NEW SECRET KEY TRAITOR TRACING SCHEME WITH DISPUTE SETTLEMENT FROM BILINEAR MAPS

KUO-YU TSAI, TZONG-CHEM WU AND CHIEN-LUNG HSU

1Taiwan Information Security Center
2Department of Information Management
National Taiwan University of Science and Technology
No. 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan
nicklas@twisc.org
3Department of Electrical Engineering
Chang Gung University
No. 259, Wen-Hwa 1st Rd., Kwei-Shan, Tao-Yuan 333, Taiwan

Received November 2009; revised May 2010

Abstract. We propose a new secret key traitor tracing scheme using bilinear maps, in which the size of the enabling-block in a broadcast message is independent of the number of subscribers. The proposed traitor tracing scheme can identify malicious subscribers (also called traitors) from a pirate decoder, which is collusively made by \( k \) or fewer traitors. Further, any malicious subscriber cannot falsify an unintended content for framing the broadcast center and cause all other personal keys to be exposed. We also give formal analyses to discuss the security of the proposed scheme in the random oracle model.

Keywords: Traitor tracing, Dispute settlement, Bilinear map

1. Introduction. Broadcast encryption [1] is one of the mechanisms to securely tackle the distribution of digital content to a specific set of authorized subscribers [2, 3, 4, 5, 6, 7]. In various well-known practical and useful applications of broadcast encryption, such as pay-per-view or subscription television broadcasts, online database publicly accessible on the Internet, or distribution of multimedia (e.g., files in MP3 or JPEG), a subscriber is in possession of a decoder that allows them to access the broadcast. The roles of a broadcast encryption mechanism can be categorized as a broadcast center and a set of subscribers each equipped with a decoder. Initially, the broadcast center assigns an authorized key, stored in a decoder, for each subscriber. Thereafter, the broadcast center encrypts an intended content with an encryption key and broadcasts the encrypted content. Only the authorized subscribers can reconstruct the encryption key with their authorized keys, and further, they recover the intended content with their encryption keys. Basically, a broadcast message is divided into two parts: an enabling-block and a cipher-block. The cipher-block is the ciphertext form of an intended content, which is encrypted by using the encryption key. The enabling-block contains the public information that is used by a decoder to reconstruct the encryption key for obtaining the intended content. Upon receiving the broadcast message, each authorized subscriber can use his/her authorized key to reconstruct the encryption key from the enabling-block and then use it to decrypt the cipher-block to obtain the intended content.

Consider such scenario in which some malicious subscribers (so-called traitors) may collude to create a pirate decoder with their authorized keys or copy intended content to non-subscribers in the above applications. Traitor tracing [8] is one of the important mechanisms to resist the piracy. Chor et al. [8] introduced the concept of a traitor tracing
scheme to discourage malicious subscribers from creating a pirate decoder. In Chor et al.'s scheme, each subscriber is assigned a distinct personal key for decrypting the broadcast messages. Some traitors may try to mix parts of their personal keys to create a pirate decoder. In case of detecting and confiscating any pirate decoder, one or more personal keys will be traced. According to traced personal keys, at least one traitor will be identified. A traitor tracing scheme is called $k$-resilient traceability if at least one traitor can be correctly identified when up to $k$ traitors collude to make a pirate decoder. Since then, several traitor tracing schemes have been developed [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. Among these schemes, some of them are secret key traitor tracing schemes, e.g., in [14, 19, 20, 23, 24, 26] and some of them are public key traitor tracing schemes, e.g., in [9, 12, 17, 18, 21, 25]. The former schemes allow the broadcaster to use a secret key to broadcast messages and hence are applicable to the case that only the broadcast center can broadcast messages to the set of subscribers. The latter schemes allow the broadcaster to use a public key to broadcast messages and hence are applicable to the case that anyone could serve as the broadcaster. In general, public key traitor tracing schemes are more costly than the secret key ones in storage requirements and bandwidth. However, to earn cost effectiveness and to achieve security robustness are two major design principles regarding to the key management issues for traitor tracing. To design a traitor tracing scheme, a flexible and efficient approach to key management is one of the major factors. Besides, it should provide a robust traitor detection mechanism against piracy without harming the innocent subscribers.

According to Boneh et al.'s discussions [11], traitor tracing can be classified into two categories: trace and trace-and-revoke. The former focuses on the capabilities of tracing a traitor. The latter provides not only tracing capabilities but also revoking capabilities. As compared with these two categories, the former requires shorter key length and lower message transmission. The former does not provide a technical solution for revoking but it can solve the revoking problem by some non-technical methods, such as seizing the decoder, punishment, etc.

Following up the above design principles, Mitsunari et al. [19] proposed a secret key traitor tracing scheme using bilinear maps, in which the size of the enabling-block in a broadcast message is independent of the number of subscribers. However, Tô et al. [25] have shown that Mitsunari et al.'s scheme cannot withstand the so-called linear attack. That is, any $k$ out of $n$ subscribers acting as colluding traitors can always find a linear combination of their personal keys to generate a pirate personal key. The pirate personal key is different from their pre-assigned personal keys on avoiding traitor tracing. Tô et al. also proposed three public key traitor tracing schemes and claimed that their proposed schemes are resistant to the linear attack. However, Chabanne et al. [12] pointed out that Tô et al.'s schemes [25] are still insecure. Chabanne et al. further introduced the property of public traceability, which means that the tracing capability can be delegated to a dishonest party and publicly computable by the delegated party.

In this paper, inspired from the original system model proposed by Mitsunari et al. [19] and the definition of collusion-resistance presented by Boneh et al. [11], we shall present a new secret key traitor tracing scheme with dispute settlement using bilinear maps, in which the size of the enabling-block in a broadcast message is independent of the number of subscribers. In the proposed scheme, it is able to identify malicious traitors from a pirate decoder which is collusively made by $k$ or fewer traitors, and no innocent subscriber is convicted, where $k$ is the number of traitors. Furthermore, any malicious subscriber neither has the ability to falsify an unintended content for framing the broadcast center nor causes all other personal keys to be exposed. We also prove the security of the proposed scheme in the random oracle model [28].
The paper is organized as follows: In Section 2, we give some definitions of the computational problems and a secret key traitor tracing scheme with dispute settlement. We also discuss a formal security model in this section. Our scheme is given in Section 3. In Section 4, we discuss the security and efficiency analyses. Finally, conclusions are given in Section 5.

2. Preliminaries. In this section, we first introduce the properties of a bilinear map, and then give definitions of the one-way hash function (OHF) [29] assumption, the weak Diffie-Hellman problem (WDHP) [30], and the collusion attack algorithm with $k$-traitors ($k$-CAA) [26, 31]. Finally, we give a formal definition and a security model for a secret key traitor tracing scheme with dispute settlement.

2.1. Bilinear maps. Let $G_1$ and $G_2$ be two cyclic groups of prime order $q$, where $G_1$ is an additive group and $G_2$ is a multiplicative group. We use a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ in our system, having the following properties:

1. For any $P, Q, R \in G_1$ and $a, b \in \mathbb{Z}_q$, we have $\hat{e}(aP + bQ, R) = \hat{e}(P, R)^a \cdot \hat{e}(Q, R)^b$ and $\hat{e}(R, aP + bQ) = \hat{e}(R, P)^a \cdot \hat{e}(Q, R)^b$.
2. Non-degenerate: There exists a point $P \in G_1$ such that $\hat{e}(P, P) \neq 1$.
3. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G_1$.

2.2. Computational assumptions.

Definition 2.1 (OHF assumption). A secure one-way hash function $H$ operates on an arbitrary length input $m$ and outputs the fixed length $y = H(m)$ such that (i) Given $m$, it is easy to compute $y = H(m)$. (ii) Given $y$, it is computationally infeasible to derive $m$ such that $y = H(m)$. (iii) Given $m$, it is computationally infeasible to find $m'$ ($m' \neq m$) such that $H(m') = H(m)$. (iv) It is computationally infeasible to find $m$ and $m'$ such that $H(m) = H(m')$.

Definition 2.2 (WDHP). Let $G_1$ be an additive group of the prime order $q$. Given $(P, aP, W)$ for $P, W \in G_1$ and $a \in \mathbb{Z}_q^*$, it is computationally infeasible to compute $aW$.

Definition 2.3 (k-CAA). Let $G_1$ be an additive group of the prime order $q$. Given $(P, yP, \frac{1}{h_1+y}P, \frac{1}{h_2+y}P, \ldots, \frac{1}{h_k+y}P)$ for $P \in G_1$, $y \in \mathbb{Z}_q^*$, and $h_1, h_2, \ldots, h_k \in \mathbb{Z}_q^*$, it is computationally infeasible to derive $\frac{1}{h+y}P$ for some $h \notin h_1, h_2, \ldots, h_k$.

2.3. Formal definitions and security model. A secret key traitor tracing scheme with dispute settlement is comprised of the following algorithms:

- **Setup:** takes a security parameter $\lambda$ and returns system parameters and a secret key $SK$.
- **Extract:** takes system parameters, a secret key $SK$ and an arbitrary identity. This algorithm is used to generate a personal key $K_i$ that is given to subscriber $i$ associated with the identity $u_i$.
- **Encrypt:** takes a secret key $SK$, a random encryption key $s$ and a message $M$. This algorithm is used to encrypt $M$ to generate a broadcast message, consisting of an enabling-block $EBK$ and a cipher-block $CBK$. Note that there exists coherent relationship between $M$ and its corresponding $EBK$.
- **Decrypt:** takes a broadcast message $(EBK, CBK)$ and a personal key $K_i$ for $u_i$. If the coherent relationship between $M$ and the corresponding $EBK$ is correct, it outputs a random encryption key $s$ and a message $M$; otherwise $\perp$.
- **Trace:** takes a secret key $SK$ and a pirate personal key $\hat{K}$ and outputs a set of malicious subscribers (also called traitors).
**Settlement:** takes a secret key $SK$ and a broadcast message ($EBK$, $CBK$) and outputs True or False.

Security requirements of traitor tracing schemes with dispute settlement can be defined as collusion-resistance, falsification-free and dispute-settlement, and exposure-resistance below.

1. **Collusion-resistance:** Let $n$ and $k$ denote respectively as the total number of subscribers and the number of malicious subscribers, where $k \leq n$. According to the property of collusion-resistance defined by Boneh et al. [11], a traitor tracing scheme is said to be $k$-collusion resistant if the tracing algorithm is able to identify malicious traitors from a pirate decoder which is collusively made by $k$ or fewer traitors. The notion of collusion-resistance against $k$ traitors is defined by Game 1 played between an adversary $A$ and a challenger $C$.

**Game 1**

**Start:** For a given security parameter $\lambda$, the challenger $C$ runs **Setup** to generate system parameters and a secret key $SK$. Then, $C$ gives the adversary $A$ the system parameters and keeps $SK$ secret.

**Query:** $A$ issues a number of personal key extract queries with respect to identities $\{u_1, u_2, \ldots, u_k\}$ to $C$, where $k < n$. The challenger $C$ responds personal keys $K_i$’s $(1 \leq i \leq k)$ to $A$ by running **Extract**.

**Collusion:** With the knowledge of $\{K_1, K_2, \ldots, K_k\}$, $A$ decides a set of the colluding personal keys to form a pirate personal key $\hat{K}$.

**Trace:** The challenger $C$ runs **Trace** to obtain a set $S$.

We say that $A$ wins the game if both of the following conditions hold.

- **C1:** The pirate personal key is said to be valid if it can decrypt all broadcast message with overwhelming probability $\varepsilon$, where $\varepsilon = 1/f(\lambda)$ for some polynomial $f$.
- **C2:** The detected set $S$ is either empty, or is not the subset of $\{K_1, K_2, \ldots, K_k\}$.

2. **Falsification-free and dispute-settlement:** This game ensures that no one can falsify an illegal broadcast message to set a trap for a broadcast center. We define falsification-free and dispute-settlement by Game 2 played between $A$ and $C$ below.

**Game 2**

**Start:** For a given security parameter $\lambda$, the challenger $C$ runs **Setup** to generate system parameters and a secret key $SK$. Then, $C$ gives the adversary $A$ the system parameters and keeps $SK$ secret.

**Phase 1:** $A$ issues a number of queries to $C$. Three types of queries are as follows:

- **Extract query:** $A$ submits an identity to $C$. Then, $C$ responds a personal key by running **Extract**.
- **Encrypt query:** $A$ submits a message $M$ to $C$. Then, $C$ responds a broadcast message ($EBK$, $CBK$) by running **Encrypt**.
- **Decrypt query:** $A$ submits a broadcast message ($EBK$, $CBK$) to $C$. Then, $C$ responds a random encryption key and a message $M$ by running **Decrypt**.

**Challenge:** Once Phase 1 is finished, $A$ decides a target message $M^*$ different from the messages $M$’s as in Phase 1.

**Phase 2:** $A$ issues more queries as in Phase 1. It is not allowed to submit an **Encrypt/Decrypt** query for the target message $M^*$.

**Output:** Finally, $A$ outputs a broadcast message ($EBK^*$, $CBK^*$) for the target message $M^*$. If the coherent relationship between $M^*$ and the corresponding $EBK^*$ is correct, $A$ wins the game. That is, $A$ successfully forges an illegal broadcast message for $M^*$. Otherwise, falsification-free requirement is achieved.
A secret key traitor tracing scheme with dispute settlement is falsification-free and dispute-settlement if the successful probability of any polynomially bounded adversary in the above game is negligible. That is, a broadcast message for the target message cannot be forged by any polynomially bounded adversary. In case of a dispute, the broadcast message can be used to settle the dispute.

3. Exposure-resistance: According to evil manufactures problems [15], a malicious subscriber may abuse the key assigned to him and cause all other personal keys to be exposed. We define exposure-resistance by Game 3 played between $A$ and $C$.

**Game 3**

- **Start**: For a given security parameter $\lambda$, the challenger $C$ runs Setup to generate system parameters and a secret key $SK$. Then, $C$ gives the adversary $A$ the system parameters and keeps $SK$ secret.

- **Phase 1**: $A$ issues a sequence of Extract, Encrypt and Decrypt queries to $C$. These are answered by $C$ as in Game 2.

- **Challenge**: Once Phase 1 is finished, $A$ decides a target identity different from the identities as in Phase 1.

- **Phase 2**: $A$ issues more queries as in Phase 1. It is not allowed to submit an Extract query for the target identity.

- **Output**: Finally, $A$ outputs a personal key for the target identity. If the personal key is valid, $A$ wins the game. That is, $A$ successfully causes the other personal key to be exposed. Otherwise, exposure-resistance requirement is achieved.

3. The Proposed Scheme. For simplicity, the following symbols are defined in advance:

- $p, q$: are prime numbers and $q < p$.
- $G_1, G_2$: are two cyclic groups of order $q$, where $G_1$ is an additive group and $G_2$ is a multiplicative group.
- $\hat{e}$: is a bilinear map defined as $\hat{e} : G_1 \times G_1 \rightarrow G_2$.
- $H_0, H_1$: are secure one-way hash functions, where $H_0 : G_1 \rightarrow Z_q^*$ and $H_1 : \{0,1\}^* \rightarrow G_1$.

Elaborating on Mitsunari et al.’s traitor tracing scheme [19] and Boneh et al.’s short signature scheme [32], we apply short signature to provide evidence for dispute settlement in a secret key traitor tracing scheme. The proposed scheme consists of five phases: Initialization, Distributing, Decoding, Traitor Detection and Dispute Settlement. The tasks performed in Initialization, Distributing, Decoding and Traitor Detection are like those in the Mitsunari et al.’s scheme [19]. Additionally, in Initialization and Dispute Settlement phases, an adjudicator or a trusted third party (or TTP for short) is required in the case that a malicious subscriber has falsified an unintended content for framing the broadcast center.

Denote $BC$ as the identity of broadcast center and $U$ as the set of subscribers in the broadcast system. Without loss of generality, let $u_i \in Z_q^*$ be the identity associated to the $i$-th subscriber in $U$. Each subscriber possesses a decoder designated in hardware box which stores the owner’s identity, personal key, and public functions. The decoder is equipped with the computation ability to reconstruct an encryption key from an enabling-block (or $EBK$ for short) and then to obtain an intended content from the cipher-block (or $CBK$ for short). We describe the Initialization, Distributing, Decoding, Traitor Detection and Dispute Settlement phases in detail as follows.

**Initialization Phase** – First, TTP runs Setup to generate system parameters $(p, q, G_1, G_2, H_0, H_1)$ and to choose a secret key $SK = (a, P)$ for BC, where $a$ is a random number in $Z_q^*$ and $P$ is an element in $G_1$. For the purpose of dispute settlement, TTP computes authentic information for a broadcast center by computing $\text{Authcode} = \hat{e}(a, P)$. Then, TTP distributes the system parameters and the secret key $SK$ to all subscribers.
After that, TTP stores \((u_i, K_i, H_0, H_1)\) in a decoder and delivers the decoder to the subscriber \(i\). Note that TTP serves as a key generation center in this phase and as an adjudicator in the subsequent Dispute Settlement phase.

**Distributing Phase** – Let \(M\) be an intended content to be broadcasted by BC at a timestamp \(t\). To do that, the broadcast center first chooses a random encryption key \(s \in G_2\) and then runs **Encrypt** to construct an enabling-block \(EBK = [V; R; P_1; t]\) and cipher-block \(CBK = [E_s(M)]\), where

\[
P_1 = H_1(M\|t),
\]

\[
V = s \cdot e(tP, P_1),
\]

and

\[
R = aP_1.
\]

Afterwards, BC broadcasts \((EBK, CBK)\) to all subscribers.

**Decoding Phase** – Upon receiving \((EBK, CBK)\), a decoder of an authorized subscriber \(i\) triggers **Decrypt** to perform the following tasks:

1. Reconstruct the decryption key \(s\) from \(EBK\) with the personal key \(K_i\) by computing:

\[
s = V \cdot e(tK_i, (u_i + H_0(Q_i))P_1 + R)^{-1}.
\]

2. Use \(s\) to obtain \(M = D_s(CBK)\) by decrypting \(CBK\).
3. Verify if \(P_1 = H_1(M\|t)\). If the equality holds, then outputs \(s\) and \(M\); otherwise \(\perp\).

**Traitor Detection Phase** – In traitor tracing schemes, the trace algorithm works based on the assumption that the suspected pirate decoders are confiscated before performing the trace algorithm [8]. Such an assumption is feasible in real world. In practice, it can be done by some non-technical methods, such as some intellectual property protection mechanisms. Anyone can report a pirate of music, software or movies to the authorities and the authorities can seize the pirate objects. Our proposed scheme focuses on the tracing technique based on the same assumption. Suppose that a confiscated pirate personal key from a pirate decoder is collusively made by \(k\) traitors. BC first retrieves a set of suspected personal keys from the pirate decoder, BC runs **Trace** to identify possible traitor(s) and then verifies Equations (1) and (2) by the secret key \(SK = (a, P)\). If the two equations hold, then the suspected personal key(s) used to form the pirate personal key is identified. The subscriber(s) with the identified personal key(s) are traitor(s).

**Dispute Settlement Phase** – Suppose that a malicious subscriber has falsified an unintended content \(M'\) at a timestamp \(t'\) for framing BC. For settling such a dispute, TTP requires BC to present his/her secret key \((a, P)\), a suspected broadcast message consisting of \(EBK = [V, R, P_1, t']\) and \(CBK = [E_s(M')]\), and the intended personal key via a secure channel. TTP first verifies the correctness of \((a, P)\) using **Authcode** stored in the directory. Then, TTP runs **Settlement** to compute \(P_1 = H_1(M'\|t')\) and verify Equation (5). If Equation (5) holds, then TTP concludes that BC cannot deny broadcasting \(M'\). Otherwise, \(M'\) is not broadcasted by BC.

4. **Security and Efficiency Analysis.** In Section 4.1, we will show that the proposed scheme achieves collusion-resistance, falsification-free and dispute-settlement, and exposure-resistance. We brief the efficiency of the proposed scheme in Section 4.2.
4.1. Security proof. The security of the proposed scheme is based on the intractability of the one-way hash function (OHF) assumption [29], the weak Diffie-Hellman problem (WDHP) [30] and the collusion attack algorithm with k-traitor (k-CAA) [26, 31].

Correctness. The proposed secret key traitor tracing with dispute settlement is correct if a decoder for an authorized subscriber i can correctly reconstruct an encryption key s from EBK by Equation (6), which is identical to that chosen by BC.

Proof: From Equations (2), (4) and (5) we have

\[
Pr[A\text{ wins Game } 1] = \left( 1 - \frac{1}{q^n} + \frac{1}{q^n \varepsilon_1} \right) \leq \varepsilon
\]

Since \( \varepsilon \) happens with non-negligible probability, we can see that the probability \( \varepsilon_1 \) is also with non-negligible advantage. This contradicts the fact that solves the OHF assumption is computationally infeasible.

Theorem 4.2. The proposed scheme is falsification-free if there is no polynomial-time algorithm that solves the WDHP with non-negligible probability.

Proof: Let \( A \) be a polynomial-time algorithm that outputs an existential forgery with probability \( \varepsilon \). Let \( q_{H_0}, q_{H_1}, q_{ext}, q_{enc} \) and \( q_{dec} \) denote the number of queries to \( H_0, H_1, \)
Extract, Encrypt and Decrypt oracles, respectively. In the following, we demonstrate how to use the algorithm $A$ to construct a polynomial-time algorithm $\beta_1$ that outputs the solution of the WDHP with probability $\varepsilon_2$.

**Start:** For a given security parameter $\lambda$, the algorithm $\beta_1$ runs **Setup** to generate system parameters $(p, q, G_1, G_2, H_0, H_1)$ and a secret key $SK = (a, P)$. Let $(P_1, aP_1, W)$ be an instance of the WDHP, where $P_1, W \in G_1$. Then, $\beta_1$ gives the algorithm $A$ the system parameters and keeps $(a, P)$ secret. Here, $H_0$ and $H_1$ are random oracles controlled by $\beta_1$.

**Phase 1:** $A$ issues a number of queries to $\beta_1$. Types of queries are described as follows:

**Q1:** $H_0$-query: $A$ can query a random oracle $H_0$ at any time. $\beta_1$ maintains a list list$_0$ to respond to $A$’s queries. Upon receiving a $H_0$-query for some $Q_i$, the algorithm $\beta_1$ performs the following:
1. Check list$_0$. If there exists $(Q_i, r_i)$ in list$_0$, it returns $r_i$ and terminates the simulation.
2. Choose a random number $r_i \in Z_q^*$. If there exists $r_i$ in list$_0$, repeats Step 2.
3. Put $(Q_i, r_i)$ in list$_0$.

**Q2:** $H_1$-query: $A$ can also query a random oracle $H_1$ at any time. $\beta_1$ maintains another list list$_1$ to respond to $A$’s queries. Upon receiving a $H_1$-query for some message $M_i$ and $t_i$, the algorithm $\beta_1$ performs the following:
1. Check list$_1$. If there exists $(M_i, t_i, H_1(M_i||t_i), c_i, b_i)$ in list$_1$, it returns $H_1(M_i||t_i)$ and terminates.
2. Generate a random number $c_i \in Z_q^*$. If $c_i = 0$
   - Compute and return $H_1(M_i||t_i) = c_i P_1 + W$
   else
   - Compute and return $H_1(M_i||t_i) = c_i P_1$
4. Put $(M_i, t_i, H_1(M_i||t_i), c_i, b_i)$ in list$_1$.

**Q3:** Extract query: $A$ submits an identity $u_i$ to $\beta_1$. The algorithm $\beta_1$ responds the personal key $K_i = (u_i + H_0(Q_i) + a)^{-1} P$ for $u_i$ by running **Extract**.

**Q4:** Encrypt query: $A$ submits a message $M_i$ and $t_i$ to $\beta_1$. Then, the algorithm $\beta_1$ performs the following:
1. Run the above algorithm for simulating $H_1$-oracle to get $(M_i, t_i, H_1(M_i||t_i), c_i, b_i)$.
2. If $b_i = 0$
   - Abort the simulation
   else
   - Compute $R_i = c_i a P$
3. Choose an encryption key $s_i$ and compute $V_i = s_i \cdot \hat{e}(tP, c_i P_1)$
4. Return $(s_i, \hat{e}(tP, c_i P_1), c_i a P_1, c_i P_1, t_i)$ to $A$.

**Q5:** Decrypt query: $A$ submits a broadcast message $(EBK_i, CBK_i)$ to $\beta_1$, where $EBK_i = (s_i, \hat{e}(tP, c_i P_1), c_i a P_1, c_i P_1, t_i)$ and $CBK_i = E_s(M_i)$. The algorithm $\beta_1$ recovers a random encryption key $s_i$ and a message $M_i$ by running **Decrypt**. If $c_i P_1 = H_1(M_i||t_i)$, $\beta_1$ returns $s_i$ and $M_i$ to $A$. Otherwise, it outputs $\perp$.

**Challenge:** $A$ decides a target message $M^*$ which is different from the messages $M_i$’s
as in Phase 1.

**Phase 2**: A issues more queries as in Phase 1. It is not allowed to submit an Encrypt/Decrypt query for the target message $M^*$.

**Output**: Suppose that A successfully outputs a broadcast message $(EBK^*, CBK^*)$ for $M^*$ with probability $\varepsilon$. If $(EBK^*, CBK^*)$ is a valid broadcast message, there exists coherent relationship between $M^*$ and the corresponding $EBK^*$ such that $H_1(M^*||t^*) = e^*P_1 + W$ and $R^* = aH_1(M^*||t^*)$. Substituting $H_1(M^*||t^*)$ into $R^*$, we can have $R^* = ac^*P_1 + aW$, which implies $aW = ac^*P_1 - R^*$. Hence, $\beta_1$ can carefully derive the solution $aW$ of the WDHP.

During the simulation, $\beta_1$ does not abort the game with the probability $\zeta^{\text{enc}}$. At the end of simulation, A successfully outputs the broadcast message for the target message $M^*$ with the probability $\varepsilon$. Finally, $\beta_1$ can transform A’s forgery into the solution of the WDHP with the probability at least $(1 - \zeta)$.

We can conclude that $\beta_1$ can solve the WDHP with the probability $\varepsilon_2 \geq \zeta^{\text{enc}}(1 - \zeta)\varepsilon$.

**Theorem 4.3**. The proposed scheme is exposure-resistant if there is no polynomial-time algorithm that solves the $k$-CAA with non-negligible probability.

**Proof**: We suppose that $H_0$ and $H_0$ are random oracles, and there exists a polynomial-time algorithm A that wins the exposure-resistance game with non-negligible probability $\varepsilon$. Let $q_{H_0}$, $q_{H_1}$, $q_{\text{ext}}$, $q_{\text{enc}}$ and $q_{\text{dec}}$ denote the numbers of queries to $H_0$, $H_1$, Extract, Encrypt and Decrypt oracles, respectively. In the following, we will demonstrate how to use A to construct a polynomial-time algorithm $\beta_2$ that solves the $k$-CAA. Let $(P, yP, \frac{1}{h_1+y}P, \frac{1}{h_2+y}P, \ldots, \frac{1}{h_k+y}P)$ be a random instance of the $k$-CAA. The goal of $\beta_2$ is to derive $\frac{1}{h_1+y}P$ for some $h^* \neq h_1, h_2, \ldots, h_k$.

**Start**: For a given security parameter $\lambda$, the algorithm $\beta_2$ runs Setup to generate system parameters $(p, q, G_1, G_2, H_0, H_1)$ and a secret key $SK = (a, P)$. Then, $\beta_2$ gives the algorithm A the system parameters and keeps $(a, P)$ secret.

**Phase 1**: A issues a number of queries as in Theorem 4.2.

**Challenge**: A decides a target identity $u^*$ different from the issued identities in Phase 1.

**Phase 2**: A issues more queries as in Phase 1. It is not allowed to submit an Extract query for the target identity $u^*$.

**Output**: After Phase 2, A can get $k$ personal keys $K_i = (u_i + H_0(Q_i) + a)^{-1}P$, where $k$ is the number of Extract queries and $1 \leq i \leq k$. Suppose that A successfully outputs a personal $K^* = (u^* + H_0(Q^*) + a)^{-1}P$ for $u^*$ with probability $\varepsilon$. Let $u_i + H(Q_i)$ and $a$ be $h_i$ and $y$, respectively. The personal keys $K_i$’s can be rewritten as $\frac{1}{h_i+y}P$ for $1 \leq i \leq k$. Hence, $K^* = (u^* + H_0(Q^*) + a)^{-1}P = \frac{1}{h^*+y}P$ is a solution to the $k$-CAA.

After the simulation, $\beta_2$ does not abort the game with the probability $\zeta^{\text{enc}}$ as in Theorem 4.2. At the end of simulation, A successfully outputs the personal key for the target identity $u^*$ with the probability $\varepsilon$. Finally, $\beta_2$ can obtain the solution to the $k$-CAA with at least $(1 - \zeta)$.

We can conclude that $\beta_2$ can solve the $k$-CAA with the probability $\varepsilon_3 \geq \zeta^{\text{enc}}(1 - \zeta)\varepsilon$.

4.2. Efficiency analysis. A comparison of the proposed scheme with previous works is shown in Table 1. Suppose that the size of the plaintext is $\lambda$, where $\lambda$ is the security parameter. Let $t$ denote the number of traitors and $n$ denote the total number of subscribers. The rates of the ciphertext and the key are defined as ciphertext/plaintext and key/plaintext, respectively. According to Table 1, the sizes of the ciphertext, the
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Ciphertext Rate</th>
<th>Personal Key Rate</th>
<th>Encryption Key Rate</th>
<th>Max Traceable Collusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chor et al.'s scheme [8]</td>
<td>$O(t^2 \log n)$</td>
<td>$O(t^2 \log n)$</td>
<td>$O(t^2 \log n)$</td>
<td>$t$</td>
</tr>
<tr>
<td>Mitsunari et al.'s scheme [19]</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$1$</td>
</tr>
<tr>
<td>Boneh et al.'s scheme [11]</td>
<td>$O(\sqrt{n})$</td>
<td>$O(1)$</td>
<td>$O(\sqrt{n})$</td>
<td>$n$</td>
</tr>
<tr>
<td>Boneh and Naor’s scheme [10]</td>
<td>$O(1)$</td>
<td>$O(t^2 \log n)$</td>
<td>$O(t^2 \log n)$</td>
<td>$t$</td>
</tr>
<tr>
<td>The proposed scheme</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$t$</td>
</tr>
</tbody>
</table>

The proposed scheme and Mitsunari et al.'s scheme cannot resist a collusion attack. An important advantage of the proposed scheme is that the lengths of the broadcast message and the personal/encryption key are independent of the number of subscribers. That is, the broadcast message and the personal/encryption key sizes in the proposed scheme are shorter than those in previous works. Furthermore, the proposed scheme will work as long as the pirate decoder has fewer than $t$ personal keys.

5. Conclusions. This paper has presented the first secret key traitor tracing scheme with dispute settlement that achieves the security requirements of collusion-resistance, falsification-free and exposure-resistance. Note that in most of the previously proposed traitor tracing schemes, the enabling-block is independent of the intended content. This may cause a dispute in the case that any malicious subscriber may have the ability to falsify an unintended content for framing the broadcast center. In the proposed scheme, a coherent relationship between the intended content and the corresponding enabling-block is made up, and this provides robust evidence for dispute settlement by the adjudicator in the system. However, making up such coherent relationship between the intended content and the corresponding enabling-block just takes a few computational and communicational overheads by introducing $P_1$ in $EBK$.

Acknowledgement. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC 97-2219-E-001-001, NSC 97-2219-E-011-006, NSC 96-2221-E-011-148-MY2 and NSC 97-2221-E-182-032.

REFERENCES


