

國立交通大學

資訊科學與工程研究所

碩士論文

最佳密文傳輸率的公開金鑰
背叛者追蹤系統



A Public-Key Traitor Tracing Scheme with Optimal
Transmission Rate

研究生：陳毅睿

指導教授：曾文貴 教授

中華民國九十八年六月

最佳密文傳輸率的公開金鑰背叛者追蹤系統
A Public-Key Traitor Tracing Scheme with Optimal Transmission Rate


研究生：陳毅睿

Student：Yi-Ruei Chem

指導教授：曾文貴

Advisor：Wen-Guey Tzeng

國立交通大學
資訊科學與工程研究所
碩士論文



A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

最佳密文傳輸率的公開金鑰背叛者追蹤系統

學生：陳毅睿

指導教授：曾文貴

國立交通大學資訊科學與工程研究所碩士班

摘要

如何在廣播環境中傳送加密的數位內容給合法的訂閱者是一個在許多的商業模式中 (像是付費電視、DVD等) 很廣泛的議題。而在這樣的環境中，為了要能夠有效嚇阻合法的數位內容訂閱者將其所擁有合法解密金鑰洩露出去，背叛者追蹤系統便因應而生了。在這篇論文中，我們提出了一個背叛者追蹤系統，其所使用的廣播金鑰可以是公開的。而且在我們的系統中，密文的傳輸率可以達到最佳 (也可以說是常數密文傳輸量)，也就是說，幾乎不需要多餘的頻寬即可加密傳送數位內容。而在追蹤背叛者的能力方面，我們提出的系統可以支援黑盒追蹤，也就是說，我們可以在不能直接察看非法解密器內部的情形下，仍然可以成功追蹤出該非法解密器是使用哪些背叛者的金鑰。而跟之前的所提出的一些相關系統相比，我們的系統在廣播金鑰以及使用者解密金鑰上，皆可以達到較低的儲存空間需求。

Keywords: 背叛者追蹤, fingerprinting碼, all-or-nothing轉換

A Public-Key Traitor Tracing Scheme with Optimal Transmission Rate


Student: Yi-Ruei Chen

Advisor: Dr. Wen-Guey Tzeng

Institute of Network Engineering College of Computer Science

National Chiao Tung University

Abstract



The way of transmitting the encrypted digital contents to the legitimate subscribers in a broadcast environment is a wide application for many commercial transactions (e.g. pay-TV, DVD, etc.). In order to discourage the legitimate subscribers from giving away their decryption keys, the traitor tracing system is very useful. In this paper, we propose a traitor tracing scheme in which the encryption key for broadcasting can be published and our scheme has optimal transmission rate. In another word, while transmitting the digital contents, our scheme can encrypt nearly without any redundancy. As for tracing, our scheme can support black-box tracing, i.e., knowing the legitimate subscribers who leak their decryption keys out without opening the pirate decoder to check the decryption inside. Moreover,

comparing to the previous schemes, the storage requirements for legitimate subscribers and digital content broadcasters can be smaller.

Keywords: traitor tracing, fingerprinting codes, all-or-nothing transform



誌謝

首先感謝我的指導老師曾文貴教授，在我碩士班兩年的學習過程中，帶領我深入密碼學的領域，老師認真積極的教學態度，使我受益良多。另外，我要感謝口試委員，中研院呂及人教授、交大謝續平教授與交大蔡錫鈞教授，在論文上給我許多建議與指導，讓我的論文更加完善。除此之外，我也要感謝博士班學長朱成康、學姊林孝盈在研究上給我我很多實質上的幫助，也感謝碩士班的同學們以及學弟們讓我的碩士班生活充滿歡樂。最後，我要感謝我的家人及身旁的朋友們，不論在精神或物質上都給我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，謹以此文獻給所有我想要感謝的人。

Contents

| | |
|---|------|
| Abstract in Chinese | i |
| Abstract | ii |
| Acknowledgement | iv |
| Contents | v |
| List of Figures | vii |
| List of Tables | viii |
| 1 Introduction | 1 |
| 2 Preliminaries | 7 |
| 3 Our Construction | 15 |
| 3.1 The Framework of Our Scheme | 15 |



| | | |
|----------|--|-----------|
| 3.2 | Basic Traitor Tracing Scheme for Two Users | 18 |
| 3.3 | Our Traitor Tracing Scheme for N Users | 22 |
| 3.4 | Security Analysis of Our TTS-AONT | 25 |
| 4 | Conclusion | 31 |
| | Bibliography | 33 |
| A | All-Or-Nothing Transform | 38 |
| B | The Public-Key Cryptosystem with AONT | 40 |
| C | The Traitor Tracing Scheme | 43 |



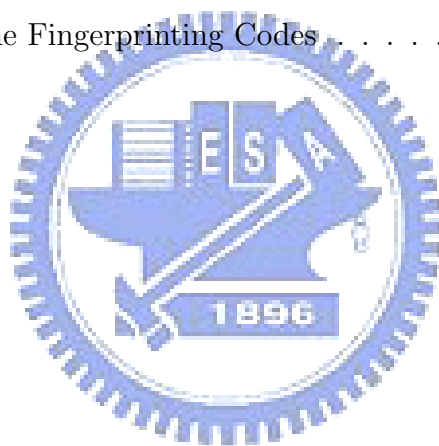
List of Figures

3.1 Tracing Part 18



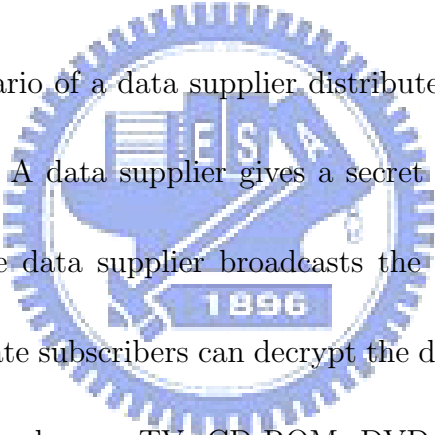
List of Tables

| | | |
|-----|--|----|
| 1.1 | Scheme Comparison | 5 |
| 2.1 | Length of the Fingerprinting Codes | 12 |



Chapter 1

Introduction



Considering the scenario of a data supplier distributes digital contents over a broadcast channel. A data supplier gives a secret key to each legitimate subscriber. Then the data supplier broadcasts the encrypted digital contents and the legitimate subscribers can decrypt the digital contents by their secret keys. For example, pay-TV, CD-ROM, DVD, and online databases are based on this scenario. However, some malicious legitimate subscribers (called *traitors*) might give the copies of their secret keys to the illegitimate subscribers (called *pirates*). Then the pirates can decrypt the digital contents for free. In order to solve the problem above, the *traitor tracing* scheme comes up.

The traitor tracing scheme was first introduced by Chor, Fiat and Naor in

[10, 11]. Its goal is to discourage legitimate subscribers from giving away their secret keys. The approach gives each subscriber a unique set of secret keys that can both decrypt the encrypted digital contents and identify ("trace") the subscribers. The traitors may collude to obfuscate their secret keys and trying to generate a new secret key set (called *pirate key*) that can still decrypt the encrypted digital contents but cannot be traced. We call a traitor tracing scheme *t-collusion resistant* if at least one of the traitors can always be identified when t traitors collude to generate a pirate key in this way. If t can reach the number of total legitimate users, the traitor tracing scheme will be called *fully-collusion resistant*. Note that the traitors may embed the pirate keys into a "tamper-resistant" hardware (called *pirate decoder*) to prevent the data supplier read any data inside. So, during the tracing, the data supplier has to treat the pirate decoder as a *black box* – it suffices to capture one pirate decoder and assumes that only the outcome of a pirate decoder can be examined.

In many approaches, the overhead of broadcasting the encrypted digital contents is proportional to the number of legitimate subscribers. But in some applications, such as pay-TV, the number of legitimate subscribers might be

upto millions. This will be a great burden for the data suppliers to broadcast the encrypted digital contents. The approach of *public-key traitor tracing* schemes proposed in Kurosawa and Desmedt [15], and Boneh and Franklin [3] eliminated this problem: it enables anyone (e.g. pay-TV stations) to broadcast the encrypted digital contents. Considering that there might be possible large number of pirate decoders, a bottleneck may appear if only the data supplier is able to run the tracing procedure. Thus, in [9], Chabanne, Phan, and Pointcheval first considered the concept of *public traceability* as an important estimate of the traitor tracing schemes. In order to measure the efficiency of the traitor tracing schemes, we consider the "transmission rate" of encrypted digital contents ("ciphertexts"), that is, the ratio of the size of ciphertext to the size of the digital contents. We also care about the storage requirements of subscribers' secret keys, and the broadcast keys.

Related work. The *traitor tracing* scheme was first introduced by Chor, Fiat and Naor in [10, 11], and was later to be refined in [16]. The concept of *public-key traitor tracing* schemes was proposed in Kurosawa and Desmedt [15], and Boneh and Franklin [3]. The traitor tracing schemes mentioned

[3, 4, 9, 13, 12, 15, 14, 17, 19, 22, 23] belong to this class. In this paper, we focus on the public-key traitor tracing schemes, and our scheme also belongs to this class. In [9], Chabanne, Phan, and Pointcheval first proposed the concept of *public traceability*. A class of traitor tracing schemes relying on the usage of *fingerprinting codes* [6, 21] was introduced by Kiayias and Yung in [14] – combining the *fingerprinting codes* defined by Boneh and Shaw [6] with the public-key traitor tracing schemes. Kiayias and Yung [14] showed that if the plaintexts to be distributed are large (e.g. multimedia contents), then it is possible to obtain constant transmission rate. For example, [9, 19, 18, 12] (including ours) belong to this class. When considering the transmission rate, we have two main categories in the traitor tracing schemes:

- Schemes with *no constant transmission rate* [3, 5]: These schemes are well-suited to encrypt small digital contents (usually using for the session-key exchanges in the "hybrid encryption"). The user-key size and the public-key size are often relatively small in these schemes. But the transmission rate in these schemes is often linear or sublinear to the maximal number of colluders.

| | transmission rate | user-key size | public-key size | black-box tracing | traceability |
|-------------|----------------------|------------------|--------------------|----------------------|--------------|
| BF99 [3] | $2t + 1$ | $2t$ | $2t + 1$ | X | private |
| BSW06 [5] | $6\sqrt{N}$ | 1 | $4\sqrt{N} + 2$ | O | public |
| KY02 [14] | ~ 3 | 2ℓ | 4ℓ | O | private |
| CPP05 [9] | ~ 1 | 2ℓ | $\ell + 1$ | X | private |
| FNP07 [12] | ~ 1 | 2ℓ | 10ℓ | O | private |
| Ours | ~ 1 | $\ell + 1$ | $\ell + 2$ | O | private |

[†] ℓ : the codeword length in fingerprinting code

[†] N : the total number of legitimate subscribers

Table 1.1: Scheme Comparison

- Schemes with *constant transmission rate* [14, 9, 12] (including ours):

These schemes are well-suited to encrypt large digital contents (e.g. multimedia contents). These schemes are all constructed by using the fingerprinting codes. The advantage in these schemes is that they often have the efficient black-box tracing algorithms. But the user-key size and the public-key size are often relatively large (according to the codeword length in the fingerprinting codes) until now.

We give a comparison of these traitor tracing schemes in Table 1.

Our Contributions. We propose a framework of public-key traitor tracing schemes with efficient black-box tracing which has optimal transmission

rate. Our framework is based on the usage of fingerprinting codes, and the *all-or-nothing transformation* defined by Rivest in [20] (and refined by [7, 8]). In order to achieve the optimal transmission rate, we mainly use the cryptosystem (PKE-AONT) proposed in [24] that is semantically secure under the random oracle model.

Using this framework, we actually construct a traitor tracing scheme which has less storage requirements than previous schemes (see Table 1). Then we show that our traitor tracing scheme is semantically secure based on the DDH assumption and the semantic security of PKE-AONT. Finally, we show that our traitor tracing scheme is t -collusion resistant under the DDH assumption.



Chapter 2

Preliminaries

Notations. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is called *negligible* if for every constant $c \in \mathbb{N}$, there exists an integer $k_0 \in \mathbb{N}$ such that $f(k) \leq k^{-c}$ for all $k \geq k_0$, denoted by $neg(k)$. We use $x \xleftarrow{\$} X$ to denote that we choose x from the set X uniformly, and $x \leftarrow X$ to denote that we set x to the output of X . Let \mathcal{M} be the plaintext set.

Traitor Tracing Scheme. A traitor tracing scheme consists of four algorithms: Setup, Encrypt, Decrypt, and Trace. The Setup algorithm generates the system parameters such as the broadcast-key BK, the trace-key TK, and the user-key SK_i for user i . The Encrypt algorithm encrypts the plaintext to the ciphertext by BK, then user i decrypts the ciphertext to the plaintext

by taking SK_i and the ciphertext as the inputs of **Decrypt** algorithm. The most interesting one – **Trace** algorithm: by taking TK as an input and the black-box access with a pirate decoder, it outputs at least one of the traitors' keys using in the pirate decoder. We follow the definition of the secure games of a traitor tracing scheme by Boneh, Sahai, and Waters in [5] as follows:

Semantic Secure Game:

- *Setup.* The challenger runs **Setup**, then it gives BK to the adversary.
- *Challenge.* The adversary chooses two plaintexts $M_0, M_1 \in \mathcal{M}$ to the challenger. Then the challenger flips a coin $b \in \{0, 1\}$, and gives a ciphertext $C_b \xleftarrow{\$} \text{Encrypt}(\text{BK}, M_b)$ to the adversary.
- *Guess.* The adversary returns a guess $b' \in \{0, 1\}$ of b to the challenger.

The advantage of the adversary wins this game is $\text{Adv}_{\text{SS}}^{\text{TTS}} := |\Pr[b' = b] - \frac{1}{2}|$.

Traceable against t -collusion Game:

- *Setup.* The adversary chooses a traitor set $\mathsf{T} = \{u_1, \dots, u_t\} \subseteq \{1, \dots, N\}$ to the challenger. Then the challenger runs **Setup**, and gives BK and $\mathsf{SK}_{u_1}, \dots, \mathsf{SK}_{u_t}$ to the adversary.
- *Trace.* The adversary produces a pirate decoder \mathcal{D} . Then the challenger runs the algorithm $\mathsf{Trace}^{\mathcal{D}}(\mathsf{TK}, \delta)$ to obtain a traitor set $\mathsf{S} \subseteq \{1, \dots, N\}$.

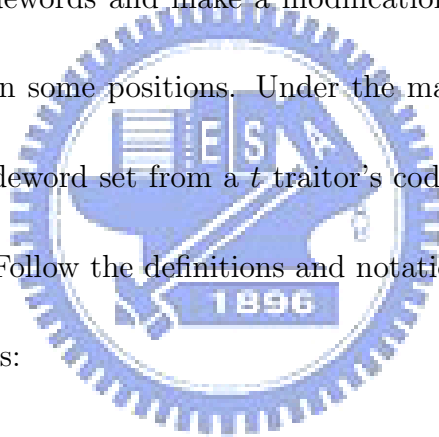
The adversary wins this game if (1) \mathcal{D} is δ -useful: \mathcal{D} can decrypt all valid ciphertext with probability δ , i.e., $\Pr[\mathcal{D}(\mathsf{Encrypt}(\mathsf{BK}, M)) = M] \geq \delta$, and (2) the set $\mathsf{S} = \phi$ or $\mathsf{S} \not\subseteq \mathsf{T}$.

The probability of adversary wins this game is $\mathsf{Adv}_{\mathsf{TR}}^{\mathsf{TTS}}$.

Definition 1. An N -user traitor tracing scheme is semantic secure if for all polynomial time adversaries \mathcal{A} , $\mathsf{Adv}_{\mathsf{SS}}^{\mathsf{TTS}}$ is a negligible function of the security parameter.

Definition 2. An N -user traitor tracing scheme is traceable against t -collusion if for all polynomial time adversaries \mathcal{A} of corrupting t users and any constant $\delta > 0$, $\mathsf{Adv}_{\mathsf{TR}}^{\mathsf{TTS}}$ is a negligible function of the security parameter.

Fingerprinting Codes. Fingerprinting (a cryptographic technique) with fingerprinting codes allows identifying a digital document among several copies by embedding a fingerprint (a *codeword*). The codeword is a collection of some alphabets. The traitors will collude and try to modify their codewords to prevent the identifications. However, the coalitions of the traitors are restricted by the *marking assumption*: the traitors are only able to compare their codewords and make a modification from their respective codewords differing in some positions. Under the marking assumption, the possible modified codeword set from a t traitor's codewords set W is called a *feasible set* of W . Follow the definitions and notations in [4], we illustrate the concept as follows:



- For a codeword $w \in \{0, 1\}^\ell$, we write $w = w_1 w_2 \dots w_\ell$, where $w_i \in \{0, 1\}$.
- Let $W = \{w^{(1)}, \dots, w^{(t)}\} \subseteq \{0, 1\}^\ell$. We say that a codeword \bar{w} is *feasible* for W if: $\forall i \in \{1, 2, \dots, \ell\} \exists j \in \{1, 2, \dots, t\}$ s.t. $\bar{w}_i = w_i^{(j)}$. For example, if $W = \{01011, 11101\}$, then the codewords $\binom{0}{1} 1 \binom{0}{1} \binom{0}{1} 1$ are feasible for W .
- For a codeword set $W \subseteq \{0, 1\}^\ell$, we say that the *feasible set* of W ,

denoted $F(W)$, is the set of all codewords that are feasible for W .

A fingerprinting code scheme consists of two algorithms: *codeword generation* algorithm G and *codeword tracing* algorithm T . The G algorithm generates codeword set $\{w^{(1)}, \dots, w^{(N)}\} \in (\{0, 1\}^\ell)^N$ (for some $\ell > 0$) and the trace-key tk . By taking an pirate codeword \bar{w} and tk as inputs, the T algorithm outputs at least one of the traitors that are collude to generate \bar{w} . We define the secure game of a fingerprinting code scheme as follows:

t-collusion Secure Game

- *Setup.* The adversary chooses a traitor set $T = \{u_1, \dots, u_t\} \subseteq \{1, \dots, N\}$ to the challenger. Then the challenger runs G , and gives $w^{(u_1)}, \dots, w^{(u_t)}$ to the adversary.
- *Trace.* The adversary produces a pirate codeword \bar{w} . Then the challenger runs the algorithm $T(\bar{w}, tk)$ to obtain a traitor set $S \subseteq \{1, \dots, N\}$.

The adversary wins this game if $S = \emptyset$ or $S \not\subseteq T$. The probability of the adversary wins this game is $\text{Adv}_{\text{CS}}^{\text{FC}}$.

| | t -collusion resistant | fully-collusion resistant |
|----------|---|----------------------------------|
| BS98 [6] | $\ell = O(t^4 \log(n/\epsilon) \log(1/\epsilon))$ | $\ell = O(n^3 \log(n/\epsilon))$ |
| T03 [21] | $\ell = O(t^2 \log(n/\epsilon))$ | $\ell = O(n^2 \log(n/\epsilon))$ |

Table 2.1: Length of the Fingerprinting Codes

Definition 3. A fingerprinting code scheme is t -collusion secure if for all polynomial time adversary \mathcal{A} of corrupting t users, $\text{Adv}_{\text{CS}}^{\text{FC}}$ is a negligible function of the security parameter.

Boneh and Shaw [6] constructed a fully-collusion resistant fingerprinting code as well as t -collusion resistant secure codes. Tardos [21] proposed a shorter codes. We give a comparison of their codeword lengths in Table 2 (n is the number of codewords, and ϵ is the security parameter).

All-Or-Nothing Transform Function. The concept of all-or-nothing transform functions was proposed in [20]. An all-or-nothing transform function is an efficient, unkeyed, and randomized function with the property that it is hard to invert unless the entire output is known. Boyko [7] defined the semantic security and indistinguishability of the all-or-nothing transform functions against adaptive and non-adaptive attacks. Then Boyko [7] also

proved that OAEP [1] is a secure implementation of all-or-nothing transform functions in the random oracle model. Simultaneously, Boyko [7] showed that the upper bounds of semantic security and indistinguishability against passive and adaptive attacks.

An all-or-nothing transform function AONT can mapping an ℓ' -block sequence x with a random string ρ to an ℓ -block sequence y with the following properties:

- Given x and ρ , $y \stackrel{\$}{\leftarrow} \text{AONT}(x; \rho)$ can be computed efficiently.
- Given all blocks of y , $x \leftarrow \text{AONT}^{-1}(y)$ can be computed efficiently.
- It is infeasible to get any information of any blocks of x if any of the blocks of y is missing.

Notice that the usage of all-or-nothing transform functions make an expansion in the plaintext size by roughly $1 + 1/\ell$, which still results in an asymptotical unitary ciphertext-to-plaintext ratio.

Decision Deffie-Hellman (DDH) Assumption. For a cyclic group \mathbb{G} with a generator g . Let \mathcal{V} be the distribution $\{(g, g^u, g^v, g^{uv})\}$ and \mathcal{R} be the

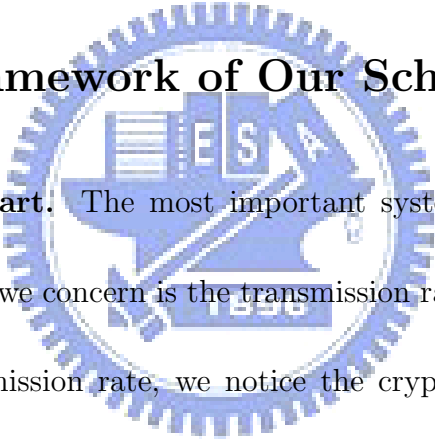
distribution $\{(g, g^u, g^v, g^w)\}$. For any polynomial time adversary \mathcal{A} , \mathcal{A} can distinguish the two distributions \mathcal{V} and \mathcal{R} with negligible function of λ , i.e.,

$$|\Pr[\mathcal{A}(X) = 1 : X \in \mathcal{V}] - \Pr[\mathcal{A}(X) = 1 : X \in \mathcal{R}]| = \text{neg}(|\mathbb{G}|).$$


Chapter 3

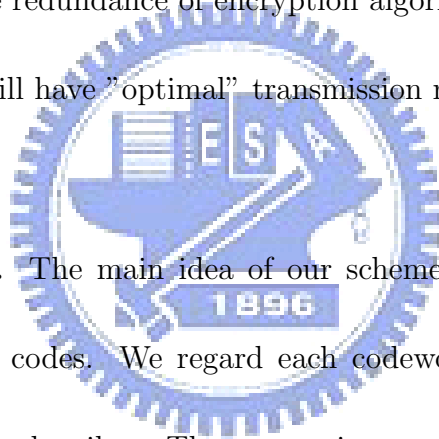
Our Construction

3.1 The Framework of Our Scheme



The Encryption Part. The most important system measure in traitor tracing schemes that we concern is the transmission rate. In order to achieve the "optimal" transmission rate, we notice the cryptosystem proposed by Zhang, Hanaoka, and Imai [24]: it encrypts some bits of the outputs of all-or-nothing transform functions by a public-key encryption scheme. Given a public-key encryption scheme $\text{PKE} = (\text{G}, \text{E}, \text{D})$ and a all-or-nothing transform function AONT , the encryption scheme first runs the algorithm G to generate a public-key and secret-key pair (pk, sk) . Then it inputs the plaintext M' and a random bit string r to AONT to get $M = m_1 || m_2 || \dots || m_\ell$.

Finally, the encryption scheme randomly chooses k -th block of M and encrypts it. When decrypting the ciphertext C , the decryption algorithm first decrypts the k -th block by sk to recover M . Then it inputs M to the inverse of the all-or-nothing transform function to get M' . Notice that the usage of all-or-nothing transform functions make an expansion in the plaintext size by roughly $1 + 1/\ell$, which guarantees that $|M'|/|M| \sim 1$ when ℓ is large. And if the size of the redundancy of encryption algorithm E is constant, the encryption scheme will have "optimal" transmission rate.



The Tracing Part. The main idea of our scheme for tracing is the usage of fingerprinting codes. We regard each codeword in a fingerprinting code as a legitimate subscriber. Then we assign an unique user-key set for each legitimate subscriber according to each codeword. The traitors may collude to create a new user-key set (can decrypt the broadcast digital contents success) and embed them in a pirate decoder. Then we have to recover the corresponding codewords (called *pirate codeword*) by identifying the user-key set using in the pirate decoders. We construct our traitor tracing by the following steps:

- Construct a "basic" 1-collusion resistant public-key traitor tracing scheme for two users, called 2-PK-TTS.
- Construct a fingerprinting code (fully-collusion or t -collusion resistant) of size N over $\{0, 1\}$: $\Gamma = \{w^{(1)}, w^{(2)}, \dots, w^{(N)}\} \subseteq \{0, 1\}^\ell$, for some $\ell > 0$.
- Construct ℓ components of 2-PK-TTS. We demonstrate such construction in Figure 3.1. For each codeword $w^{(i)} = w_1^{(i)} w_2^{(i)} \dots w_\ell^{(i)}$, we assign $sk_{j,0}$ to legitimate subscriber i if $w_j^{(i)} = 0$; else assign $sk_{j,1}$, $\forall j \in \{1, 2, \dots, \ell\}$. For example, let $\ell = 3$, if the codeword corresponding to legitimate subscriber u is $(0, 1, 1)$, then its user-key set is $(sk_{1,0}, sk_{2,1}, sk_{3,1})$.
- We replace the public-key encryption scheme PKE above by 2-PK-TTS.
- When we do the tracing procedures, we can use the i -th 2-PK-TTS to identify the i -th symbol of the pirate codeword, for all $i \in \{1, 2, \dots, \ell\}$. Finally, by the tracing algorithm in the fingerprinting code, we can find the collusion codeword set for constructing a pirate codeword, i.e., we can find the collusion traitor set.

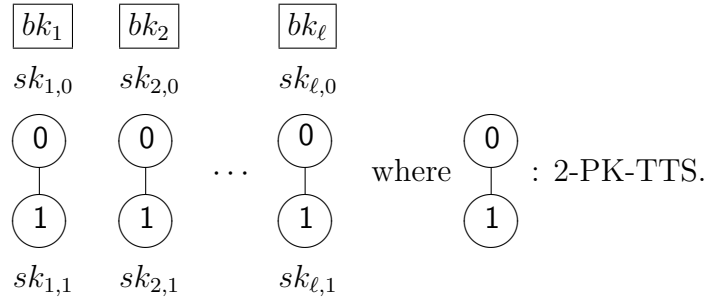


Figure 3.1: Tracing Part

In section 3.2, we construct the "basic" scheme 2-PK-TTS. In section 3.3, we use 2-PK-TTS and the fingerprinting codes to construct our traitor tracing scheme TTS-ANOT for N legitimate subscribers. Finally, in section 3.4, we give the security proofs for TTS-ANOT.

3.2 Basic Traitor Tracing Scheme for Two Users

We modify the traitor tracing scheme in [23] as our basic scheme for two users: $2\text{-PK-TTS} = (2\text{-Setup}, 2\text{-Encrypt}, 2\text{-Decrypt}, 2\text{-Trace})$, where

2-Setup: Given a security parameter λ , the algorithm generates a λ -bit prime

q , a group \mathbb{G} of order q , and a generator g of \mathbb{G} . Then the algorithm

chooses $f(x) = a_0 + a_1x \pmod{q}$, where $a_0, a_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ and sets

- Public broadcast-key $bk := \langle g, (g^{a_0}, g^{a_1}) \rangle$

- Secret trace-key $tk := \langle f(x) \rangle$
- User-key $sk_\sigma := \langle i_\sigma, f(i_\sigma) \rangle$, where $i_\sigma \in \mathbb{Z}_q^*$, $\forall \sigma \in \{0, 1\}$

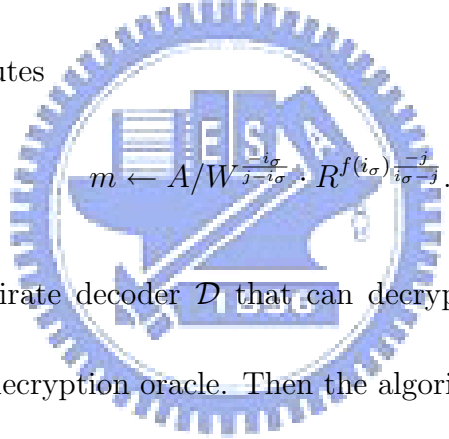
2-Encrypt: Given bk and a plaintext $m \in \mathcal{M}$, the algorithm chooses an

unused share $j \xleftarrow{\$} \mathbb{Z}_q^*$ and $r \xleftarrow{\$} \mathbb{Z}_q^*$ then outputs the ciphertext

$$c \leftarrow \langle mg^{ra_0}, g^r, (j, g^{rf(j)}) \rangle.$$

2-Decrypt: Given a ciphertext $c = \langle A, R, (j, W) \rangle$ and user-key sk_σ , the

algorithm computes

$$m \leftarrow A/W^{\frac{-i_\sigma}{j-i_\sigma}} \cdot R^{f(i_\sigma) \cdot \frac{-j}{i_\sigma-j}}.$$


2-Trace: Given a pirate decoder \mathcal{D} that can decrypt all valid ciphertext

perfectly as a decryption oracle. Then the algorithm does:

1. 2-TrEncrypt: Given bk and a plaintext $m \xleftarrow{\$} \mathcal{M}$, the algorithm chooses the distinct $r, \hat{r} \xleftarrow{\$} \mathbb{Z}_q^*$, and an unused share $j \xleftarrow{\$} \mathbb{Z}_q^*$.

Then it computes a probe ciphertext

$$\hat{c} \xleftarrow{\$} \langle A = mg^{a_0}, R = g^r, (j, \hat{W} = g^{\hat{r}f(j)}) \rangle.$$

2. For all $\sigma \in \{0, 1\}$, pre-compute

$$V_\sigma = \hat{W}^{\frac{-i_\sigma}{j-i_\sigma}} \cdot R^{f(i_\sigma) \cdot \frac{-j}{i_\sigma-j}}.$$

3. For all $\sigma \in \{0, 1\}$, if $\mathcal{D}(\hat{c}) = A/V_\sigma$, then output $S = \{\sigma\}$; else output $S = \{0, 1\}$.

Theorem 1. 2-PK-TTS is semantic secure under the DDH assumption.

Proof. It is a special case of the traitor tracing scheme describe in [23]. \square

Theorem 2. 2-PK-TTS is traceable against 1-collusion under the DDH assumption.

Proof. By contradiction, assume that there exists an adversary \mathcal{A} that given the public key and one of user-key in 2-TTS, \mathcal{A} can produce a pirate decoder \mathcal{D} that can decrypt all valid ciphertexts perfectly, i.e., $\Pr[\mathcal{D}(\text{2-Encrypt}(bk, m)) = m : \mathcal{D} \xleftarrow{\$} \mathcal{A}(bk, sk_\sigma), \sigma \in \{0, 1\}] = 1$. But when given a probe ciphertext \hat{c} to \mathcal{D} , where $\hat{c} \xleftarrow{\$} \langle A, g^r, (j, \hat{W} = g^{\hat{r}f(j)}) \rangle$, it can output a different value than A/V_σ in 2-Trace algorithm of 2-TTS with non-negligible probabilistic $\epsilon > 0$, i.e.,

$$\Pr[\mathcal{D}(\hat{c}) \neq A/V_\sigma : V_\sigma \leftarrow \hat{W}_{j-i\sigma}^{-i\sigma} \cdot R^{f(i\sigma) \cdot \frac{-j}{i\sigma-j}}, \forall \sigma \in \{0, 1\}] = \epsilon$$

Then we can construct an algorithm \mathcal{B} that can break the DDH assumption with non-negligible advantage $\frac{\epsilon}{4}$ as follows:

- *Setup.* Algorithm \mathcal{B} is given as input an instance (g, g^u, g^v, X) of DDH assumption, and it wants to determine whether $X = g^{uv}$ or X is a random element in \mathbb{G} . \mathcal{B} chooses $i, z \xleftarrow{\$} \mathbb{Z}_q^*$ and sets $sk = \langle i, z \rangle$, $bk = \langle g, (g^u, g^{a_1} = (\frac{g^z}{g^u})^{i-1}) \rangle$, then gives (bk, sk) to \mathcal{A} .
- *Trace.* Adversary produces a pirate decoder \mathcal{D} that has the property above. Then \mathcal{B} runs the modified **2-Trace** as follows:

1. Choose $A \xleftarrow{\$} \mathbb{G}$, and $j \xleftarrow{\$} \mathbb{Z}_q^*$, where $j \neq i$. Set the ciphertext as

$$c \leftarrow \langle A, g^v, (j, W = X(\frac{(g^v)^z}{X})^{ji^{-1}}) \rangle.$$

2. Pre-compute $V \leftarrow W^{\frac{i}{j-i}} \cdot (g^v)^{z \cdot \frac{-j}{i-j}}$.
3. If $\mathcal{D}(c) = A/V$, then \mathcal{B} answers that $X = g^{uv}$ or X is a random element in \mathbb{G} randomly; else \mathcal{B} answers that X is a random element in \mathbb{G} .

If $X = g^{uv}$, then ciphertext c is a valid ciphertext, since

$$X(\frac{(g^v)^z}{X})^{ji^{-1}} = g^{uv}(\frac{(g^v)^z}{g^{uv}})^{ji^{-1}} = g^{uv}((\frac{g^z}{g^u})^{i-1})^{vj} = g^{uv}(g^{a_1})^{vj} = g^{v(u+a_1j)}.$$

In this case, $\mathcal{D}(c) = A/V$, therefore \mathcal{B} only can give the correct answer with probability $\frac{1}{2}$;

If X is a random element in \mathbb{G} , then ciphertext c is a non-valid ciphertext. In this case, $\mathcal{D}(c) \neq A/V$ with probability ϵ , and $\mathcal{D}(c) = A/V$ with probability $1 - \epsilon$, therefore \mathcal{B} can give the correct answer with probability $\epsilon + \frac{1}{2}(1 - \epsilon) = \frac{1}{2} + \frac{\epsilon}{2}$.

Hence, \mathcal{B} can solve DDH problem with non-negligible advantage $\frac{\epsilon}{4}$, this is a contradiction to the DDH assumption, so we can conclude that such adversary \mathcal{A} does not exist. \square

3.3 Our Traitor Tracing Scheme for N Users

Our traitor tracing scheme for N users follows the framework in previously section but replace the public-key encryption scheme PKE by 2-PK-TTS.

For convenience, we introduce some notations we use in our scheme:

- $\text{MINUS}_k(M)$: Given an $\ell\lambda$ -bit message $M = m_1 || \dots || m_\ell$ and a position index $k \in \{1, \dots, \ell\}$, the algorithm outputs M "minus" k -th block of M of size λ , i.e., $\text{MINUS}_k(M) = m_1 || \dots || m_{k-1} || m_{k+1} || \dots || m_\ell$.
- $\text{COMB}_k(Y, m)$: Given an $(\ell - 1)\lambda$ -bit message $Y = y_1 || \dots || y_{\ell-1}$, λ -bit message m and a position index $k \in \{1, \dots, \ell - 1\}$, the algorithm

first split Y to $X_1||X_2$, where X_1 is the front $(k-1)\lambda$ bits of Y and X_2 is the rest bits of Y . Then the algorithm "combines" and output the messages with order X_1 , m , and X_2 , i.e. $\text{COMB}_k(Y, m) = y_1||\dots||y_{k-1}||m||y_k||\dots||y_{\ell-1}$.

Our traitor tracing scheme for N users TTS-AONT = (Setup, Encrypt, Decrypt, Trace)

Setup: Given a security parameter λ and user number N , the algorithm

generates a fingerprinting code $\Gamma = \{w^{(1)}, \dots, w^{(N)}\}$ over $\{0, 1\}^\ell$ (can be public). Then it runs 2-Setup ℓ times to generate the keys $\langle (bk_i, tk_i, (sk_{0,i}, sk_{1,i}))_{i=1}^N \rangle$ (but use the same $q, \mathbb{G}, q, i_0, i_1, a_0$). Finally the algorithm picks a randomized all-or-nothing transform-function AONT and sets

- Public broadcast-key $\text{BK} := \langle q, g, g^{a_0}, (g^{a_{1,j}})_{j=1}^\ell, \text{AONT} \rangle$
 (we denote the k -th key of BK by $\text{BK}_k = (q, g, g^{a_0}, g^{a_{1,k}})$)
- Secret trace-key $\text{TK} := \langle (f_j(x))_{j=1}^\ell \rangle$
- User-key $\text{SK}_\sigma := \langle w^{(\sigma)}, i_0, i_1, (f_j(i_{w_j^{(\sigma)}}))_{j=1}^\ell \rangle, \forall \sigma \in \{1, 2, \dots, N\}$
 (we denote k -th key of SK_σ by $\text{SK}_{\sigma,k} = (i_{w_k^{(\sigma)}}, f_k(i_{w_k^{(\sigma)}}))$)

Encrypt: Given BK and a plaintext $M' \in \mathcal{M}^\ell$, the algorithm chooses a

random string $\rho \xleftarrow{\$} \{0, 1\}^\tau$, and computes $M \xleftarrow{\$} \text{AONT}(M'; \rho)$. Then

it chooses a position $k \xleftarrow{\$} \{1, 2, \dots, \ell\}$, and computes the ciphertext

$$C \xleftarrow{\$} \langle k, 2\text{-Encrypt}(\text{BK}_k, m_k), \text{MINUS}_k(M) \rangle$$

Decrypt: Given a ciphertext $C = \langle k, c_k, Y \rangle$, user σ computes

$$M' \leftarrow \text{AONT}^{-1}(\text{COMB}_k(Y, m_k)), \text{ where } m_k \leftarrow 2\text{-Decrypt}(\text{SK}_{\sigma, k}, c_k)$$

Trace: Given a pirate decoder \mathcal{D} that can decrypt all valid ciphertext perfectly as a decryption oracle.

- For all position $k \in \{1, 2, \dots, \ell\}$, do:

- (1) Choose an $M' \xleftarrow{\$} \mathcal{M}^\ell$, a random string $\rho \xleftarrow{\$} \{0, 1\}^\tau$, and run

$$\text{AONT}(M'; \rho) = M = m_1 || m_2 || \dots || m_\ell.$$

- (2) Call $2\text{-TrEncrypt}(\text{BK}_k, m_k) \xrightarrow{\$} \hat{c}$. Set the probe ciphertext as

$$\hat{C} \xleftarrow{\$} \langle k, \hat{c}, Y = \text{MINUS}_k(M) \rangle$$

- (3) $\forall \sigma \in \{0, 1\}$, pre-compute

$$M_{k, \sigma} = \text{COMB}_k(Y, \hat{W}_{j-i\sigma}^{-i\sigma} \cdot R^{f_k(i\sigma) \cdot \frac{-j}{i\sigma-j}}).$$

- (4) $\forall \sigma \in \{0, 1\}$, if $\text{AONT}(\mathcal{D}(\hat{C}); \rho) = M_{k, \sigma}$, then set $w_k^* = \sigma$.

- Recover $w^* = (w_1^*, w_2^*, \dots, w_\ell^*)$, then call the tracing algorithm in fingerprinting code to obtain collude codewords in \mathcal{C} . Finally, output the corresponding traitor set S .

3.4 Security Analysis of Our TTS-AONT

Theorem 3. TTS-AONT is semantic secure under the semantic secure of 2-PK-TTS and PKE-AONT.

Proof. For all position $k \in \{1, 2, \dots, \ell\}$, we use two games to bound the advantage of semantically secure in TTS-AONT with $\text{Adv}_{\text{SS}}^{2\text{-TTS}}$ and $\text{Adv}_{\text{ind}}^{\text{PKE-AONT}}$ as follows:

Game \mathbf{G}_0 . Define \mathbf{G}_0 as the original semantic secure game and let S_0 be the event where $b' = b$, i.e., $\text{Adv}_{\text{SS}}^{\text{TTS-AONT}} := |\Pr[S_0] - \frac{1}{2}|$.

Game \mathbf{G}_1 . This game is identical to \mathbf{G}_0 , except that in the Encrypt

$$c_1 \stackrel{\$}{\leftarrow} \langle A \stackrel{\$}{\leftarrow} \{0, 1\}^\lambda, R = g^r, (j, W = g^{rf_k(j)}) \rangle$$

and we let S_1 be the event that $b' = b$ in this game.

Claim: $|\Pr[S_0] - \Pr[S_1]| \leq 2\text{Adv}_{\text{SS}}^{2\text{-TTS}}, \forall k \in \{1, 2, \dots, \ell\}$.

By reduction, for all position $k \in \{1, 2, \dots, \ell\}$, if there exists an adversary \mathcal{A} that can distinguish the challenge of \mathbf{G}_0 and \mathbf{G}_1 with non-negligible probability $\epsilon > 0$, then we can use \mathcal{A} to construct an adversary \mathcal{B} that can break the semantic security of 2-TTS with non-negligible advantage $\frac{\epsilon}{2}$ as follows:

- *Setup.* Algorithm \mathcal{B} is given as input an instance $bk = \langle g, g^{a_0}, g^{a_1} \rangle$ of 2-TTS, and it wants to determine whether the challenge C is constructed by \mathbf{G}_0 or \mathbf{G}_1 . \mathcal{B} chooses $a_{\alpha,1} \xleftarrow{\$} \mathbb{Z}_q^*$, where $\alpha \in \{1, 2, \dots, \ell\} \setminus \{k\}$ and let $g^{a_{k,1}} = g^{a_1}$, then sets $BK = \langle g, g^{a_0}, (g^{a_{\alpha,1}})_{\alpha=1}^{\ell} \rangle$ to \mathcal{A} .
- *Challenge.* \mathcal{A} chooses two plaintexts $M_0, M_1 \in \mathcal{M}^{\ell}$ to \mathcal{B} , then \mathcal{B} flips a coin $b' \in \{0, 1\}$, and it calls $\text{AONT}(M_{b'}; \rho) = m_{b',1} || m_{b',2} || \dots || m_{b',\ell}$, lets $m_{1-b',k} \xleftarrow{\$} \{0, 1\}^{\lambda}$, and sends $m_{b'} = m_{b',k}, m_{1-b'} = m_{1-b',k}$ to 2-TTS challenger. Then 2-TTS challenger flips a coin $b \in \{0, 1\}$ and sets the challenge $c_b \xleftarrow{\$} \text{2-Encrypt}(bk, m_b)$ to \mathcal{B} . Finally, \mathcal{B} sends \mathcal{A} the challenge

$$C_{b'} = \langle k, c_b, Y = m_{b',1} || \dots || m_{b',k-1} || m_{b',k+1} || \dots || m_{b',\ell} \rangle.$$

- *Guess.* \mathcal{A} outputs $\hat{b} \in \{0, 1\}$ to \mathcal{B} . Then \mathcal{B} gives \hat{b} as its guess to 2-TTS challenger.

By construction above, we can see that \mathcal{B} 'interpolate' between \mathbf{G}_0 and \mathbf{G}_1

for \mathcal{A} :

- If $b' = b$, then \mathcal{A} gets a challenge in \mathbf{G}_0 ;
- If $b' = 1 - b$, then \mathcal{A} gets a challenge in \mathbf{G}_1 .

Thus, it holds that $\Pr[S_0] = \Pr[\hat{b} = b' | b' = b]$ and $\Pr[S_1] = \Pr[\hat{b} = b' | b' = 1 - b]$, and we get

$$\begin{aligned}
 \Pr[\hat{b} = b] &= \Pr[\hat{b} = b | b' = b] \cdot \Pr[b' = b] + \Pr[\hat{b} = b | b' = 1 - b] \cdot \Pr[b' = 1 - b] \\
 &= \frac{1}{2} (\Pr[\hat{b} = b | b' = b] + \Pr[\hat{b} = b | b' = 1 - b]) \\
 &= \frac{1}{2} (\Pr[\hat{b} = b | b' = b] + 1 - \Pr[\hat{b} = 1 - b | b' = 1 - b]) \\
 &= \frac{1}{2} + \frac{1}{2} (\Pr[\hat{b} = b' | b' = b] - \Pr[\hat{b} = b' | b' = 1 - b]) \\
 &= \frac{1}{2} + \frac{1}{2} (\Pr[S_0] - \Pr[S_1])
 \end{aligned}$$

It follows that $|\Pr[S_0] - \Pr[S_1]| = 2|\Pr[\hat{b} = b] - \frac{1}{2}| = 2\text{Adv}_{\text{SS}}^{2\text{-TTS}}$, then we done the claim.

Since PKE-AONT is semantic secure, that is, the adversary can distinguish two ciphertexts in \mathbf{G}_1 with probability $\frac{1}{2} + \text{Adv}_{\text{ind}}^{\text{PKE-AONT}}$, where $\text{Adv}_{\text{ind}}^{\text{PKE-AONT}}$ is negligible function of λ .

Hence, by the discussion above and the triangle inequality,

$$\begin{aligned}
|\Pr[S_0] - \frac{1}{2}| &= |\Pr[S_0] - \Pr[S_1] + \Pr[S_1] - \frac{1}{2}| \\
&\leq |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1] - \frac{1}{2}| \\
&= 2\text{Adv}_{\text{SS}}^{2\text{-TTS}} + \text{Adv}_{\text{ind}}^{\text{PKE-AONT}},
\end{aligned}$$

and since $\text{Adv}_{\text{SS}}^{2\text{-TTS}}$ and $\text{Adv}_{\text{ind}}^{\text{PKE-AONT}}$ are two negligible functions of λ , so we can conclude that the advantage of \mathcal{A} wins the semantic secure game is bounded by a negligible function of λ . \square

Theorem 4. **TTS-AONT** is traceable against t -collusion under the DDH assumption.

Proof. By contradiction, assume that there exist an adversary \mathcal{A} that given the public key and one of user private keys and an all-or-nothing transform function in TTS-AONT, \mathcal{A} can produce a pirate decoder \mathcal{D} that can decrypt all valid ciphertexts perfectly, i.e., $\Pr[\mathcal{D}(\text{Encrypt}(\text{BK}, \text{AONT}, M')) = M' : \mathcal{D} \stackrel{\$}{\leftarrow} \mathcal{A}(\text{BK}, \text{SK}_\sigma), \sigma \in \{1, 2, \dots, N\}] = 1$. But when given a probe ciphertext to \mathcal{D} , it can output a different value than our expectation in Trace algorithm of TTS-AONT with non-negligible probabilistic $\epsilon > 0$, i.e.,

$$\Pr[\text{AONT}(\mathcal{D}(\hat{C}); \rho) \neq M'_{k,\sigma}] = \epsilon$$

Then we can construct an algorithm \mathcal{B} that can break the DDH assumption with advantage $\frac{\epsilon}{4}$ as follows:

- *Setup.* Algorithm \mathcal{B} is given as input an instance (g, g^u, g^v, X) of DDH assumption, and it wants to determine whether $X = g^{uv}$ or X is a random element in \mathbb{G} . \mathcal{B} chooses $k \xleftarrow{\$} \{1, 2, \dots, \ell\}$, $w \xleftarrow{\$} \{0, 1\}^\ell$, and $i_0, i_1, z_1, z_2, \dots, z_\ell \xleftarrow{\$} \mathbb{Z}_q^*$, then sets

$$\text{BK} = \langle g, g^u, (g^{a_{\alpha,1}} = \left(\frac{g^{z_\alpha}}{g^u}\right)^{(i_{w_\alpha})^{-1}})_{\alpha=1}^\ell \rangle$$

$$\text{SK} = \langle w, i_0, i_1, (z_\alpha)_{\alpha=1}^\ell \rangle, \text{ and simply denote } z_k = z \text{ and } i_{w_k} = i.$$

Then \mathcal{B} picks a randomized all-or-nothing transform function AONT, and it gives $(\text{BK}, \text{SK}, \text{AONT})$ to \mathcal{A} .

- *Trace.* Adversary produces a pirate decoder \mathcal{D} to \mathcal{B} that \mathcal{B} has the property above. Then \mathcal{B} runs the modified Trace algorithm as follows:

1. Compute $M \xleftarrow{\$} \text{AONT}(M'; \rho)$, where $M' \xleftarrow{\$} \mathcal{M}^{\ell'}$, $\rho \xleftarrow{\$} \{0, 1\}^\tau$.
2. Choose $j \xleftarrow{\$} \mathbb{Z}_q^*$, where $j \neq i_0$ or i_1 . Compute

$$c \leftarrow \langle A = m_k X, R = g^v, (j, W = X \left(\frac{g^v}{X}\right)^{ji^{-1}}) \rangle$$

and set the ciphertext as $C \leftarrow \langle k, c, Y = \text{MINUS}_k(M) \rangle$.

3. Pre-compute $M'_{k,\sigma} = \text{COMB}_k(\hat{W}_{\frac{-i\sigma}{j-i\sigma}} \cdot R^{f_k(i\sigma) \cdot \frac{-j}{i\sigma-j}})$.

4. $\forall \sigma \in \{0, 1\}$, if $\text{AONT}(\mathcal{D}(C); \rho) = M'_{k,\sigma}$, then \mathcal{B} answers $X = g^{uv}$ or X is a random element in \mathbb{G} randomly; else \mathcal{B} answers X is a random element in \mathbb{G} .

If $X = g^{uv}$, then ciphertext c is a valid ciphertext, since

$$X \left(\frac{(g^v)^z}{X} \right)^{ji^{-1}} = g^{uv} \left(\frac{(g^v)^z}{g^{uv}} \right)^{ji^{-1}} = g^{uv} \left(\frac{g^z}{g^u} \right)^{ji^{-1}} = g^{uv} (g^{a_{k,1}})^{vj} = g^{v(u+a_{k,1}j)}.$$

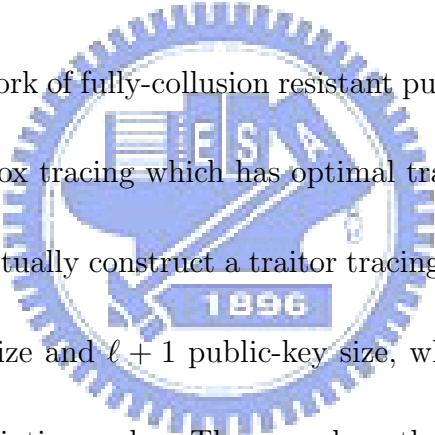
In this case, $\text{AONT}(\mathcal{D}(C); \rho) = M'_{k,\sigma}$, therefore \mathcal{B} only can give the correct answer with probability $\frac{1}{2}$;

If X is a random element in \mathbb{G} , then ciphertext C is a non-valid ciphertext. In this case, $\text{AONT}(\mathcal{D}(C); \rho) \neq M'_{k,\sigma}$ with probability ϵ , and $\text{AONT}(\mathcal{D}(C); \rho) = M'_{k,\sigma}$ with probability $1 - \epsilon$, therefore \mathcal{B} can give the correct answer with probability $\epsilon + \frac{1}{2}(1 - \epsilon) = \frac{1}{2} + \frac{\epsilon}{2}$.

Hence, \mathcal{B} can solve DDH problem with non-negligible advantage $\frac{\epsilon}{4}$, this is a contradiction to the DDH assumption, so we can conclude that such adversary \mathcal{A} does not exist. \square

Chapter 4

Conclusion



We propose a framework of fully-collusion resistant public-key traitor tracing schemes with black-box tracing which has optimal transmission rate. Using this framework, we actually construct a traitor tracing scheme **TTS-AONT** with $\ell + 1$ user-key size and $\ell + 1$ public-key size, where ℓ is the codeword length of the fingerprinting codes. Then we show that our **TTS-AONT** is semantically secure based on the hardness of DDH assumption and the semantic secure of the cryptosystem PKE-AONT. Also, our **TTS-AONT** is t -collusion resistant or fully-collusion resistant based on the DDH assumption.

There are some open problems: (1) How to improve the storage requirements of the user-key and public-key further? Maybe we can find a new

construction of fingerprinting codes with short length or use some tricks to decrease the storage requirements, etc. (2) In [2], Billet and Phan proposed a general attack "Pirate 2.0" against the code-base traitor tracing schemes (including ours). In Pirate 2.0, traitors can give their "part" of secret-key away but the data supplier can trace them with some uncertainties only. How to prevent such attack efficiently is also an important problem to make the code-base traitor tracing schemes more practical.



Bibliography

- [1] BELLARE, M., AND ROGAWAY, P. Optimal asymmetric encryption. In *EUROCRYPT* (1994), pp. 92–111.
- [2] BILLET, O., AND PHAN, D. H. Traitors collaborating in public: Pirates 2.0. In *EUROCRYPT* (2009), A. Joux, Ed., vol. 5479 of *Lecture Notes in Computer Science*, Springer, pp. 189–205.
- [3] BONEH, D., AND FRANKLIN, M. An efficient public key traitor tracing scheme. In *CRYPTO* (1999), Springer-Verlag, pp. 338–353.
- [4] BONEH, D., AND NAOR, M. Traitor tracing with constant size ciphertext. In *ACM Conference on Computer and Communications Security* (2008), P. Ning, P. F. Syverson, and S. Jha, Eds., ACM, pp. 501–510.
- [5] BONEH, D., SAHAI, A., AND WATERS, B. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*

- (2006), S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer, pp. 573–592.
- [6] BONEH, D., AND SHAW, J. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory* 44, 5 (1998), 1897–1905.
- [7] BOYKO, V. On the security properties of oaep as an all-or-nothing transform. In *CRYPTO* (1999), M. J. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, Springer, pp. 503–518.
- [8] CANETTI, R., DODIS, Y., HALEVI, S., KUSHILEVITZ, E., AND SAHAI, A. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT* (2000), pp. 453–469.
- [9] CHABANNE, H., PHAN, D. H., AND POINTCHEVAL, D. Public traceability in traitor tracing schemes. In *EUROCRYPT* (2005), R. Cramer, Ed., vol. 3494 of *Lecture Notes in Computer Science*, Springer, pp. 542–558.

- [10] CHOR, B., FIAT, A., AND NAOR, M. Tracing traitors. In *CRYPTO* (1994), Y. Desmedt, Ed., vol. 839 of *Lecture Notes in Computer Science*, Springer, pp. 257–270.
- [11] CHOR, B., FIAT, A., NAOR, M., AND PINKAS, B. Tracing traitors. *IEEE Transactions on Information Theory* 46, 3 (2000), 893–910.
- [12] FAZIO, N., NICOLOSI, A., AND PHAN, D. H. Traitor tracing with optimal transmission rate. In *ISC (2007)*, J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta, Eds., vol. 4779 of *Lecture Notes in Computer Science*, Springer, pp. 71–88.
- [13] FURUKAWA, J., AND ATTRAPADUNG, N. Fully collusion resistant black-box traitor revocable broadcast encryption with short private keys. In *ICALP (2007)*, L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, Eds., vol. 4596 of *Lecture Notes in Computer Science*, Springer, pp. 496–508.
- [14] KIAYIAS, A., AND YUNG, M. Traitor tracing with constant transmission rate. In *EUROCRYPT (2002)*, L. R. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, Springer, pp. 450–465.

- [15] KUROSAWA, K., AND DESMEDT, Y. Optimum traitor tracing and asymmetric schemes. In *EUROCRYPT* (1998), pp. 145–157.
- [16] NAOR, M., AND PINKAS, B. Threshold traitor tracing. In *CRYPTO* (1998), H. Krawczyk, Ed., vol. 1462 of *Lecture Notes in Computer Science*, Springer, pp. 502–517.
- [17] NAOR, M., AND PINKAS, B. Efficient trace and revoke schemes. In *Financial Cryptography* (2000), Y. Frankel, Ed., vol. 1962 of *Lecture Notes in Computer Science*, Springer, pp. 1–20.
- [18] PHAN, D. H. Traitor tracing for stateful pirate decoders with constant ciphertext rate. In *VIETCRYPT* (2006), P. Q. Nguyen, Ed., vol. 4341 of *Lecture Notes in Computer Science*, Springer, pp. 354–365.
- [19] PHAN, D. H., SAFAVI-NAINI, R., AND TONIEN, D. Generic construction of hybrid public key traitor tracing with full-public-traceability. In *ICALP (2)* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer, pp. 264–275.

- [20] RIVEST, R. L. All-or-nothing encryption and the package transform. In *FSE* (1997), E. Biham, Ed., vol. 1267 of *Lecture Notes in Computer Science*, Springer, pp. 210–218.
- [21] TARDOS, G. Optimal probabilistic fingerprint codes. In *STOC* (2003), ACM, pp. 116–125.
- [22] TÔ, V. D., SAFAVI-NAINI, R., AND ZHANG, F. New traitor tracing schemes using bilinear map. In *Digital Rights Management Workshop* (2003), M. Yung, Ed., ACM, pp. 67–76.
- [23] TZENG, W.-G., AND TZENG, Z.-J. A public-key traitor tracing scheme with revocation using dynamic shares. *Des. Codes Cryptography* 35, 1 (2005), 47–61.
- [24] ZHANG, R., HANAOKA, G., AND IMAI, H. On the security of cryptosystems with all-or-nothing transform. In *ACNS* (2004), M. Jakobsson, M. Yung, and J. Zhou, Eds., vol. 3089 of *Lecture Notes in Computer Science*, Springer, pp. 76–90.

Appendix A

All-Or-Nothing Transform

Let Ω be all mappings from infinite binary strings set $\{0, 1\}^\infty$ to finite binary strings set $\{0, 1\}^*$. Let $H \leftarrow \Omega$ denote that we choose a function from Ω uniformly. We define the secure game of an all-or-nothing transform function as follows:



Indistinguish.

- *Setup.* The challenger chooses a security parameter λ and constructs an all-or-nothing transform function AONT^Γ by the random oracle $\Gamma \leftarrow \Omega$. Then it gives λ and AONT^Γ to the adversary.
- *Challenge.* The adversary selects a position set $L \in \{1, 2, \dots, n'\}$ and two strings $x_0, x_1 \in \{0, 1\}^{n'}$ to the challenger, where $|L| = \lambda$.

Then the challenger flips a coin $b \in \{0, 1\}$, and generates $C \leftarrow \text{AONT}(x_b; \rho)$. After hiding the bits of C in position set L , send the ciphertext C_L to the adversary.

- *Guess.* The adversary returns a guess $b' \in \{0, 1\}$ of b to the challenger.

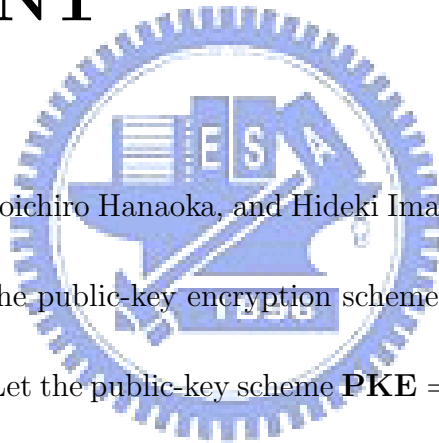
We define the advantage of the adversary wins this game as $\text{Adv}_{\text{ind}}^{\text{AONT}} := |\Pr[b' = b] - \frac{1}{2}|$.

Definition. We say that an all-or-nothing transform function is indistinguishable if for all polynomial time adversary \mathcal{A} , $\text{Adv}_{\text{ind}}^{\text{AONT}}$ is a negligible function of $\lambda > 0$, i.e.,

$$\Pr \left[b' = b \mid \begin{array}{l} \langle \Gamma, \lambda, \text{AONT}^\Gamma \rangle \leftarrow \text{Setup}, \\ L, x_0, x_1 \leftarrow \mathcal{A}(\lambda, \text{AONT}^\Gamma), b \xleftarrow{\$} \{0, 1\}, \\ C_{b,L} \leftarrow \text{AONT}(x_b; \rho), b' \leftarrow \mathcal{A}(C_{b,L}) \end{array} \right] \leq \frac{1}{2} + \text{neg}(\lambda).$$

Appendix B

The Public-Key Cryptosystem with AONT



In [24], Rui Zhang, Goichiro Hanaoka, and Hideki Imai proposed a cryptosystem that combined the public-key encryption scheme and the all-or-nothing transform function. Let the public-key scheme $\mathbf{PKE} = (G, E, D)$. The public-key cryptosystem with AONT $\mathbf{PKE-AONT} = (Gen, Enc, Dec)$ is as follows

- $Gen(1^\lambda) \rightarrow \langle (pk, sk), AONT \rangle$

Given a security parameter λ , the algorithm calls $G(1^\lambda)$ to generate a public key and secret key pair (pk, sk) , and it picks an randomized all-or-nothing transform function AONT.

- $Enc(PK, AONT, M') \rightarrow C$

Given the public key pk , the all-or-nothing transform function AONT and a plaintext $M' \in \mathcal{M}^{n'}$, the algorithm chooses a random string $\rho \in \{0, 1\}^\tau$, it calls $\text{AONT}(M'; \rho)$ to generate $M = m_1 || m_2 || \dots || m_n \in \mathcal{M}^n$, and chooses a position $k \xleftarrow{\$} \{1, 2, \dots, n\}$, then it outputs the ciphertext

$$C := \langle k, c_1, c_2 \rangle \xleftarrow{\$} \langle k, \mathbf{E}(pk, m_k), m_1 || \dots || m_{k-1} || m_{k+1} || \dots || m_n \rangle.$$

- $\text{Dec}(sk, \text{AONT}, C) \rightarrow M'$

Given the secret key sk and the ciphertext $C = \langle k, c_1, c_2 \rangle$, the algorithm calls $\mathcal{D}(sk, c_1)$ to recover m_k , then it puts m_k into the k -th block of c_2 to recover M . Finally,

$$M' \leftarrow \text{AONT}^{-1}(M).$$

We define the security of a public key encryption scheme with all-or-nothing transform functions as follows:

Semantic Secure.

- *Setup.* The challenger runs Gen , then it gives the public key pk and the function AONT to the adversary \mathcal{A} .

- *Challenge.* The adversary chooses two plaintexts $M'_0, M'_1 \in \mathcal{M}^{n'}$ to the challenger. Then the challenger flips a coin $b \in \{0, 1\}$, and it gives $C_b \stackrel{\$}{\leftarrow} \text{Enc}(pk, M'_b)$ to \mathcal{A} .
- *Guess.* The adversary returns a guess $b' \in \{0, 1\}$ of b to the challenger.

We define the advantage of \mathcal{A} wins this game as $\text{Adv}_{\text{SS}}^{\text{PKE-AONT}} := |\Pr[b' = b] - \frac{1}{2}|$.

Definition. We say that a public key encryption scheme with all-or-nothing transform functions is semantic secure if for all polynomial time adversaries \mathcal{A} , $\text{Adv}_{\text{SS}}^{\text{PKE-AONT}}$ is a negligible function of λ , i.e.

$$\Pr \left[b' = b \mid \begin{array}{l} \langle (pk, sk), \text{AONT} \rangle \stackrel{\$}{\leftarrow} \text{Gen}(1^\lambda) \\ (M'_0, M'_1) \leftarrow \mathcal{A}(pk, \text{AONT}), b \stackrel{\$}{\leftarrow} \{0, 1\}, \\ C_b \stackrel{\$}{\leftarrow} \text{Enc}(pk, \text{AONT}, M'_b), b' \leftarrow \mathcal{A}(C_b) \end{array} \right] \leq \frac{1}{2} + \text{neg}(\lambda).$$

Appendix C

The Traitor Tracing Scheme

In [23], Tzeng and Tzeng proposed a t -collusion resistant public key traitor tracing scheme $\mathbf{TR} = (\text{Setup}, \text{Encrypt}, \text{Decrypt}, \text{Trace})$, and it can revoke at most z traitors ($z \geq 2k$), where

$$\text{Setup}(1^\lambda, n) \rightarrow \langle bk, tk, (sk_1, sk_2, \dots, sk_n) \rangle$$

Given a security parameter λ and the number of users n , the algorithm generates a λ -bit prime q , a group \mathbb{G} of order q , chooses a generator g of \mathbb{G} and $a_0, a_1, \dots, a_z \xleftarrow{\$} \mathbb{Z}_q^*$, then it lets $f(x) = a_0 + a_1x + \dots + a_zx^z \pmod{q}$. Sets

- Public broadcast key $bk := \langle q, g, g^{a_0}, g^{f(1)}, g^{f(2)}, \dots, g^{f(z)} \rangle$
- Secret tracing key $tk := \langle f(x) \rangle$

- User private key $sk_\sigma := \langle i_\sigma, f(i_\sigma) \rangle$, where $\sigma \xleftarrow{\$} \mathbb{Z}_q^*$, $i_\sigma > z, \forall \sigma \in \{1, 2, \dots, n\}$

Encrypt(bk, m) $\xrightarrow{\$}$ c

Given the broadcast key bk and a plaintext $m \in \mathcal{M}$, the algorithm chooses the unused shares $j_1, j_2, \dots, j_z \xleftarrow{\$} \mathbb{Z}_q^*$, where $j_i \neq i_\sigma, \forall i \in \{1, 2, \dots, z\}, \forall \sigma \in \{1, 2, \dots, n\}$, and chooses $r \xleftarrow{\$} \mathbb{Z}_q^*$, then it computes

$$c \xleftarrow{\$} \langle (sg^{ra_0}, g^r, (j_1, g^{rf(j_1)}), \dots, (j_z, g^{rf(j_z)})) \rangle$$

Decrypt($sk_\sigma, c = \langle A, R, (j_1, Y_1), \dots, (j_z, Y_z) \rangle$) $\rightarrow m$

Given a user private key sk_σ and the ciphertext c , the algorithm compute

$$s \leftarrow A / [R^{f(i_\sigma)\lambda_z} \cdot \prod_{i=0}^{z-1} (Y_i^{\lambda_i})], \text{ where } \lambda_i: \text{Lagrange coefficients.}$$

Trace $^{\mathcal{D}}$ ($tk, 1$) $\rightarrow S$

Given the tracing key tk and a pirate decoder \mathcal{D} that can decrypt all valid ciphertexts perfectly as a decryption oracle.

- (1) For every possible m -user set $\{u_1, u_2, \dots, u_m\}$, where $m \leq k$ do:

- Randomly select $z - m$ unused shares j_1, j_2, \dots, j_{z-m} , then set

the probe ciphertext

$$c \stackrel{\$}{\leftarrow} \langle (sg^{ra_0}, g^r, (u_1, g^{rf(u_1)}), \dots, (u_m, g^{rf(u_m)}), (j_1, g^{rf(j_1)}), \dots, (j_{z-m}, g^{rf(j_{z-m})})) \rangle$$

- If $\mathcal{D}(C)$ does not output m , then $\{u_1, u_2, \dots, u_m\}$ is a possible traitor set.

(2) Output the smallest of all possible traitor sets found in (1).

Theorem. **TR** is semantic secure under the DDH assumption.

