exploiting the statistical distance between two probability distributions. For reducing such distance, Kiayias and Pehlivanoglu (Kiayias and Pehlivanoglu, 2009) introduce a random cutoff point d that is the switch point between $r_i = R_0$ to $r_i = R_1$. In particular, if $i \leq d$ then they set $r_i = R_1$, otherwise $r_i = R_0$. While encrypting, the data supplier selects a random integer $d \in \{0,\ldots,\ell-1\}$ and generates ℓ headers H_i, $i = 0,\ldots,\ell-1$, one for each subset of users $\mathcal{U}_i$, as follows

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

3 AN IMPROVED SCHEME

The introduction of a random cutoff point ensures that, monitoring the headers H_i, traitors are not able to distinguish normal ciphertext from tracing ones. This new approach fixes one problem, however, the gap between two consecutive tracing ciphertexts — i.e. the gap between $CTrace(e, j-1, \cdot)$ and $CTrace(e, j, \cdot)$ — still remain. In order to make traitors unable to recognize tracing activities, in (Kiayias and Pehlivanoglu, 2009) authors assume that all traitors are in the same user group $\mathcal{U}_i$ and they apply the tracing algorithm in parallel to $\mathcal{U}_0 \ldots \mathcal{U}_{\ell-1}$. For mitigating all these problems, we present a solution that repair the scheme totally without restriction on the geometry of traitors. In order to reduce the gap between two probability distributions, we suggest to modify the black box tracing phase, thereby preserving the encryption phase. In particular, when there exists a subset $\mathcal{U}_t$ with $0 \leq t \leq \ell-1$ such that $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$, $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$, we suggest to modify the construction of the header H_i for $i \neq t$ as follows

$$\hat{h}_i = g^{r_i}, \qquad r_i = R_0 \text{ or } R_1$$
$$h_{i,j} = \begin{cases} y_{0,j}^{R_0} & (j \neq i \bmod 2k, r_i = R_0) \\ g^{c_j} y_{0,j}^{R_1} & (j \neq i \bmod 2k, r_i = R_1) \\ s y_{1,i}^{R_0} & (j = i \bmod 2k, \mathcal{X} \cap \mathcal{U}_i = \emptyset, r_i = R_0) \\ s g^{c_j} y_{1,i}^{R_1} & (j = i \bmod 2k, \mathcal{X} \cap \mathcal{U}_i = \emptyset, r_i = R_1) \\ g^{z_i} & (j = i \bmod 2k, \mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i) \end{cases}$$

This new black box tracing phase ensures that traitors are not able to distinguish normal ciphertext from tracing ones, and at the same time, fixes the gap between two consecutive tracing ciphertexts. Tables 1 and 2 can help us to figure out the improvements suggested. In particular, first and second rows of Table 1 show the statistical gap between $CTrace(e, j-1, \cdot)$ and $CTrace(e, j, \cdot)$ in Matsushita-Imai protocol while third and fourth rows show how our solution closes such gap, making two consecutive tracing ciphertext indistinguishable. Table 2 shows how traitors can exploit the Matsushita-Imai distribution of r_i in order to distinguish tracing activities and how our approach avoids this unwanted behavior. It is not hard to show that in the improved scheme a set of users $\mathcal{U}_i$ — i.e. users that are not chosen to be revoked — are able to compute the session key s. In fact, suppose there exists a subgroup $\mathcal{U}_t$ with $0 \leq t < \ell$ such that $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$. Then, for $i \neq t$ users residing in the subset U_i such that $\mathcal{X} \cap \mathcal{U}_i = \emptyset$ and $r_i = R_1$ are able compute the session key s using the header H_i as shown in Equation 2

$$\left\{ \frac{h_{i,0} \times h_{i,1}^{u} \times \cdots \times h_{i,2k-1}^{u^{2k-1}}}{\hat{h}_i^{f_i(u)}} \right\}^{1/u^{i \bmod 2k}} \qquad (2)$$
$$= \left\{ \frac{s^{u^{i \bmod 2k}} g^{\sum_{j=0}^{2k-1} c_j u^j} g^{R_1 (\sum_{j=0}^{2k-1} a_j u^j + b_i u^i - a_i u^i)}}{g^{r_i f_i(u)}} \right\}^{1/u^{i \bmod 2k}}$$
$$= \left\{ s^{u^{i \bmod 2k}} \right\}^{1/u^{i \bmod 2k}} = s$$

4 PROOF OF SECURITY

The security proof of the improved tracing algorithm relies on the Decision Diffie-Hellman (DDH) problem (Boneh, 1998). For this reason, we will introduce (a) $\mathcal{M}^{DDH}$, a probabilistic polynomial time (p.p.t.) algorithm which solve the DDH problem in G_q and (b) three lemmas which can help us to show that the traitor tracing scheme we propose is able to identify at least one traitor of a coalition with non-negligible probability.

Lemma 4.1 (Indistinguishability of an input)**.** *The computational complexity for k non revoked subscribers to distinguish a valid input from an invalid one is as hard as DDH in G_q.*

Proof. Let $\mathcal{C}$ be a set of k non-revoked subscribers and $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ be a p.p.t. algorithm used by users in $\mathcal{C}$ to distinguish a valid from an invalid one. Let $\mathcal{M}^{DDH}$ be a p.p.t. algorithm that is able to solve the DDH problem in G_q. In particular, the p.p.t. algorithm $\mathcal{M}^{DDH}$ inputs a 4-tuple (g_1, g_2, g_3, g_4) and outputs whether such tuple is a Diffie-Hellman tuple or a Random tuple. We prove that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Leftrightarrow \mathcal{M}^{DDH}$ for any $\mathcal{C}$ such that $\mathcal{X} \cap \mathcal{C} = \emptyset$ and $|C| = k$. It is straightforward to prove that $\mathcal{M}^{DDH} \Rightarrow \mathcal{D}_{\mathcal{C}}^{\text{dist}}$, therefore we prove that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{DDH}$. Split the set of subscribers $\mathcal{U}$ into

Table 1: Distribution of r_i, case $CTrace(e, j-1, \cdot)$ and $CTrace(e, j, \cdot)$.

	$\mathcal{U}_0$	...	$\mathcal{U}_{t-1}$	$\mathcal{U}_t$	$\mathcal{U}_{t+1}$	...	$\mathcal{U}_{\ell-1}$
Matsushita-Imai scheme: $CTrace(e, j-1, \cdot)$	R_0/R_1	...	R_0/R_1	R_0/R_1	R_0/R_1	...	R_0/R_1
Matsushita-Imai scheme: $CTrace(e, j, \cdot)$	R_0/R_1	...	R_0/R_1	R_1	R_0	...	R_0
Our scheme: $CTrace(e, j-1, \cdot)$	R_0/R_1	...	R_0/R_1	R_0/R_1	R_0/R_1	...	R_0/R_1
Our scheme: $CTrace(e, j, \cdot)$	R_0/R_1	...	R_0/R_1	R_1	R_0/R_1	...	R_0/R_1

Table 2: Distribution of r_i with Normal Ciphertext (1st row) and Tracing Ciphertext (2nd, 3rd rows— case: $\exists \mathcal{U}_t (0 \le t \le \ell-1)$ such that $\mathcal{X} \cap \mathcal{U}_t \neq 0$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$).

	$\mathcal{U}_0$	...	$\mathcal{U}_{t-1}$	$\mathcal{U}_t$	$\mathcal{U}_{t+1}$	...	$\mathcal{U}_{\ell-1}$
Matsushita-Imai scheme - Normal ciphertext	R_0/R_1	...	R_0/R_1	R_0/R_1	R_0/R_1	...	R_0/R_1
Matsushita-Imai scheme - Tracing ciphertext	R_0/R_1	...	R_0/R_1	R_1	R_0	...	R_0
Our scheme - Tracing ciphertext	R_0/R_1	...	R_0/R_1	R_1	R_0/R_1	...	R_0/R_1

ℓ disjoint subset $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. Let $\mathcal{X}$ be the set of subscribers, or users, chosen to be revoked such that $\mathcal{X} \subseteq \mathcal{U}$. Choose $\mathcal{C} = \{x_1, \ldots, x_k\}$ a set of k subscribers such that $\mathcal{X} \cap \mathcal{C} = \emptyset$. Choose $k-1$ distinct elements $x_{k+1}, \ldots, x_{2k-1} \in_R \mathbb{Z}_q \setminus \mathcal{C}$ and random numbers $\beta_1, \ldots, \beta_k, \lambda, \mu, \psi_t, \omega_t \in_R \mathbb{Z}_q$, $k+1 \le t \le 2k-1$. There exists a unique polynomial $\alpha(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_{2k-1} x^{2k-1} \bmod q$ such that $g_1^{\alpha_0} = g_1^{\lambda} g_2^{\mu}$ and

$$(\alpha(x_1), \ldots, \alpha(x_{2k-1}))^T = (\beta_1, \ldots, \beta_{2k-1})^T$$
$$= (\alpha_0, \ldots, \alpha_0)^T + V(\alpha_1, \ldots, \alpha_{2k-1}) \bmod q$$
$$(\alpha_1, \ldots, \alpha_{2k-1})^T = V^{-1}(\beta_1 - \alpha_0, \ldots, \beta_{2k-1} - \alpha_0)^T \bmod q$$

where $g_1^{\beta_t} = g_1^{\psi_t} g_2^{\omega_t}$, $k+1 \le t \le 2k-1$, and V is the Vandermonde matrix. Let $(v_{m,1}, \ldots, v_{m,2k-1})$ be the mth row of V^{-1} matrix, then compute α_m, $1 \le m \le 2k-1$, as follows:

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

hence $g_1^{\alpha_m} = g_1^{v_{m,1}\beta_1 + \cdots + v_{m,2k-1}\beta_{2k-1}} / (g_1^{\lambda} g_2^{\mu})^{v_{m,1} + \cdots + v_{m,2k-1}}$. For computing users' personal key (x_j, i_j, d_j), we (a) choose a user $x_j \in \mathcal{U}_{i_j}$ where $1 \le j \le k$ and $i_j \in \{0, \ldots, \ell-1\}$, (b) define a set $I = \{i_j | 1 \le j \le k, x_j \in \mathcal{U}_{i_j}\}$, (c) randomly choose $\lambda_i, \mu_i \in_R \mathbb{Z}_q$ for each i, $0 \le i \le \ell-1$, and $\delta_{i_j} \in_R \mathbb{Z}_q$ for all elements $i_j \in I$. For each $i_j \in I$, there exists an element $\gamma_{i_j} \in_R \mathbb{Z}_q$ such that $\delta_{i_j} = b_{i_j} + \gamma_{i_j} - \alpha_{i_j \bmod 2k}$ and $g_1^{b_i} = g_1^{\lambda_i} g_2^{\mu_i}$, therefore we can compute last parameter of the key d_j as follows: $d_j = \alpha(x_j) + \delta_{i_j} x_j^{i_j \bmod 2k}$. Since the user's private key is $(u, i, f_i(u))$, we impose that $d_j = f_i(u)$, computing coefficients $a_0, \ldots, a_{2k-1}$ as follows. Consider the set $\{0, \ldots, 2k-1\} \setminus \{i_j \bmod 2k | i_j \in I\}$. It is possible to select k elements from this set which can be represented as $\theta_1, \ldots, \theta_k$. Compute $g_1^{\alpha'_{\theta_1}}, \ldots, g_1^{\alpha'_{\theta_k}}$ such that $g_1^{\sum_{\tau \in \{\theta_1, \ldots, \theta_k\}} \alpha'_\tau x_j^\tau} = g_1^{\gamma_{i_j} x_j^{i_j \bmod 2k}}$. In order to compute the public key e, it is necessary to calculate $g_1^{a_m}$ with $0 \le m \le 2k-1$ as follows,

$$g_1^{a_m} = \begin{cases} g_1^{\alpha_m} & m \notin \{\theta_1, \ldots, \theta_k\} \\ g_1^{\alpha_m} g_1^{\alpha'_m} & m \in \{\theta_1, \ldots, \theta_k\} \end{cases}$$

and the public key e will be $e = (g_1, g_1^{a_0}, \ldots, g_1^{a_{2k-1}}, g_1^{b_0}, \ldots, g_1^{b_{\ell-1}})$. Let $s \in_R G_q$ be the session key and $r \in_R \mathbb{Z}_q$ be a random number. For each i, $0 \le i \le \ell-1$, if $\mathcal{U}_i \cap \mathcal{X} = \emptyset$ or $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ set $B_i = 0$ or $B_i = 1$, otherwise, set $B_i = 1$. It is possible to identify eight compact headers H. Four are related to case $\exists \mathcal{U}_l$ such that $\mathcal{U}_l \cap \mathcal{X} \neq \emptyset$ and $\mathcal{U}_l \cap \mathcal{X} \neq \mathcal{U}_l$, $0 \le l \le \ell-1$, while the remaining when such $\mathcal{U}_l$ does not exist. In particular, if $\mathcal{U}_l$ exists then compact header H will be computed as follow,

1. if $B_i = 0$ for $i < l$, and $B_i = 0$ for $i > l$,
$$\left\langle \begin{matrix} g_1^r, g_1^{a_0 r}, \ldots, g_1^{a_{2k-1} r} \\ g_3, g_3^{a_0}, \ldots, g_3^{a_{2k-1}} \\ sg_1^{b_0 r}, \ldots, sg_1^{b_{l-1} r}, sg_3^{\lambda_l} g_4^{\mu_l}, sg_1^{b_{l+1} r}, \ldots, sg_1^{b_{\ell-1} r} \end{matrix} \right\rangle$$
2. if $B_i = 1$ for $i < l$, and $B_i = 0$ for $i > l$,
$$\left\langle \begin{matrix} g_1^r, g_1^{a_0 r}, \ldots, g_1^{a_{2k-1} r} \\ g_3, g_3^{a_0}, \ldots, g_3^{a_{2k-1}} \\ sg_3^{\lambda_0} g_4^{\mu_0}, \ldots, sg_3^{\lambda_{l-1}} g_4^{\mu_{l-1}}, sg_3^{\lambda_l} g_4^{\mu_l}, sg_1^{b_{l+1} r}, \ldots, sg_1^{b_{\ell-1} r} \end{matrix} \right\rangle$$
3. if $B_i = 0$ for $i < l$, and $B_i = 1$ for $i > l$,
$$\left\langle \begin{matrix} g_1^r, g_1^{a_0 r}, \ldots, g_1^{a_{2k-1} r} \\ g_3, g_3^{a_0}, \ldots, g_3^{a_{2k-1}} \\ sg_1^{b_0 r}, \ldots, sg_1^{b_{l-1} r}, sg_3^{\lambda_l} g_4^{\mu_l}, sg_3^{\lambda_{l+1}} g_4^{\mu_{l+1}}, \ldots, sg_3^{\lambda_{\ell-1}} g_4^{\mu_{\ell-1}} \end{matrix} \right\rangle$$
4. if $B_i = 1$ for $i < l$, and $B_i = 1$ for $i > l$,
$$\left\langle \begin{matrix} g_3, g_3^{a_0}, \ldots, g_3^{a_{2k-1}} \\ sg_3^{\lambda_0} g_4^{\mu_0}, \ldots, sg_3^{\lambda_{\ell-1}} g_4^{\mu_{\ell-1}} \end{matrix} \right\rangle$$

If such $\mathcal{U}_l$ does not exist, then $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$, or $\mathcal{U}_i \cap \mathcal{X} = \emptyset$, $1 \le i \le \ell-1$. Hence, compact header H will be computed as follow,

5. if $B_i = 0$ for $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ and $B_i = 1$ for $\mathcal{U}_i \cap \mathcal{X} = \emptyset$, then the header H is computed as in case 3
6. if $B_i = 1$ for $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ and $B_i = 1$ for $\mathcal{U}_i \cap \mathcal{X} = \emptyset$, then the header H is computed as in case 4
7. if $B_i = 0$ for $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ and $B_i = 0$ for $\mathcal{U}_i \cap \mathcal{X} = \emptyset$,

$$\left\langle \begin{matrix} g_1^r, g_1^{a_0 r}, \ldots, g_1^{a_{2k-1} r} \\ s g_1^{b_0 r}, \ldots, s g_1^{b_{\ell-1} r} \end{matrix} \right\rangle$$

8. if $B_i = 1$ for $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ and $B_i = 0$ for $\mathcal{U}_i \cap \mathcal{X} = \emptyset$,

$$\left\langle \begin{matrix} g_1^r, g_1^{a_0 r}, \ldots, g_1^{a_{2k-1} r} \\ g_3, g_3^{a_0}, \ldots, g_3^{a_{2k-1}} \\ s g_3^{\lambda_0} g_4^{\mu_0}, \ldots, s g_3^{\lambda_{l-1}} g_4^{\mu_{l-1}}, s g_1^{b_l r}, s g_1^{b_{l+1} r}, \ldots, s g_1^{b_{\ell-1} r} \end{matrix} \right\rangle$$

Knowing that

$$g_3^{a_j} = \begin{cases} g_3^{\alpha_j} & j \notin \{\theta_1, \ldots, \theta_k\} \\ g_3^{\alpha_j} g_3^{\alpha'_j} & j \in \{\theta_1, \ldots, \theta_k\} \end{cases}$$

and $g_3^{\sum_{\tau \in \{\theta_1, \ldots, \theta_k\}} \alpha'_\tau x_z^\tau} = \left(g_3^{\delta_{i_z}} g_3^{\alpha_{i_z \bmod 2k}} / g_3^{\lambda_{i_z}} g_4^{u_{i_z}} \right)^{x_z^{i_z \bmod 2k}}$ with $1 \le z \le k$, it is not hard to note that the compact headers described above behaves as normal encryption headers when (g_1, g_2, g_3, g_4)-tuple is a Diffie-Hellman tuple, and as tracing header when (g_1, g_2, g_3, g_4)-tuple is a random tuple in which traitors in $\mathcal{C}$ are not revoked. Public key e, private keys $(x_1, i_1, d_1), \ldots, (x_k, i_k, d_k)$ and compact header H input $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$. The p.p.t. algorithm decides weather header H is a valid input or an invalid one, and outputs "Diffie-Hellman tuple" or "Random tuple". Since $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$ is able to distinguish a valid input from an invalid one, $\mathcal{C}$ chosen arbitrarily such that $\mathcal{X} \cap \mathcal{C} = \emptyset$ and $|C| = k$, we constructed $\mathcal{M}^{DDH}$ using $\mathcal{D}_{\mathcal{C}}^{\text{dist}}$, hence $\mathcal{M}^{DDH}$ can solve the DDH problem. We can conclude that $\mathcal{D}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{DDH}$ for any $\mathcal{C}$ arbitrarily chosen. □

Lemma 4.2 (Secrecy of a Session Key in an Invalid Input)**.** *The computational complexity to compute the session key, for k subscribers revoked that received an invalid input, is as hard as DDH in G_q.*

Proof. The proof of Lemma 4.2 follows step by step the proof provided in (Matsushita and Imai, 2004, p. 269,270). However it is necessary to replace the following condition: - if $\mathcal{X} \cap \mathcal{U}_i = \emptyset$, then compute H as:

$$\hat{h}_i = g_3^r$$

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

with this: - if $\mathcal{X} \cap \mathcal{U}_i = \emptyset$, set $B_i = 0$ or 1 and compute H_i as follows:

$$\hat{h}_i = \begin{cases} g_3^r & (B_i = 0) \\ g_1^x g_3^y & (B_i = 1) \end{cases}$$

$$h_{i,j} = \begin{cases} g_3^{a_j r} & (j \neq i \bmod 2k, B_i = 0) \\ (g_1^x g_3^y)^{a_j} & (j \neq i \bmod 2k, B_i = 1) \\ s(g_3^{\lambda_i} g_4^{\mu_i})^r & (j = i \bmod 2k, B_i = 0) \\ s(g_1^{\lambda_i} g_2^{\mu_i})^x (g_3^{\lambda_i} g_4^{\mu_i})^y & (j = i \bmod 2k, B_i = 1) \end{cases}$$

□

Lemma 4.3 (Indistinguishability of a Suspect)**.** *Given a subscriber u_j, the computational complexity for a coalition of k subscribers to distinguish an invalid input in which the user u_j is not revoked — i.e. $\mathcal{X} = \{u_1, \ldots, u_{j-1}\}$ — from another one in which u_j is revoked — i.e. $\mathcal{X} = \{u_1, \ldots, u_j\}$ — is as hard as DDH in G_q.*

Proof. Let $\mathcal{C}$ be the set of k colluders. Let $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$ be a p.p.t. algorithm used by the coalition $\mathcal{C}$ to distinguish two invalid input, one in which a given user is revoked and the other one in which the user is not revoked. We prove that $\mathcal{A}_{\mathcal{C}}^{\text{dist}} \Leftrightarrow \mathcal{M}^{DDH}$ for any coalition $\mathcal{C}$ such that $|\mathcal{C}| = k$. Firstly, it is clear that $\mathcal{M}^{DDH} \Rightarrow \mathcal{A}_{\mathcal{C}}^{\text{dist}}$ for any coalition C. Secondly, we prove that $\mathcal{A}_{\mathcal{C}}^{\text{dist}} \Rightarrow \mathcal{M}^{DDH}$ for any $\mathcal{C}$ by constructing $\mathcal{M}^{DDH}$ using $\mathcal{A}_{\mathcal{C}}^{\text{dist}}$ as a subroutine. The algorithm $\mathcal{M}^{DDH}$ takes in input a challenge tuple (g_1, g_2, g_3, g_4) and outputs "Diffie-Hellman tuple" or "Random tuple". The construction of the algorithm is as follows. Split the set of subscribers $\mathcal{U}$ into ℓ disjoint subset $\mathcal{U}_0, \ldots, \mathcal{U}_{\ell-1}$. Let $\mathcal{X}$ a set of revoked subscribers. Choose k users which form the set $\mathcal{C}$ and choose a user u_j which has to be a subscriber who does not belong to the set of colluders, i.e., $u_j \in_{\mathrm{R}} \mathcal{U} \setminus \mathcal{C}$. We suppose that user $u_j \in \mathcal{U}_t$. We also suppose that for $i = 0, \ldots, t-1$ sets $\mathcal{U}_i$ are such that $\mathcal{U}_i \cap \mathcal{X} = \mathcal{U}_i$ and for $i = t+1, \ldots, \ell-1$ sets $\mathcal{U}_i$ are such that $\mathcal{U}_i \cap \mathcal{X} = \emptyset$. According to its relation with the set of revoked users, the set $\mathcal{U}_t$ can be such that:

1. $\mathcal{U}_t \cap \mathcal{X} \neq \mathcal{U}_t$ and $\mathcal{U}_t \cap \mathcal{X} \neq \emptyset$. This case has to be considered both when $u_j \in \mathcal{X}$ and when $u_j \notin \mathcal{X}$.
2. $\mathcal{U}_t \cap \mathcal{X} = \emptyset$ when $u_j \notin \mathcal{X}$ and $\mathcal{U}_t \cap \mathcal{X} = \{u_j\}$ when $u_j \in \mathcal{X}$.
3. $\mathcal{U}_t \cap \mathcal{X} = \mathcal{U}_t \setminus \{u_j\}$ when $u_j \notin \mathcal{X}$ and $\mathcal{U}_t \cap \mathcal{X} = \mathcal{U}_t$ when $u_j \in \mathcal{X}$.

Now, choose $\mathcal{C} = \{x_1, \ldots, x_k\}$ a set of k subscribers. Consider the user $x_j \in \mathcal{U}_{i_j}$ and compute its personal key (x_j, i_j, d_j) and public key $e =$

$(g_1, g_1^{a_0}, \ldots, g_1^{a_{2k-1}}, g_1^{b_0}, \ldots, g_1^{\ell-1})$ using the same procedure of Lemma 4.1. Construct the header $H = (H_0, \ldots, H_{\ell-1})$ executing the following procedure for $0 \leq i \leq \ell - 1$. The single header H_i for the group $\mathcal{U}_i$ will be $H_i = (\hat{h}_i, h_{i,0}, \ldots, h_{i,2k-1})$ where the single elements in H_i are computed as follows:

- if $\mathcal{X} \cap \mathcal{U}_i = \emptyset$, set $B_i = 0$ or 1:

$$\hat{h}_i = \begin{cases} g_3^r & (B_i = 0) \\ g_1^x g_3^y & (B_i = 1) \end{cases}$$

$$h_{i,j} = \begin{cases} g_3^{a_j r} & (j \neq i \bmod 2k, B_i = 0) \\ (g_1^x g_3^y)^{a_j} & (j \neq i \bmod 2k, B_i = 1) \\ s(g_3^{\lambda_i} g_4^{\mu_i})^r & (j = i \bmod 2k, B_i = 0) \\ s g_3^{b_i y} (g_1^{\lambda_i} g_2^{\mu_i})^x & (j = i \bmod 2k, B_i = 1) \end{cases}$$

- if $\mathcal{X} \cap \mathcal{U}_i = \mathcal{U}_i$, set $B_i = 0$ or 1 and compute H_i as follows, selecting each time a random $z_i \in_R \mathbb{Z}_q$.

$$\hat{h}_i = \begin{cases} g_3^r & (B_i = 0) \\ g_1^x g_3^y & (B_i = 1) \end{cases}$$

$$h_{i,j} = \begin{cases} h'_{i,j} & (j \neq i \bmod 2k) \\ g_1^{z_i} & (j = i \bmod 2k) \end{cases}$$

where $h'_{i,j}$ is computed in two different ways depending on the existence of a subset $\mathcal{U}_t$ with $0 \leq t \leq \ell - 1$ such that $\mathcal{X} \cap \mathcal{U}_t \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_t \neq \mathcal{U}_t$. If such a set exists 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:

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

- if $\mathcal{X} \cap \mathcal{U}_i \neq \emptyset$ and $\mathcal{X} \cap \mathcal{U}_i \neq \mathcal{U}_i$, the header H_t will be constructed as follows. First, construct a polynomial $C(x) = \sum_{j=0}^{2k-1} c_j x^j$ such that for $u \in \mathcal{U}$ with $u \neq u_j$, $C(u) \equiv 0 \bmod q$ if and only if it holds that $u \in (\mathcal{U}_t \setminus \mathcal{X})$. We also suppose that for the user u_j it holds that $C(u_j) \equiv 0 \bmod q$. Then:

$$\hat{h}_i = g_1^x g_3^y$$

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

Note that if (g_1, g_2, g_3, g_4) is a Diffie-Hellman tuple, the subscriber u_j is not revoked in the header H, otherwise, if (g_1, g_2, g_3, g_4) is a Random tuple then the subscriber u_j is revoked. Run the algorithm $\mathcal{A}_C^{\text{dist}}$, by giving in input to it $u, H, e, (x_1, i_1, d_1), \ldots, (x_k, i_k, d_k)$. This algorithm is able to distinguish invalid input in which the subscriber u_j is not revoked from an invalid input in which u_j is revoked. Since we have constructed $\mathcal{M}^{DDH}$ using $\mathcal{A}_C^{\text{dist}}$ as a subroutine, we can conclude that $\mathcal{M}^{DDH}$ can solve the DDH problem. □

Theorem 4.4. *Given the traitor tracing scheme described in Section 3 and a pirate decoder constructed by a coalition of k traitors, a tracer is able to identify at least one of the traitors with non-negligible probability.*

REFERENCES

Boneh, D. (1998). The Decision Diffie-Hellman Problem. In *ANTS*, volume 1423 of *LNCS*, pages 48–63.

Boneh, D. and Franklin, M. K. (1999). An efficient public key traitor tracing scheme. In *CRYPTO*, volume 1666 of *LNCS*, pages 338–353.

Chor, B., Fiat, A., and Naor, M. (1994). Tracing traitors. In *CRYPTO*, volume 839 of *LNCS*, pages 257–270.

Chor, B., Fiat, A., Naor, M., and Pinkas, B. (2000). Tracing traitors. In *IEEE Transactions on Information Theory*, volume 46, pages 893–910.

Dodis, Y. and Fazio, N. (2002). Public key broadcast encryption for stateless receivers. In *DRM Workshop*, volume 2696 of *LNCS*, pages 61–80.

Dwork, C., Lotspiech, J. B., and Naor, M. (1996). Digital signets: Self-enforcing protection of digital information. In *STOC*, pages 489–498.

Kiayias, A. and Pehlivanoglu, S. (2009). On the security of a public-key traitor tracing scheme with sublinear ciphertext size. In *DRM Workshop*, pages 1–10. ACM.

Kiayias, A. and Pehlivanoglu, S. (2011). Attacking traitor tracing schemes using history recording and abrupt decoders. In *ISC*, volume 7001 of *LNCS*, pages 17–31.

Kiayias, A. and Yung, M. (2001). On crafty pirates and foxy tracers. In *DRM Workshop*, volume 2320 of *LCNS*.

Kurosawa, K. and Desmedt, Y. (1998). Optimum traitor tracing and asymmetric schemes. In *EUROCRYPT*, pages 145–157.

Kurosawa, K. and Yoshida, T. (2002). Linear code implies public-key traitor tracing. In *Public Key Cryptography*, volume 2274 of *LNCS*, pages 172–187.

Matsushita, T. (2002). A flexibly revocable key-distribution scheme for efficient black-box tracing. In *ICICS*, volume 2513 of *LNCS*, pages 197–208.

Matsushita, T. and Imai, H. (2004). A public-key black-box traitor tracing scheme with sublinear ciphertext size against self-defensive pirates. In *ASIACRYPT*, volume 3329 of *LNCS*, pages 260–275.

Matsushita, T. and Imai, H. (2006). Hierarchical key assignment for black-box tracing with efficient ciphertext size. In *ICICS*, volume 4307 of *LNCS*, pages 92–111.

Naor, M. and Pinkas, B. (1998). Threshold traitor tracing. In *CRYPTO*, volume 1462 of *LNCS*, pages 502–517.

Naor, M. and Pinkas, B. (2010). Efficient trace and revoke schemes. *Int. J. Inf. Sec.*, 9(6):411–424.