Fundamenta Informaticae XXI (2001) 1001–1014
IOS Press

Generic Codes Based Traitor Tracing Scheme with Revocation Ability from Attributes Based Encryption

Xingwen Zhao [C]
School of Telecommunication Engineer
Xidian University
Xi'an, 710071, China
sevenzhao@hotmail.com

Fangguo Zhang[*]
School of Information Science and Technology
Sun Yat-sen University
Guangzhou, 510275, China
isszhfg@mail.sysu.edu.cn

Abstract. Traitor tracing is needed because some users in broadcast encryption system may give out their decryption keys to construct pirate decoders. Many codes based traitor tracing scheme were proposed. However, as stated by Billet and Phan in ICITS 2008, they lack of revocation ability. We provide a generic scheme of codes based traitor tracing with revocation ability, based on ciphertext-policy attributes based encryption with expressive access policies consisted of multiple AND-gates and OR-gates. Revocation ability helps to disable identified traitors in each broadcast, so as the broadcast encryption system can be more practical. Our method shows how to construct a trace and revoke system based on collusion secure codes, and it can be extended to adopt other codes such as identifiable parent property (IPP) codes. Our method presents an answer to the problem left open by Billet and Phan.

Keywords: Broadcast encryption, traitor tracing, revocation, collusion secure codes, identifiable parent property codes.

1. Introduction

Broadcast encryption provides a convenient method to distribute digital content to subscribers over an insecure broadcast channel so that only the qualified users can recover the data. Broadcast encryption is quite useful and enjoys many applications including pay-TV systems, distribution of copyrighted materials such as DVD.

Address for correspondence: Mail Box #105, Xidian University, Xi'an, 710071, China
[C]Corresponding author
[*]This work is partially supported by the National Natural Science Foundation of China (No. 60773202, 61070168).

Because some users (called traitors) may give out their decryption keys to construct pirate decoders, the ability of traitor tracing is needed for broadcast encryption system. The first traitor tracing scheme against pirate decoders was presented by Chor, Fiat and Naor in [11]. Since then, many works have been presented. Here, we discuss some of them in details.

1.1. Related Works On Traitor Tracing Against Pirate Decoders

Since the introduction of traitor tracing by Chor, Fiat and Naor in [11], many traitor tracing schemes against pirate decoders were proposed and they can be roughly classified into three categories.

The first category is called combinatorial, as in [11, 28, 13, 22]. These schemes carefully choose some subsets of keys to be put in each decoder box. By analyzing the keys used in a pirate decoder, it is possible to trace one of the traitors. Another category is called algebraic, as in [18, 5, 23, 21, 7, 8, 14, 25]. These schemes use algebraic method to assign private keys to users, and the broadcasting can be done in public since public-key techniques are used. Collusion secure codes based schemes can be regarded as the third category, which combines ideas from the two previous classes. For instance, [17, 9, 1, 6, 3, 10] belong to this category. These schemes assign keys to each user according to each bit of his/her codeword. By analyzing the keys used in each bit positions, the tracer can recover the codeword embedded in the decoder and trace back to at least one of the traitors. However, it is pointed out in [3] that it is still an open problem whether codes based traitor tracing scheme can achieve revocation.

Some schemes [9, 8, 14, 25] allow public traceability, which means the tracing can be performed by anyone and is not limited to the tracing authority.

When traitors are found, it is desirable to make them useless. However, not all traitor tracing schemes support revocation. Many schemes merely consider the tracing of traitors, and they do not consider the revocation of traitors. Some schemes [23, 22, 8, 14] combine the tracing and revoking abilities to make the schemes more practical.

Some works [16, 4] focus on attacks against traitor tracing schemes. Kiayias and Pehlivanoglu [16] presented pirate evolution attack against schemes based on subset-cover revocation framework [22]. Billet and Phan [4] presented new attack named “Pirates 2.0” mainly against schemes based on traceability codes (collusion secure codes and identifiable parent property codes) and schemes based on subset-cover revocation framework. The attack shows that users can release certain part of their private keys in a public way, so that pirate decoders can be built from the public information while each traitor remains anonymous.

1.2. Traitor Tracing in Attributes Based Encryption

Some works in attributes based encryption [20, 31, 29] also consider finding key abusers in ABE systems. Li et al. [20] presented an accountable anonymous CP-ABE scheme, in which additional information of each user is embedded into his attribute private key. The tracer sets ciphertext-policy as initial attributes for which the decoder can decrypt the ciphertext, then gradually adds attributes into ciphertext-policy so as to reduce the suspected set of users to small group (and finally the traitor). Yu et al. [31] defined an abuse free KP-ABE scheme. They define bits of user’s ID as his attributes and trace to abusers by specifying access structure with attributes corresponding to each ID. Wang et al. [29] introduced attributes based traitor tracing by following the method in [1], however, their scheme does not support revocation.

1.3. Our Contributions

As discussed above, revocation ability is needed if traitors are found. Moreover, revocation ability can enable commercial broadcast encryption systems to send messages only to users who fulfill the payments. As stated in [3], many codes based traitor tracing scheme were proposed, however, they lack of revocation ability. In this paper, we shows how to realize codes based traitor tracing scheme with revocation ability, which presents an answer to the problem left open by [3].

There are two main contributions in this paper.

1. We describe a generic construction of codes based traitor tracing scheme with revocation ability, based on ciphertext-policy attributes based encryption (CP-ABE) with expressive access policies that consisted of multiple AND-gates and OR-gates. Since OR-gate can be realized by concatenation of different sets of ciphertext, CP-ABE realizing only AND operator can also be used to construct codes based traitor tracing scheme with revocation ability.

2. Our generic scheme is based on collusion secure codes. We show that our method can be extended to identifiable parent property (IPP) codes with only a few adjustments.

1.4. Organization

The remainder of this paper is organized as follows. In Section 2 we introduce some tools useful in our traitor tracing scheme. In Section 3 we describe the protocol model and security requirements of our codes based traitor tracing scheme. The generic scheme of traitor tracing with revocation and its security analysis are described in Section 4. We also describe how to extend it to IPP codes. We make some discussions on the proposed scheme in Section 5. Section 6 concludes our paper.

2. Building Tools

2.1. Collusion Secure Codes

We first review the definition of collusion secure codes required for constructing our traitor tracing scheme. The definition is similar to that in [6].

- For a word $\bar{w} \in \{0,1\}^L$ we write $\bar{w} = w_1 \dots w_L$, where $w_i \in \{0,1\}$ is the ith bit of $\bar{w}$ for $i = 1, \dots, L$.

- Let $W = \{\bar{w}^{(1)}, \dots, \bar{w}^{(t)}\}$ be a set of words in $\{0,1\}^L$. We say that a word $\bar{w} \in \{0,1\}^L$ is feasible for W if for all $i = 1, \dots, L$ there is a $j \in \{1, \dots, t\}$ such that $\bar{w}_i = \bar{w}_i^{(j)}$. For example, if W consists of the two words $\binom{0\ 1\ 1\ 0\ 0}{0\ 0\ 1\ 1\ 0}$, then all words of the form $[0\ \binom{1}{0}\ 1\ \binom{0}{1}\ 0]$ are feasible for W.

- For a set of words $W \subseteq \{0,1\}^L$ we say that the feasible set of W, denoted $F(W)$, is the set of all words that are feasible for W.

The collusion secure code can be denoted with a pair of polynomial time algorithms (G, T) defined as follows:

- Algorithm G, called a code generator is a probabilistic algorithm that takes a pair (N, ϵ) as input, where N is the number of words to output and $\epsilon \in (0,1)$ is a security parameter. The algorithm outputs a pair (Γ, TK). Here Γ (called a code) contains N words in $\{0,1\}^L$ for some $L > 0$ (called the code length). TK is called the tracing key.
- Algorithm T, called a tracing algorithm, is a deterministic algorithm that takes as input a pair ($\bar{w}^*$, TK) where $\bar{w}^* \in \{0,1\}^L$. The algorithm outputs a subset S of $\{1,\ldots,N\}$. Informally, elements in S are accused of creating the word $\bar{w}^*$.

The collusion resistant property of collusion secure code (G, T) is defined using the following game between a challenger and an adversary. Let N be an integer and $\epsilon \in (0,1)$. Let C be a subset of $\{1,\ldots,N\}$. Both the challenger and adversary are given (N, ϵ, C) as input. Then the game proceeds as follows:

1. The challenger runs $G(N, \epsilon)$ to obtain (Γ, TK) where $\Gamma = \{\bar{w}^{(1)},\ldots,\bar{w}^{(N)}\}$. It sends the set $W := \{\bar{w}^{(i)}\}_{i\in C}$ to the adversary.
2. The adversary outputs a word $\bar{w}^* \in F(W)$.

We say that the adversary $\mathcal{A}$ wins the game if $T(\bar{w}^*, \text{TK})$ is empty or not a subset of C. We denote $Adv_{CR}^{\mathcal{A},G(N,\epsilon),T,C}$ as the advantage that $\mathcal{A}$ wins the collusion resistant game.

Definition 2.1. A collusion secure code (G,T) is said to be fully collusion resistant if for all polynomial time adversaries $\mathcal{A}$, all $N > 0$, all $\epsilon \in (0,1)$, and all $C \subseteq \{1,\ldots,N\}$, we have $Adv_{CR}^{\mathcal{A},G(N,\epsilon),T,C}$ is negligible (less than ϵ).

A collusion secure code (G,T) is said to be t-collusion resistant if for all polynomial time adversaries $\mathcal{A}$, all $N > t$, all $\epsilon \in (0,1)$, and all $C \subseteq \{1,\ldots,N\}$ of size at most t, we have $Adv_{CR}^{\mathcal{A},G(N,\epsilon),T,C}$ is negligible (less than ϵ).

Our readers can refer to [6] for known results on collusion secure codes. Additionally, Boneh and Naor [6] also constructed δ-robust Boneh-Shaw codes in order to trace high error-rate pirate decoders.

2.2. Identifiable Parent Property Codes

The identifiable parent property (IPP) code was introduced in [15] to fight against piracy of software.

Let $Code(L,d)$ be an IPP code over the finite field of q elements $\mathbb{F}_q$, where L is the code length and d is the minimum distance of the code. We denote $Desc(W)$ as the set of descendants of any subset $W = \{w^1,\ldots,w^c\} \subseteq Code(L,d)$, where $w^i = (w_1^i,\ldots,w_L^i)$, $w_j^i \in \mathbb{F}_q$, $j = 1,\ldots,L$. $Desc(W)$ is defined as

$$Desc(W) = \{y = (y_1,\ldots,y_L) \in \mathbb{F}_q^L | y_i \in \{w_i^j | w^j \in W\}, 1 \le i \le L\}. \quad (1)$$

Definition 2.2. A code C is a c-traceability code (denoted as c-TA), for $c > 0$, if for all subsets $W \subseteq C$ of at most c codewords, if $y \in Desc(W)$, then there exists a $t \in W$ such that $d(y,t) \le d(y,w)$ for all $w \in C - W$.

Silverberg, Staddon and Walker (in [26]), Fernandez and Soriano (in [12]) showed how to obtain IPP codes for traitor tracing. We refer our readers to these references for more details about IPP codes.

2.3. Attributes Based Encryption

A ciphertext-policy attributes based encryption (CP-ABE) scheme consists of four algorithms as follows:

- **Setup**(1^λ, L). This algorithm inputs L as the number of attributes in the system, 1^λ as security parameter and outputs public key PK and master secret key MK.
- **KeyGen**(PK, MK, A). Inputting public key PK, master secret key MK and the user's set of attributes A, this algorithm outputs the private key of the user SK.
- **Enc**(PK, AP, m). Inputting public key PK, the specified access policy AP and the message m, this algorithm outputs ciphertext C.
- **Dec**(PK, SK, AP, C). Inputting public key PK, a user's private key SK, a ciphertext C and the specified access policy AP for the ciphertext, the algorithm outputs the recovered message m if the user's set of attributes A satisfies AP.

For instance, if an access policy AP as (A_1 = "Manager") AND (A_2 = "ACC Com." OR A_2 = "DFF Com.") is used in encryption, only users with attributes A_1 = "Manager" and A_2 = "ACC Com." or users with attributes A_1 = "Manager" and A_2 = "DFF Com." can recover the messages. We use $\wedge$ ($\vee$) to denote AND (OR) operators respectively. Readers can refer to [30] for more details about access policy (access structure) and ciphertext-policy attribute-based encryption.

3. Protocol Model and Security Requirements

3.1. Protocol Model

The protocol model for our traitor tracing scheme consists of four algorithms (Setup, Encrypt, Decrypt, Trace) described as follows.

- **Setup**(1^λ, N). It is a probabilistic algorithm that given 1^λ and the number of users in the system N, outputs a public broadcast-key BK, a secret trace-key TK, and the private user-key SK_u for each user $u \in \{1, \ldots, N\}$.
- **Encrypt**(BK, S M). It is a probabilistic algorithm that given a broadcast-key BK, a set of receivers S and a message M, a broadcast ciphertext C is generated.
- **Decrypt**(SK_u, C). It is an algorithm that given a broadcast ciphertext C and the private user-key SK_u of user u, returns the recovered messages M or $\perp$.
- **Trace**$^{\mathcal{D}}$(TK). It is an algorithm that given a pirate decoder $\mathcal{D}$ and private trace-key TK, it queries decoder $\mathcal{D}$ as a black-box oracle and then outputs a traitor set $T \subseteq \{1, \ldots, N\}$.

3.2. Security Requirements

- **Correctness**. Each honest user is able to recover the messages in normal broadcasting.

- **Semantic Security**. The users cannot obtain any information of messages encrypted in the broadcast ciphertext, if their identities are not included in the specified receiver set.

  The semantic security of proposed traitor tracing scheme is defined using the following game between a challenger and an adversary. The game proceeds as follows:

  1. The challenger runs $G(N, \epsilon)$ to obtain (Γ, TK) where $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ and $\bar{w}^{(i)}$ is the codeword (and attributes) for user i. The challenger also selects a ABE scheme with public parameters mpk. It sends Γ and mpk to the adversary.
  2. The adversary selects a subset of Γ, denoted as C. The adversary can query the challenger for decryption keys of the codewords in C. The challenger generates the keys as in ABE and gives them to the adversary.
  3. The adversary submits two messages (m_1, m_0) and a set of codewords S for challenging, with $\bar{w}^* \notin C$ for each $\bar{w}^* \in S$. The challenger chooses a fair coin $b \in \{0, 1\}$ and encrypts m_b using S as the set of designated receivers. The ciphertext is sent to the adversary, and the adversary is required to output a guessed bit b'.

  If $b' = b$, the adversary wins the game.

- **Collusion Resistant**. Collusion of users cannot produce a decoder that cannot be traced to any of these users.

  The collusion resistant property of proposed traitor tracing scheme is defined using the following game between a challenger and an adversary. Let (G, T) be a collusion secure code. Let N be an integer and $\epsilon \in (0, 1)$. Then the game proceeds as follows:

  1. The challenger runs $G(N, \epsilon)$ to obtain (Γ, TK) where $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ and $\bar{w}^{(i)}$ is the codeword (and attributes) for user i. The challenger also selects a ABE scheme with public parameters mpk. It sends Γ and mpk to the adversary.
  2. The adversary selects a subset of Γ, denoted as C. The adversary can query the challenger for decryption keys of the codewords in C. The challenger generates the keys as in ABE and gives them to the adversary.
  3. The challenger asks the adversary to decrypt ciphertexts a number of times and recovers a codeword $\bar{w}^*$.

  We say that the adversary $\mathcal{A}$ wins the game if $T(\bar{w}^*, \text{TK})$ is empty or not a subset of C.

4. The Proposed Generic Scheme Based on Collusion Secure Codes

4.1. Our Idea

As a traitor tracing scheme based on collusion secure codes, a proper codeword and its corresponding decryption key should be assigned to each user. By running $G(N, \epsilon)$, (Γ, TK) are obtained, where N is total number of users, ϵ is the security parameter, $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ is a set of N codewords and TK is the tracing key for Γ. Suppose the codeword length is L, and we need an attributed based encryption (ABE) scheme with L attributes. We denote these attributes as (A_1, $\ldots$, A_L), and the value of each

attribute is 1 or 0. For instance, if a codeword $\bar{w}^{(i)}$ is assigned to user i, he obtain a set of attributes with $A_k = \bar{w}_k^{(i)}$, $k = 1, \ldots, L$. The key generation algorithm KeyGen of ABE is used to generate decryption key for each codeword.

The next step is to find a method to combine broadcasting and tracing, avoiding the pirate decoder noticing the tracing behavior while achieving revocations. The revocation is achieved by specifying all receivers' attributes in access policy. For instance, if a message is sent to user i (with codeword $\bar{w}^{(i)}$) and user j (with codeword $\bar{w}^{(j)}$), the access policy will be $(A_1 = \bar{w}_1^{(i)} \wedge \cdots \wedge A_L = \bar{w}_L^{(i)}) \vee (A_1 = \bar{w}_1^{(j)} \wedge \cdots \wedge A_L = \bar{w}_L^{(j)})$. Combination of broadcasting and tracing is done by using the method by Boneh and Naor [6], which will provide two sets of ciphertext for the tracing position. For instance, if the tracing position is $i \in \{1, \ldots, L\}$, two sets of ciphertext are generated (one set for attribute $A_i = 1$ and the other set for $A_i = 0$). Thus, the access policy will be expressed as $(A_i = 1 \wedge AP_S) \vee (A_i = 0 \wedge AP_S)$, where AP_S is access policy for the set of specified receivers (denoted as S) and computed as $AP_S = \bigvee_{\forall \bar{w}^{(j)} \in S} (\bigwedge_{i=1}^{L} (A_i = \bar{w}_i^{(j)}))$. When in broadcasting, both sets of ciphertext are generated for a same message. When in traitor tracing, these two sets of ciphertext are generated for different messages (m_1 and m_0). If the pirate decoder returns m_1, the tracer decides that attribute $A_i = 1$. Else, $A_i = 0$. After all tracing position are completed, the set of recovered attributes is regarded as the recovered codeword $\bar{w}^*$. $T(\bar{w}^*, \text{TK})$ is run to output a subset of traitors.

The above method is enough for tracing perfect pirate decoders that should correctly decrypt all well-formed ciphertext [6]. As for imperfect pirate decoders, which may refuse to decrypt on certain tracing positions in addition to coalition positions, collusion secure codes against bit erasure [27, 6, 24] are enough for fighting against such pirate behaviors. At least a traitor can be captured with high probability.

What we need is an expressive CP-ABE scheme that is at least semantically secure (CPA secure: secure against chosen plaintext attacks). It is a must for an ABE scheme to consider collusion resistance in its security model, so semantic security is enough for us. Since OR-gate can be realized by concatenation of different sets of ciphertext, CP-ABE scheme that realizes only AND operator can also be used to construct the tracing scheme. We mainly focus on how to construct codes based traitor tracing scheme with revocation ability from ABE scheme, so we do not discuss much about ABE. Secure expressive CP-ABE can be found in [2, 19, 30].

4.2. The Scheme

We use (ABE.Setup, ABE.KeyGen, ABE.Enc, ABE.Dec) to denote a CP-ABE scheme that supports expressive access policies of multiple AND and OR operators. Our generic traitor tracing scheme based on collusion secure codes can be expressed as follows.

- **Setup**$(1^\lambda, N)$.It is a probabilistic algorithm, in which a trusted party, given 1^λ and the number of users in the system N, selects $\epsilon \in (0, 1)$ and runs collusion secure code generation algorithm **G**(N, ϵ) to generate a pair (Γ, TK). The set $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ contains N codewords in $\{0, 1\}^L$, where L is the codeword length which is decided by total number of users N, collusion threshold t and ϵ [6]. TK is the tracing key for Γ. $\bar{w}^{(u)}$ is assigned to user u as the set of attributes, where $1 \leq u \leq N$. $\bar{w}^{(u)}$ is used to represent user u when in broadcasting. The trusted party selects an ABE scheme and runs its **ABE.Setup**$(1^\lambda, L)$ to generate a master key pair (PK, MK). The trusted party runs **ABE.KeyGen**$(PK, MK, \bar{w}^{(u)})$ to generate decryption key SK_u for each user

u. Each decryption key is transferred to user over a secure and authenticated channel which is not considered in this paper. MK is the master public key and also the broadcast-key of the tracing scheme.

- **Encrypt**(PK, S, m). Anyone who wants to encrypt a message m to a set of receivers denoted as S (users are represented by their codewords), given the broadcast-key PK, selects random position $i \in \{1, \ldots, L\}$ to generate an access policy AP as (A_i = 1) $\wedge AP_S$, where

$$AP_S = \bigvee_{\forall \bar{w}^{(j)} \in S} \left(\bigwedge_{i=1}^{L} (A_i = \bar{w}_i^{(j)})\right)$$

and runs **ABE.Enc**(PK, AP, m) to obtain a ciphertext C_1. Then AP is set as (A_i = 0) $\wedge AP_S$, and **ABE.Enc**(PK, AP, m) is run to obtain another ciphertext C_0. The final ciphertext C is the concatenation of these two sets of ciphertext (C_1, C_0). (i, C) is broadcast to all users.

- **Decrypt**(SK$_u$, C). Any user $u \in S$, given (i, C) and the user's attribute value ($A_i = b, b \in \{0, 1\}$) in position i, obtains the set of ciphertext C_b for access policy ($A_i = b$) $\wedge AP_S$ from C. Then he recovers the message by running **ABE.Dec**(PK, SK$_u$, ($A_i = b$) $\wedge AP_S$, C_b) by using his decryption key SK$_u$. User u returns the recovered messages m or $\perp$.

- **Trace**$^{\mathcal{D}}$(TK). Given a perfect pirate decoder $\mathcal{D}$ constructed by a set of traitors $\mathcal{T}$, the trusted party queries decoder $\mathcal{D}$ as a black-box oracle. Denoted S as a set of suspected users and $\mathcal{D}$ can decrypt the ciphertext for S. For i = 1, ..., L, the trusted party acts as follows:

  1. generates access policy AP_1 as ($A_i = 1$) $\wedge AP_S$ and AP_0 as ($A_i = 0$) $\wedge AP_S$;
  2. selects two messages m_1 and m_0. The tracer runs **ABE.Enc**(PK, AP_1, m_1) to obtain ciphertext C_1, and runs **ABE.Enc**(PK, AP_0, m_0) to obtain ciphertext C_0. These two sets of ciphertext (C_1, C_0) form C. (i, C) is fed to the decoder;
  3. if the pirate decoder outputs m_b where $b \in \{0, 1\}$, the trusted party decides that the decoder contains a codeword $\bar{w}^*$ with $\bar{w}_i^* = b$.

  After the trusted party obtains the recovered codeword $\bar{w}^*$ = $\bar{w}_1^* \ldots \bar{w}_L^*$, it runs the tracing algorithm of collusion secure code as **T**($\bar{w}^*$, TK) and outputs a set of traitors as T = **T**($\bar{w}^*$, TK) $\cap S$.

4.3. Security Analysis

Correctness. The correctness is straightforward due to the correctness of the ABE scheme.

Theorem 4.1. The generic traitor tracing scheme is semantically secure assuming the ABE scheme is semantically secure.

Proof:
Suppose there is an adversary $\mathcal{A}$ for our generic traitor tracing scheme. The challenger $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

1. $\mathcal{B}$ runs $G(N, \epsilon)$ to obtain (Γ, TK) where $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ and $\bar{w}^{(i)}$ is the codeword (and attributes) for user i. Suppose the length of codeword is L. $\mathcal{B}$ (as the adversary) forwards L to the challenger $\mathcal{CH}$ to initial semantic security game of an ABE scheme. Suppose the public parameter of ABE scheme is mpk. $\mathcal{B}$ sends Γ and mpk to the adversary $\mathcal{A}$.

2. $\mathcal{A}$ selects a subset of Γ, denoted as C. $\mathcal{A}$ queries $\mathcal{B}$ for decryption keys of the codewords in C. $\mathcal{B}$ forwards the queries to $\mathcal{CH}$ who will generate the keys as in ABE. $\mathcal{B}$ returns them to the adversary.

3. $\mathcal{A}$ submits two messages (m_1, m_0) and a set of codeword W^* for challenging, with requirement that $\bar{w}^{(j)} \notin C$ for each $\bar{w}^{(j)} \in W^*$. $\mathcal{B}$ selects a tracing position i, a random $b \in \{0, 1\}$ and constructs an access policy AP^* as ((A_i = b) $\wedge AP_{W^*}$), where

$$AP_{W^*} = \bigvee_{\forall \bar{w}^{(j)} \in W^*} (\bigwedge_{i=1}^{L} (A_i = \bar{w}_i^{(j)})).$$

   The operation above imports a loose factor 1/2 into reduction, denoted as case 1. $\mathcal{B}$ selects a random message $m_r \notin \{m_1, m_0\}$ and forwards (m_1, m_r), AP^* to $\mathcal{CH}$. In a selective-attribute model, AP^* should be sent to $\mathcal{CH}$ before initialing ABE scheme. Here we introduce another loose factor 1/2 into reduction, denoted as case 2. After obtaining the ciphertext C_b, $\mathcal{B}$ constructs another set of ciphertext C_{1-b} for access policy ((A_i = 1-b) $\wedge AP_{W^*}$) to encrypt m_r. If b=1, $\mathcal{B}$ arranges the ciphertext C as (C_b, C_{1-b}). Else, $\mathcal{B}$ arranges the ciphertext C as (C_{1-b}, C_b). $\mathcal{B}$ forwards C as well as the position i to the $\mathcal{A}$, and $\mathcal{A}$ is required to output a guessed bit b'. If $b' = 0$, $\mathcal{A}$ is not helpful and $\mathcal{B}$ return a random bit to $\mathcal{CH}$. If $b' = 1$, $\mathcal{B}$ return 1 to $\mathcal{CH}$ as the answer.

As we notice that, $\mathcal{B}$ is running against the semantic security game of ABE with the help of $\mathcal{A}$. There are two cases of reduction lost. In case 1, $\mathcal{A}$ may be able to decrypt ciphertext for ((A_i = 1-b) $\wedge AP_{W^*}$), may be unable to decrypt ciphertext for ((A_i = b) $\wedge AP_{W^*}$). In case 2, $\mathcal{CH}$ may select m_r as the challenge message, which will make $\mathcal{A}$ useless for $\mathcal{B}$. Thus, if $\mathcal{A}$ has any advantage ϵ in breaking semantic security of our generic traitor tracing scheme, it can be used to break the semantic security game of ABE with advantage $\epsilon/4$. □

Theorem 4.2. The generic traitor tracing scheme is t-collusion secure assuming the ABE scheme is semantically secure and the the collusion secure code is t-collusion secure.

Proof:

The t-collusion resistant game of our proposed scheme is played between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ as described in Section 3.

Let (G, T) be a collusion secure code. Let N be an integer and $\epsilon \in (0, 1)$. The challenger runs $G(N$, $\epsilon)$ to obtain (Γ, TK) where $\Gamma = \{\bar{w}^{(1)}, \ldots, \bar{w}^{(N)}\}$ and $\bar{w}^{(i)}$ is the attributes for user i. The challenger also selects a ABE scheme with public parameters mpk. It sends Γ and mpk to the adversary. Then, the adversary selects a subset of Γ, denoted as C with $|C| \leq t$. The adversary can query the challenger for decryption keys of the codewords in C. The challenger generates the keys as in ABE and gives them to the adversary.

When it is time for the challenger to query the adversary on decryptions, for the tracing position i = $1, \ldots, L$, the challenger queries the adversary with messages m_1 and m_0 encrypted in the access policies

AP_1 = $((A_i = 1) \wedge AP_S)$ and AP_0 = $((A_i = 0) \wedge AP_S)$ respectively. S is a set of suspect users, and AP_S (computed as in algorithm **Encrypt**) is access policy for users in S. There are four cases for the decoder:

- Case 1: The adversary does not hold any codeword in S. As we proved in Theorem 4.1, the adversary will always output a random message other than m_1 and m_0 since the attributes do not satisfy the access policies. The probability that the adversary outputs the right message is at most $2/|\mathcal{M}|$, where $|\mathcal{M}|$ is the number of messages in the message space. In this case, the tracer changes to another set to continue. The tracer will be deceived with probability at most $2/|\mathcal{M}|$;

- Case 2: The adversary holds at least one codeword in S (we denote the set of these codewords that are in S and at the same time held by the adversary as S_A), and all codewords in S_A contain "1" in tracing position i. That is to say, all $\bar{w}^{(j)} \in S_A$ satisfy $\bar{w}_i^{(j)}$ = 1. Thus, the adversary will always output m' = m_1. The recovered bit $\bar{w}_i^*$ will always be 1. Since all codewords in S_A does not contain "0" in position i, the probability that the adversary outputs "0" is less than $Adv_{SS}^{\mathcal{A}}$, the probability that the adversary breaks the semantic security game of ABE scheme;

- Case 3: The adversary holds at least one codeword in S (denoted as S_A), and all codewords in S_A contain "0" in tracing position i. That is to say, all $\bar{w}^{(j)} \in S_A$ satisfy $\bar{w}_i^{(j)}$ = 0. Thus, the adversary will always output m' = m_0. The recovered bit $\bar{w}_i^*$ will always be 0. Since all codewords in S_A does not contain "1" in position i, the probability that the adversary outputs "1" is less than $Adv_{SS}^{\mathcal{A}}$, the probability that the adversary breaks the semantic security game of ABE scheme;

- Case 4: The adversary holds at least one codeword in S (denoted as S_A), and codewords in S_A contain either "0" or "1" in tracing position i. No matter which message the adversary outputs (m_1 or m_0), $\bar{w}_i^*$ must be in the feasible set of S_A.

Therefore, the final recovered codeword $\bar{w}^* \in F(S_A)$. From the assumption that collusion secure code (G, T) is t-collusion resistant, the probability that $T(\bar{w}^*, \text{TK})$ is empty or not a subset of S_A is less than ϵ. Thus, the probability that the adversary breaks the property of t-collusion resistance of our generic traitor tracing scheme is less than $(2/|\mathcal{M}|)^L + 2L \cdot Adv_{CPA}^{\mathcal{A}} + \epsilon$. □

As we notice that, when $t = N$, our generic scheme is fully collusion resistant.

4.4. Extension to IPP codes

Our generic scheme can be extended to traitor tracing scheme based on IPP codes. As for a q-ary IPP code, let $\Theta = \{Sym_1, \ldots, Sym_q\}$ be the set containing these q symbols. Then each user will receive a codeword (also the set of attributes) from Θ^L, denoted as $\bar{w} = (\bar{w}_1, \ldots, \bar{w}_L)$, with $\bar{w}_i \in \Theta$, $i = 1, \ldots, L$. For instance, user i receives a codeword $\bar{w}^{(i)}$, so he obtains a set of attributes $(A_1, \ldots, A_L)$ with $A_k = \bar{w}_k^{(i)}$, $k = 1, \ldots, L$. Each user will receive decryption key for the specified set of attributes as in ABE scheme.

When in broadcasting, the access policy will be $((A_i = Sym_1) \wedge AP_S) \vee \ldots \vee ((A_i = Sym_q) \wedge AP_S)$, where S is the set of receivers and AP_S is access policy constructed as described in algorithm **Encrypt**. The ciphertext is in fact the concatenation of q sets of ciphertext (encrypting the same message) for access policies $((A_i = Sym_1) \wedge AP_S), \ldots, ((A_i = Sym_q) \wedge AP_S)$ respectively.

When in decryption, each user in S chooses the part of ciphertext corresponding to his attributes to recover the message. For instance, if the ciphertext is encrypted for position i and user $j \in S$ holds attribute $A_i = Sym_k$, he will chooses the part of ciphertext of access policy $((A_i = Sym_k) \wedge AP_S)$ to perform decryption.
When in tracing, the ciphertexts will encrypt different messages for different symbols in tracing position i, i.e. q distinct messages $(m_1, \ldots, m_q)$ are encrypted and m_j is encrypted for $((A_i = Sym_j) \wedge AP_S)$, $j = 1, \ldots, q$. If the pirate decoder outputs m_j, the tracer decides that the decoder contains a codeword $\bar{w}^*$ with $\bar{w}^*_j = Sym_j$. When tracing on all positions is completed, the recovered codeword $\bar{w}^* = (\bar{w}^*_1 \ldots \bar{w}^*_L)$ will be input to the decoding (tracing) algorithm for IPP code, and a list of parent codewords are obtained.

5. Discussion of the Proposed Scheme

We will analyze the efficiency of our proposed generic scheme on ciphertext length, public key size, private key size, encryption cost and decryption cost. We only compare our scheme with codes based traitor tracing schemes in which all bits of the codeword are used [17, 9, 32] in each encryption. Traitor tracing schemes in which one bit in codeword (or u bits with $u < L$) is used [6, 3, 10] are not considered, since in these schemes the pirate decoders may be untraceable [32].
We also make a discussion on an useful property named public collaboration resistance.

5.1. Efficiency Discussion

- Ciphertext Length. The ciphertext length is twice the ciphertext length of ABE scheme that our scheme is based on. If using [30] for an instance, there are $2L$ rows in Linear Secret Sharing Scheme (LSSS) matrix, so the ciphertext length is roughly $4L$, longer than the length roughly L in [17, 9] and $2L$ in [32].

- Public key size. The public key size is the same as that of ABE scheme. If using [30] for an instance, the public key size is constant, while the size is O(L) in [17, 9, 32].

- Private key size. The private key size is the same as that in ABE scheme. If using [30] for an instance, the private key size is O(L), roughly the same as in [17, 9], longer than constant length in [32].

- Encryption cost. The encryption cost is proportion to $|S|L$, i.e. the size of receivers set S multiplied by the length of codeword. If using [30] for an instance, each encryption needs roughly O($|S|L$) multiplications and O(L) exponentiations, while it needs roughly O(L) exponentiations in [17, 9, 32].

- Decryption cost. The decryption cost is proportion to L, i.e. the length of codeword. If using [30] for an instance, each decryption needs roughly O(L) pairings, O(L) multiplications and O(L) exponentiations, while it needs roughly O(L) exponentiations and O(L) multiplications in [17, 32], roughly O(L) multiplications and O(L) pairings in [9].

We can notice that, our scheme is less efficient (in ciphertext length, encryption cost and decryption cost) than codes based traitor tracing schemes without revocation ability. However, it will be better with more

advances achieved in the research of ABE. We also leave it as an open problem to construct efficient codes based traitor tracing scheme with revocation ability.

5.2. Public Collaboration Resistance

Public collaboration is presented by Billet and Phan [4] in EUROCRYPT 2009 as an attack against code-based traitor tracing schemes. The attack shows that users can release certain part of their private keys in a public way, so that pirate decoders can be built from the public information. Each traitor remains anonymous because a large number of users contain the same keys as those released in public. In our scheme, if several parts of incomplete decryption key are able to construct a useful pirate decoder for the original codewords, it presents a contradiction to semantic security of ABE scheme that our scheme is based on. If several parts of incomplete decryption key are able to construct a useful pirate decoder for a codeword other than original codewords, it presents a contradiction to collusion resistant security of ABE scheme (already considered in semantic security of most ABE schemes). The traitor should release its key as a whole in order to render it useful, so the codeword (and identity of traitor) can be immediately found out via $2L$ rounds of computations if the codeword (the set of attributes) is not released. Thus, Our scheme is resistant to public collaboration.

6. Conclusion

We describe a generic codes based traitor tracing scheme with revocation ability, based on CP-ABE with expressive access policies. In the scheme, the sender can specify a set of receivers who can recover the message. The generic scheme is based on collusion secure codes, and it can be extended to use IPP codes with only a few adjustments.

References

[1] Abdalla, M., Dent, A. W., Malone-Lee, J., Neven, G., Phan, D. H., Smart, N. P.: Identity-Based Traitor Tracing, *Public Key Cryptography* (T. Okamoto, X. Wang, Eds.), 4450, Springer, 2007.

[2] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption, *IEEE Symposium on Security and Privacy*, IEEE Computer Society, 2007.

[3] Billet, O., Phan, D. H.: Efficient Traitor Tracing from Collusion Secure Codes, *ICITS* (R. Safavi-Naini, Ed.), 5155, Springer, 2008.

[4] Billet, O., Phan, D. H.: Traitors Collaborating in Public: Pirates 2.0, *EUROCRYPT* (A. Joux, Ed.), 5479, Springer, 2009.

[5] Boneh, D., Franklin, M. K.: An Efficient Public Key Traitor Tracing Scheme, *CRYPTO*, 1999.

[6] Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext, *ACM Conference on Computer and Communications Security* (P. Ning, P. F. Syverson, S. Jha, Eds.), ACM, 2008.

[7] Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys, *EUROCRYPT*, 2006.

[8] Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system, *ACM Conference on Computer and Communications Security*, 2006.

[9] Chabanne, H., Phan, D. H., Pointcheval, D.: Public Traceability in Traitor Tracing Schemes, *EUROCRYPT*, 2005.

[10] Chen, Y.-R., Tzeng, W.-G.: A Public-Key Traitor Tracing Scheme with an Optimal Transmission Rate, *ICICS* (S. Qing, C. J. Mitchell, G. Wang, Eds.), 5927, Springer, 2009.

[11] Chor, B., Fiat, A., Naor, M.: Tracing Traitors, *CRYPTO*, 1994.

[12] Fernandez, M., Soriano, M.: Decoding codes with the identifiable parent property, *ISCC*, 2002.

[13] Fiat, A., Tassa, T.: Dynamic Traitor Training, *CRYPTO*, 1999.

[14] Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes, *ACM Conference on Computer and Communications Security* (E. Al-Shaer, A. D. Keromytis, V. Shmatikov, Eds.), ACM, 2010.

[15] Hollmann, H. D. L., van Lint, J. H., Linnartz, J.-P. M. G., Tolhuizen, L. M. G. M.: On Codes with the Identifiable Parent Property, *J. Comb. Theory, Ser. A*, **82**(2), 1998, 121–133.

[16] Kiayias, A., Pehlivanoglu, S.: Pirate Evolution: How to Make the Most of Your Traitor Keys, *CRYPTO*, 2007.

[17] Kiayias, A., Yung, M.: Traitor Tracing with Constant Transmission Rate, *EUROCRYPT*, 2002.

[18] Kurosawa, K., Desmedt, Y.: Optimum Traitor Tracing and Asymmetric Schemes, *EUROCRYPT*, 1998.

[19] Lewko, A. B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption, *EUROCRYPT* (H. Gilbert, Ed.), 6110, Springer, 2010.

[20] Li, J., Ren, K., Zhu, B., Wan, Z.: Privacy-Aware Attribute-Based Encryption with User Accountability, *ISC* (P. Samarati, M. Yung, F. Martinelli, C. A. Ardagna, Eds.), 5735, Springer, 2009.

[21] Mitsunari, S., Sakai, R., Kasahara, M.: A New Traitor Tracing, *IEICE transactions on fundamentals of electronics, communications and computer sciences*, **E85-A**(2), 2002, 481–484.

[22] Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers, *CRYPTO* (J. Kilian, Ed.), 2139, Springer, 2001.

[23] Naor, M., Pinkas, B.: Efficient Trace and Revoke Schemes, *Financial Cryptography*, 2000.

[24] Nuida, K.: A General Conversion Method of Fingerprint Codes to (More) Robust Fingerprint Codes against Bit Erasure, *ICITS* (K. Kurosawa, Ed.), 5973, Springer, 2009.

[25] Park, J. H., Lee, D. H.: Fully collusion-resistant traitor tracing scheme with shorter ciphertexts, *Des. Codes Cryptography*, **60**(3), 2011, 255–276.

[26] Silverberg, A., Staddon, J., Walker, J. L.: Efficient Traitor Tracing Algorithms Using List Decoding, *ASIACRYPT* (C. Boyd, Ed.), 2248, Springer, 2001.

[27] Sirvent, T.: Traitor tracing scheme with constant ciphertext rate against powerful pirates, Cryptology ePrint Archive, Report 2006/383, 2006, `http://eprint.iacr.org/`.

[28] Stinson, D. R., Wei, R.: Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes, *SIAM J. Discrete Math.*, **11**(1), 1998, 41–53.

[29] Wang, Y.-T., Chen, K.-F., Chen, J.-H.: Attribute-Based Traitor Tracing, *J. Inf. Sci. Eng.*, **27**(1), 2011, 181–195.

[30] Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, *Public Key Cryptography* (D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi, Eds.), 6571, Springer, 2011.

[31] Yu, S., Ren, K., Lou, W., Li, J.: Defending against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems, *SecureComm* (Y. Chen, T. Dimitriou, J. Zhou, Eds.), 19, Springer, 2009.

[32] Zhao, X., Zhang, F.: Traitor Tracing against Public Collaboration, *ISPEC* (F. Bao, J. Weng, Eds.), 6672, Springer, 2011.