論文 / 著書情報
Article / Book Information

題目(和文)	放送型暗号系に関する研究
Title(English)	A study on broadcast encryption
著者(和文)	吉田琢也
Author(English)	
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第5040号, 授与年月日:2002年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第5040号, Conferred date:2002/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

A Study on Broadcast Encryption

Takuya Yoshida

Department of Electrical and Electronic Engineering,
Graduate School of Science and Engineering,
Tokyo Institute of Technology
takuya@ss.titech.ac.jp

Advisor: Professor Kaoru Kurosawa
Associate Professor Wakaha Ogata

January 2002

Contents

Chapter 1

Introduction

As wide bandwidth broadcast channels such as broadband internet connections and digital satellite broadcasts are coming into wide use, in addition to large capacity digital media such as CD-ROM and DVD-ROM are becoming cheaper and more available, distribution of large digital data is becoming much more popular. Among typical applications are pay TV, online databases, group telecommunication and secure distribution of copyright-protected materials such as music or video clips. In these applications, data should only be available to authorized users. To prevent unauthorized users from accessing data, the data supplier first provides the authorized users with decryption keys when they join the system. The data supplier then encrypts and broadcasts the data. Broadcast Encryption provides a secure and efficient solution to this problem and so has been drawing wide attention recently.

Within this thesis, we deal with three different aspects of broadcast encryption which include, Conditional Access Scheme, Traitor Tracing Scheme and Identification via Channels.

Conditional Access Scheme (CAS)

It is often desirable for the data supplier to broadcast data in such a way that only members of a subset of authorized users have access to it, while non-members of the subset cannot obtain any information on the data even if they have their own decryption keys. It is also important that the data supplier can determine the privileged subset dynamically and without affecting any decryption keys because re-distribution of secret keys comes at a significant cost. Consider an application to pay TV as an example. When an authorized user does not pay the fee for sometime or cancels a subscription for the service, then the service to the user is revoked and his decryption key has to be made ineffective for decryption. If the data supplier provides two or more channels, users may subscribe to different channels according to their own preferences; Some users may want to watch a sports channel and others may not, for instance. Thus, the set of subscribers varies channel by channel.

The obvious solution to this problem is to give every user an independent key and transmit individually encrypted data to every member of the privileged class. This requires a very long transmission (the number of members in the class times the length of the massage). Still, another simple solution may be to provide every possible subset of users with a key, that is, give every user the keys corresponding to the subsets it belongs to. However, this requires every user to store a large number of keys.

The goal of conditional access schemes (CAS) is to provide solutions which are efficient in both transmission length and key size, while aiming to have schemes that are computationally efficient. The notion of conditional access schemes was first formally introduced by Fiat and Naor[27] as "Broadcast Encryption Scheme." However, we will use the terminology "Conditional Access Scheme" throughout this thesis in order to clarify the functionality of the scheme and to avoid confusion between other broadcast schemes such as the Traitor Tracing Scheme.

Traitor Tracing Scheme

A set of malicious authorized users called traitors may conspire to construct a pirate decoder which decrypt encrypted broadcast data then duplicate and distribute them illegally. If a decryption key is implemented as software, it is easy to construct such a pirate decoder Even if it is provided as hardware, it may be possible via reverse engineering or due to sloppy manufacturing of the device. When this piracy happens, it is highly desirable to trace traitors from the pirate decoders in order to discourage this type of illegal act.

Chor, Fiat and Naor have shown the first solution to this problem by introducing the first Traitor Tracing Scheme in [19]. Upon confiscating a pirate decoder, a traitor tracing scheme enables the data supplier to identify at least one of the traitors who constructed the decoder. It is also desirable for a traitor tracing scheme to be efficient from the point of transmission, the size of its keys and the Computation of encryption, decryption and tracing procedures.

Identification via Channels

When a user receives data transmitted through a broadcast channel, the data may be valuable for or intended to only a few authorized users. Hence, to reduce redundant decryption computation on the users' side, it is desirable that users are able to determine if they want to decrypt it. Identification via Channels, proposed by Ahlswede and Dweck, has beneficial properties for this problem.

A conventional transmission code is suitable in situations where a recipient wants to know what event occurred. On the other hand, if a recipient is only interested in verifying whether or not a certain event e_i occurred, then identification coding is suitable. An identification code employs a randomized encoding procedure in contrast to the deterministic encoding procedure of conventional transmission codes. The recipient is allowed to select a list of events allowing to determine whether the event e_i is in it or not. The most remarkable property is that the number of events that can be reliably identified using an identification code is exponentially larger than a conventional transmission code.

Organization

In Chapter 2, we first present two tight lower bounds on the size of the secret keys of each user in an unconditionally secure one-time use conditional access scheme (OTCAS). We then show how to construct a computationally secure multiple-use conditional access scheme (MCAS) from a key predistribution scheme (KPS) by using the ElGamal cryptosystem. We prove that our MCAS is secure against chosen (message, privileged subset of users) attacks if the ElGamal cryptosystem is secure and if the original KPS is simulated. This is the first MCAS with security that is proven formally.

In Chapter 3, we show an efficient construction of a class of conditional access schemes. We say that a conditional access scheme is a (w, n)-revocation scheme if a center can exclude w or less users among n users. In this chapter, we present efficient (w, n)-revocation schemes such that $\rho_T = O(w^{1+\epsilon})$ and $\rho_T = 1 + \epsilon$ for any $\epsilon > 0$ where transmission rate ρ_T is defined as

$$\rho_T \triangleq \frac{\text{the length of a ciphtertext}}{\text{the length of a plaintext}}$$

by showing new constructions of cover free families. We also show a construction of cover free families which yields a (w, n)-revocation scheme such that not only $\rho_T = O(w^2)$ but it can also be used as a w-resilient traceability scheme.

In Chapter 4, we first show that three public-key (k, n)-traceability schemes can be derived from a $[n, u, d]$-linear code C such that $d \geq 2k + 1$. The previous schemes are obtained as special cases. This observation provides more freedom and new insight into this field. For example, we demonstrate that Boneh-Franklin scheme[14] is equivalent to a slight modification of the corrected Kurosawa-Desmedt scheme[39]. This means that BF scheme is redundant or overdesigned because the modified KD scheme is much simpler. It is also shown that the corrected KD scheme is the best among them. In addition, we show a tracing algorithm which can detect *all* traitors by using a confiscated pirate decoder as a black box. This algorithm is applicable to all the public-key traceability schemes discussed in this chapter and the trivial scheme. This is the first black box full tracing algorithm with traceability that is formally proven.

In Chapter 5 we provide attacks and comments on some of the revocation and tracing schemes. They include Chor, Fiat and Naor(CFN) traceability scheme[19, 20], Naor, Naor and Lotspiech(NNL) revocation schemes with traceability[48], Matsuzaki, Anzai and Matsumoto(MAM) revocation scheme[47], Yoshida and Fujiwara(YF) revocation scheme with traceability[65] and Tzeng and Tzeng(TT) revocation scheme with traceability[61]. For example, Naor, Naor and Lotspiech showed two revocation schemes and a traitor tracing algorithm for them at Crypto 2001[48]. However, we illustrate that NNL revocation schemes cannot be traceable.

Finally, in Chapter 6, we show that ϵ-almost strongly universal classes of hash functions can yield better explicit constructions of identification codes via channels (ID codes) and identification plus transmission codes (IT codes) than the previous explicit constructions of Verdú and Wei.

Chapter 2

Some Bounds and a Construction for Secure Conditional Access Schemes

2.1 Introduction

Secure broadcast encryption is one of the central problems in communication and network security. In this chapter we link One-Time use Conditional Access Schemes (OTCASs) [27, 4, 58] with Key Predistribution Schemes (KPS)[46]. Both schemes are closely related but they have a different structure. In a KPS, a Trusted Authority (TA) distributes secret information to a set of users such that, each member of a privileged subset P of users can compute a specified key k_P, but no coalition F (forbidden subset) is able to recover any information on the key k_P that it is not supposed to know. In an OTCAS, the TA distributes secret information to a set of users and then broadcasts a ciphertext c_P over a network. The secret information is such that each member of a particular subset P of users can decrypt c_P, but no coalition F (forbidden subset) is able to recover any information on the plaintext m_P of c_P that it is not supposed to know.

A natural way to construct an OTCAS from a KPS is to use a key k_P of the KPS to encrypt the message m_P, that is

$$c_P = k_P + m_P. \tag{2.1}$$

Stinson *et al.* [11, 58] have shown that there is a tradeoff between $|C_P|$ and $|U_i|$ in OTCASs, where C_P is the set of ciphertexts c_P and U_i is the set of secrets of user i. That is, $|C_P|$ can be decreased by increasing $|U_i|$ and vice versa.

A $(\mathcal{P}, \mathcal{F})$-KPS is a KPS for which $\mathcal{P} \triangleq \{P \mid P \text{ is a privileged subset}\}$ and $\mathcal{F} \triangleq \{F \mid F \text{ is a forbidden subset}\}$. In particular,

- A $(t, \leq w)$-KPS is a $(\mathcal{P}, \mathcal{F})$-KPS with $\mathcal{P} = \{P \mid |P| = t\}$, $\mathcal{F} = \{F \mid |F| \leq w\}$,
- A $(\leq n, \leq w)$-KPS is a $(\mathcal{P}, \mathcal{F})$-KPS with $\mathcal{P} = 2^{\mathcal{U}}$, $\mathcal{F} = \{F \mid |F| \leq w\}$, where $\mathcal{U}$ is the set of users and $n \triangleq |\mathcal{U}|$.

We define $(\mathcal{P}, \mathcal{F})$-OTCASs, $(t, \leq w)$-OTCASs and $(\leq n, \leq w)$-OTCASs in a similar way. Below we list some of the known KPSs and OTCASs.

Key Predistribution Schemes. Blom obtained a $(2, \leq w)$-KPS in [8] by using MDS codes (also see [46]). Blundo *et al.* obtained a $(t, \leq w)$-KPS in [9] by using symmetric polynomials. Fiat and Naor presented a $(\leq n, \leq w)$-KPS in [27]. Blundo *et al.* found tight lower bounds on $|U_i|$ for $(t, \leq w)$-KPSs [9] and for $(\leq n, \leq w)$-KPSs [10].[1] Recently, Luby and Staddon found some bounds and constructions for some classes of $(n - w, \leq w)$-OTCASs [45]. However, there is a gap between their bounds and the constructions.

One-Time use Conditional Access Schemes. Stinson *et al.* gave constructions for $(t, \leq w)$-OTCASs [11] and $(\leq n, \leq w)$-OTCASs [58] which can realize the tradeoff between $|C_P|$ and $|U_i|$. Blundo, Mattos and Stinson found a lower bound on $|C_P|$ and $|U_i|$ for $(t, \leq w)$-OTCASs which reflects the tradeoff [11]. Recently, Desmedt and Viswanathan presented a $(\leq n, \leq n)$-KPS [22]. This can be considered as a complement of the Fiat and Naor $(\leq n, \leq n)$-KPS.

In this chapter, we first prove that a $(\mathcal{P}, \mathcal{F})$-KPS is equivalent to a $(\mathcal{P}, \mathcal{F})$-OTCAS when $|C_P| = |M|$, where M denotes the set of messages (Theorems 2.4.1, 2.4.2). Then, by using the bounds in [9, 10] for KPSs we get directly a lower bound on $|U_i|$ for $(\leq n, \leq w)$-OTCASs and a lower bound for $(t, \leq w)$-OTCASs. The former is the first lower bound for $(\leq n, \leq w)$-OTCASs. The latter is tighter than the bound of Blundo, Mattos and Stinson for $|C_P| = |M|$. Both bounds are tight because the natural schemes which use

[1] The model for broadcast encryption in [10, 27] corresponds to our model for KPSs. So, for example, the bounds in [10] hold only for KPSs, and not for OTCASs.

(2.1) meet the equalities of our bounds. We also present a general lower bound on $|U_i|$ for KPSs which includes all the previous known bounds as special cases (Theorem 2.4.3).

Next, we show how to construct a computationally secure $(\mathcal{P}, \mathcal{F})$-Multiple use Conditional Access Scheme ($(\mathcal{P}, \mathcal{F})$-MCAS) from a $(\mathcal{P}, \mathcal{F})$-KPS by using the ElGamal cryptosystem. We prove (Theorem 2.5.1) that our $(\mathcal{P}, \mathcal{F})$-MCAS is secure against *chosen (message, privileged subset of users) attacks* (Definition 2.5.1) if the ElGamal cryptosystem is secure and if the original $(\mathcal{P}, \mathcal{F})$-KPS is *simulatable* (Definition 2.5.3).

We then show that the Blundo *et al.* scheme, the Fiat-Naor scheme and the Desmedt-Viswanathan scheme are all simulatable (Theorems 2.5.2, 2.5.3). By combining this result with our earlier construction we get $(\mathcal{P}, \mathcal{F})$-MCASs for $(\mathcal{P}, \mathcal{F}) = (t, \leq w)$ and $(\leq n, \leq w)$ whose security is proven formally.

The proposed construction is the first MCAS whose security is proven formally (Corollary 2.5.1). Furthermore, our technique can be generalized to many of the OTCASs in [58], and our argument holds for Multiple use $(\mathcal{P}, \mathcal{F})$-KPSs.

2.2 Mathematical models [11, 58]

Our model for key distribution and broadcast encryption consists of a Trusted Authority (TA) and a set of users $\mathcal{U} = \{1, 2, \ldots, n\}$.

2.2.1 Key predistribution

In a key pre-distribution scheme, the TA generates and distributes secret information to each user. The information given to user i is denoted by u_i and must be distributed "off-band" (i.e., not using the network) in a secure manner. This secret information will enable various *privileged subsets* to compute keys.

Let $2^{\mathcal{U}}$ denote the set of all subsets of users. $\mathcal{P} \subseteq 2^{\mathcal{U}}$ will denote the collection of all privileged subsets to which the TA distributes keys. $\mathcal{F} \subseteq 2^{\mathcal{U}}$ will denote the collection of all possible coalitions (called *forbidden subsets*) against which each key is to remain secure.

Once the secret information is distributed, each user i in a privileged set P should be able to compute the key k_P associated with P. On the other hand, no forbidden set $F \in \mathcal{F}$ disjoint from P should be able to compute any information about k_P.

Let K_P denote the set of possible keys associated with P. We assume that $K_P = K$ for each $P \in \mathcal{P}$.

For $1 \leq i \leq n$, let U_i denote the set of all possible secret values that might be distributed to user i by the TA. For any subset of users $X \subseteq \mathcal{U}$, let U_X denote the cartesian product $U_{i_1} \times \cdots \times U_{i_j}$, where $X = \{i_1, \ldots, i_j\}$ and $i_1 < \cdots < i_j$. We assume that there is a probability distribution on $U_\mathcal{U}$, and that the TA chooses $u_\mathcal{U} \in U_\mathcal{U}$ according to this probability distribution.

We say that the scheme is a $(\mathcal{P}, \mathcal{F})$-*Key Predistribution Scheme* ($(\mathcal{P}, \mathcal{F})$-KPS) if the following conditions are satisfied:

1. Each user i in any privileged set P can compute k_P:
   $\forall i \in P$, $\forall P \in \mathcal{P}$, $\forall u_i \in U_i$, $\exists k_P \in K_P$ s.t.,
   $$\Pr[K_P = k_P \mid U_i = u_i] = 1.$$
2. No forbidden subset F disjoint from any privileged subset P has any information on k_P:
   $\forall P \in \mathcal{P}$, $\forall k_P \in K_P$, $\forall F \in \mathcal{F}$ s.t. $P \cap F = \emptyset$, $\forall u_F \in U_F$ s.t. $\Pr(U_F = u_F) > 0$,
   $$\Pr[K_P = k_P \mid U_F = u_F] = \Pr[K_P = k_P]. \tag{2.2}$$

We denote a $(\mathcal{P}, \mathcal{F})$-KPS by $(U_1, \ldots, U_n, K)$.

2.2.2 One-time Broadcast Encryption

We will use the notation from Section 2.2.1. We assume that the network is a *broadcast channel*, i.e., it is insecure, and that any information transmitted by the TA will be received by every user.

In a set-up stage, the TA generates and distributes secret information u_i to each user i off-band. At a later time, the TA will want to broadcast a message to a privileged subset P. The particular privileged subset P is, in general, not known ahead of time.

$\mathcal{P} \subseteq 2^\mathcal{U}$ will denote the collection of all privileged subsets to which the TA might want to broadcast a message. $\mathcal{F} \subseteq 2^\mathcal{U}$ will denote the collection of all possible coalitions (forbidden subsets) against which a broadcast is to remain secure.

Now, suppose that the TA wants to broadcast a message to a given privileged set $P \in \mathcal{P}$ at a later time. (The particular privileged set P is not known when the scheme is set up, except for the restriction that $P \in \mathcal{P}$.) Let M_P denote the set of possible messages that might be broadcast to P. We assume that $M_P = M$ for each $P \in \mathcal{P}$. Furthermore, we assume that there is a probability distribution on M, and that the TA chooses a *message* (i.e., a plaintext) $m_P \in M$ according to this probability distribution. Then the *broadcast* c_P (which is an element of a specified set C_P) is computed as a function of m_P and u_P.

Once c_P is broadcast, each user $i \in P$ should be able to decrypt c_P and obtain m_P. On the other hand, no forbidden set $F \in \mathcal{F}$ disjoint from P should be able to compute any information about m_P.

The security of the scheme is in terms of a single broadcast, so we call the scheme *one-time.* We say that the scheme is a $(\mathcal{P}, \mathcal{F})$-*One-Time Conditional Access Scheme* ($(\mathcal{P}, \mathcal{F})$-OTCAS) if the following conditions are satisfied:

1. Without knowing the broadcast c_P, no subset of users has any information about the message m_P, even if given all the secret information $U_{\mathcal{U}}$:
$\forall P \in \mathcal{P}$, $\forall m_P \in M_P$, $\forall u_{\mathcal{U}} \in U_{\mathcal{U}}$ s.t. $\Pr[U_{\mathcal{U}} = u_{\mathcal{U}}] > 0$,
$$\Pr[M_P = m_P \mid U_U = u_U] \;=\; \Pr[M_P = m_P]. \tag{2.3}$$

2. The message for a privileged user is uniquely determined by the broadcast message and the user's secret information:
$\forall i \in P$, $\forall P \in \mathcal{P}$, $\forall u_i \in U_i$, $\forall c_P \in C_P$, $\exists m_P \in M_P$ s.t.,
$$\Pr[M_P = m_P \mid U_i = u_i, C_P = c_P] \;=\; 1. \tag{2.4}$$

3. After receiving the broadcast message, no forbidden subset F disjoint from P has any information on m_P:
$\forall P \in \mathcal{P}$, $\forall F \in \mathcal{F}$ s.t. $P \cap F = \emptyset$, $\forall m_P \in M_P$, $\forall u_F \in U_F$, $\forall c_P \in C_P$,
$$\Pr[M_P = m_P \mid U_F = u_F, C_P = c_P] \;=\; \Pr[M_P = m_P]. \tag{2.5}$$

We denote a $(\mathcal{P}, \mathcal{F})$-OTCAS by $(U_1, \ldots, U_n, M, \{C_P\})$.

2.2.3 Conventional notation

We first consider key predistribution schemes. If $\mathcal{P}$ consists of all t-subsets of $\mathcal{U}$, then we will write $(t, \mathcal{F})$-KPS. Similarly, if $\mathcal{P}$ consists of all subsets of

$\mathcal{U}$ of size at most t, we write $(\leq t, \mathcal{F})$-KPS. An analogous notation will be used for $\mathcal{F}$. Thus, for example, a $(\leq n, 1)$-KPS is a KPS for which there is a key associated with any subset of users (i.e., $\mathcal{P} = 2^{\mathcal{U}}$) and no key k_P can be computed by any individual user $i \notin P$. Note that in any $(\mathcal{P}, \mathcal{F})$-KPS, if $F \in \mathcal{F}$ and $F' \subseteq F$, then $F' \in \mathcal{F}$. Hence, a $(\mathcal{P}, \leq w)$-KPS is a $(\mathcal{P}, \leq w)$-KPS.

The same notation is used for one-time use conditional access schemes.

2.3 Known results

For a random variable X, $H(X)$ denotes the entropy of X. Generally,

$$0 \leq H(X) \leq \log_2 |X|, \text{ where } X \triangleq \{x \mid \Pr[X = x] > 0\}.$$

In particular, $H(X) = \log_2 |X|$ iff X is uniformly distributed.

2.3.1 A $(t, \leq w)$-KPS (The Blundo *et al.* scheme)

Blom presented a $(2, \leq w)$-KPS in [8]. This was generalized to a $(t, \leq w)$-KPS by Blundo *et al.* as follows [9]. Let q be a prime such that $q \geq n$ (the number of users). The TA chooses a random *symmetric* polynomial in t variables over $GF(q)$ in which the degree of any variable is at most w, that is, a polynomial

$$f(x_1, \ldots, x_t) = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} a_{i_1 \cdots i_t} x_1^{i_1} \cdots x_t^{i_t},$$

where, $a_{i_1 \cdots i_t} = a_{\pi(i_1 \cdots i_t)}$ for any permutation π on $(i_1, \ldots, i_t)$. The TA computes u_i as $u_i = f(i, x_2, \ldots, x_t)$ and gives u_i to user i secretly for $1 \leq i \leq n$. The key associated with the t-subset $P = \{i_1, \ldots, i_t\}$ is $k_P = f(i_1, \ldots, i_t)$. Each user $j \in P$ can compute k_P from u_j easily. In this scheme, $|K_P| = q = |K|$ and

$$\log |U_i| = \binom{t + w - 1}{t - 1} \log |K|.$$

This scheme is optimum because Blundo *et al.* have shown that the following lower bound on $|U_i|$ applies.

Proposition 2.3.1 *[9] In a $(t, \leq w)$-KPS,*

$$\log |U_i| \geq \binom{t + w - 1}{t - 1} H(K).$$

Beimel and Chor gave a combinatorial proof of Proposition 2.3.1 [4]. Blundo and Cresti obtained the following more general lower bound.

Proposition 2.3.2 *[10] In a $(\mathcal{P},\mathcal{F})$-KPS with $\{1,2,\cdots,n\}\setminus P\in\mathcal{F}$ for all $P\in\mathcal{P}$,*

$$\log|U_i|\geq\tau_i H(K),$$

where $\tau_i=|\{P\in\mathcal{P}\mid i\in P\}|$

Note that Proposition 2.3.1 is obtained from Proposition 2.3.2 by letting $n=t+w$.

2.3.2 A $(\leq n,\leq w)$-KPS (The Fiat-Naor scheme)

Fiat and Naor presented the following $(\leq n,\leq w)$-KPS [27]. Let q be any positive integer. For every subset $F\subseteq\mathcal{U}$ of cardinality at most w, the TA chooses a random value $s_F\in Z_q$ and gives s_F to every member of $\mathcal{U}\setminus F$ as the secret information. Then the key associated with a privileged set P is defined to be

$$k_P=\sum_{F:F\in\mathcal{F},F\cap P=\emptyset} s_F \pmod{q},$$

Here is a small example for illustration. Take $n=3$, $q=17$ and $w=1$, and suppose that the TA chooses the values,

$$s_\emptyset=11,\quad s_{\{1\}}=8,\quad s_{\{2\}}=3,\quad s_{\{3\}}=8.$$

The secret information of the users is,

$$u_1=\{s_\emptyset,s_{\{2\}},s_{\{3\}}\},\quad u_2=\{s_\emptyset,s_{\{1\}},s_{\{3\}}\},\quad u_3=\{s_\emptyset,s_{\{1\}},s_{\{2\}}\}.$$

The keys determined by this information are,

$$k_{\{1,2\}}=s_\emptyset+s_{\{3\}}=2 \bmod 17,\quad \ldots\quad ,k_{\{1,2,3\}}=s_\emptyset=11 \bmod 17.$$

In this scheme, $|K_P|=q=|K|$ and

$$\log|U_i|=\sum_{j=0}^{w}\binom{n-1}{j}\log|K|.$$

This scheme is optimum because Blundo and Cresti have shown the following Proposition and Corollary.

Proposition 2.3.3 *[10] In a $(\leq n, \mathcal{F})$-KPS,*

$$\log|U_i| \geq v_i H(K)$$

where $v_i = |\{F \in \mathcal{F} \mid i \notin F\}|$.

Corollary 2.3.1 *[10] In a $(\leq n, \leq w)$-KPS,*

$$\log|U_i| \geq \sum_{j=0}^{w} \binom{n-1}{j} H(K).$$

2.3.3 The $(\leq n, \leq n)$-KPS (The Desmedt-Viswanathan scheme)

Desmedt and Viswanathan presented a $(\leq n, \leq n)$-KPS [22]. This scheme can viewed as a complement of the Fiat-Naor $(\leq n, \leq n)$-KPS. The TA initially generates $2^n - n - 1$ independent keys, i.e., one for each $P \subseteq \{1, 2, \ldots, n\}$ such that $|P| \geq 2$. Each user i receives from the TA the keys of those subsets for which $i \in P$. Hence, each user gets $2^{n-1} - 1$ keys. This scheme is optimum because of the following lower bound which follows from Corollary 2.3.1.

Corollary 2.3.2 *In a $(\leq n, \leq n)$-KPS,*

$$\log|U_i| \geq (2^{n-1} - 1)H(K).$$

(Desmedt and Viswanathan gave another direct proof [22].)

2.3.4 Lower bounds for $(t, \leq w)$-OTCASs

Blundo, Mattos and Stinson obtained the following lower bound for $(t, \leq w)$-OTCASs [11],

Proposition 2.3.4 *In any $(t, \leq w)$-OTCAS with $t \geq w + 1$,*

$$H(C_P) + \sum_{j=1}^{w} H(U_{i_j}) \geq (2w + 1)H(M),$$

for any $P \in \mathcal{P}$.

2.4 New lower bounds on $|U_i|$

In this section we first prove that a $(\mathcal{P},\mathcal{F})$-KPS is equivalent to a $(\mathcal{P},\mathcal{F})$-OTCAS when $|C_P| = |M|$. Then, by using the bounds in [9, 10] for KPSs, we get directly a lower bound on $|U_i|$ for $(\leq n, \leq w)$-OTCASs and a lower bound for $(t, \leq w)$-OTCASs. The former is the first lower bound presented for $(\leq n, \leq w)$-OTCASs. The latter is tighter than the bound of Blundo, Mattos and Stinson for $|C_P| = |M|$. Our bounds are both tight. We also present a general lower bound on $|U_i|$ for KPSs which includes all the previous bounds as special cases.

2.4.1 Equivalence between KPS and OTCAS

Theorem 2.4.1 *If there exists a $(\mathcal{P},\mathcal{F})$-KPS $(U_1,\ldots,U_n,K)$, then there exists a $(\mathcal{P},\mathcal{F})$-OTCAS $(U_1,\ldots,U_n,M,\{C_P\})$ with $|C_P| = |M| = |K|$ for all $P \in \mathcal{P}$.*

(Proof) Use a key k_P of the $(\mathcal{P},\mathcal{F})$-KPS to encrypt a message m_P, that is

$$c_P = k_P + m_P,$$

and broadcast c_P. We then get a $(\mathcal{P},\mathcal{F})$-OTCAS. □

Theorem 2.4.2 *If there exists a $(\mathcal{P},\mathcal{F})$-OTCAS $(U_1,\ldots,U_n,M,\{C_P\})$ such that $|C_P| = |M|$ for all $P \in \mathcal{P}$, then there exists a $(\mathcal{P},\mathcal{F})$-KPS $(U_1,\ldots,U_n,K)$ such that $|K| = |M|$ and $H(K) = H(M)$.*

(Proof) From a $(\mathcal{P},\mathcal{F})$-OTCAS construct a KPS as follows. Fix $c_P \in C_P$ arbitrarily for all $P \in \mathcal{P}$. Since $|C_P| = |M|$, there is a bijection from C_P to M for any $(u_1,\ldots,u_n)$. Then there is an $\hat{m}_P \in M$ such that each member of P decrypts the c_P as $\hat{m}_P$ for any $(u_1,\ldots,u_n)$. Now take $k_P = \hat{m}_P$ in our KPS. It is easy to see that we get a $(\mathcal{P},\mathcal{F})$-KPS with $|K| = |M|$ and $H(K) = H(M)$. □

2.4.2 Lower bounds for OTCASs

From Theorem 2.4.2, Proposition 2.3.1, and Corollary 2.3.1, we obtain immediately the following lower bounds on $|U_i|$ for OTCASs.

Corollary 2.4.1 *In a $(t, \leq w)$-OTCAS, if $|C_P| = |M|$ for all $P \in \mathcal{P}$, then*

$$\log |U_i| \geq \binom{t+w-1}{t-1} H(M).$$

Corollary 2.4.2 *In a $(\leq n, \leq w)$-OTCAS, if $|C_P| = |M|$ for all $P \in \mathcal{P}$, then*

$$\log |U_i| \geq \sum_{j=0}^{w} \binom{n-1}{j} H(M).$$

These bounds are tight because the construction in the proof of Theorem 2.4.1 meets the equalities if we use the KPSs of Section 3.1 and Section 3.2.

2.4.3 A general lower bound on $|U_i|$

In this subsection, We generalize Proposition 2.3.1.

Lemma 2.4.1 *Let P and Q be distinct subsets of $\{1, 2, \ldots, n\}$. Let $F \triangleq \{1, 2, \ldots, n\} \setminus Q$. If $|Q| \leq |P|$, then*

$$F \cap P \neq \emptyset$$

(Proof) First, suppose that $|Q| < |P|$. If $F \cap P = \emptyset$, then

$$n \geq |F \cup P| = |F| + |P| = n - |Q| + |P| > n.$$

This is a contradiction. Therefore, $F \cap P \neq \emptyset$.

Next, suppose that $|Q| = |P|$. If $F \cap P = \emptyset$, then

$$|F \cup P| = |F| + |P| = n - |Q| + |P| = n.$$

Therefore,

$$F = \{1, 2, \ldots, n\} \setminus P.$$

This means that $P = Q = \{1, 2, \ldots, n\} \setminus P$. This is a contradiction. Hence, $F \cap P \neq \emptyset$.

□

Theorem 2.4.3 *In a $(\mathcal{P}, \mathcal{F})$-KPS,*

$$\log |U_i| \geq \delta_i \log |K|,$$

where

$$\delta_i = |\{P \mid i \in P \in \mathcal{P} \ , \ \{1, 2, \ldots, n\} \backslash P \in \mathcal{F}\}|.$$

Our proof is a generalization of the proof in [4, Theorem 3.1]. (Proof) For simplicity, we give a proof for $|U_1|$. Take

$$\tilde{P} \stackrel{\triangle}{=} \{P \mid 1 \in P \in \mathcal{P} \ , \ \{1, 2, \ldots, n\} \backslash P \in \mathcal{F}\}.$$

Let $l = \delta_1 = |\tilde{P}|$ and let $\tilde{P} = \{P_1, P_2, \ldots, P_l\}$, where $|P_1| \geq |P_2| \geq \cdots \geq |P_l|$. Let $\vec{u} = (u_1, \ldots, u_n)$ be a vector of secret information of the users such that

$$\Pr[U_U = \vec{u}] > 0.$$

We define $\vec{u}_F$ similarly.

For all $k_1 \in K_{P_1}$, for all F such that $P_1 \cap F_1 = \emptyset$ and for all $\vec{u}_F$,

$$\Pr[K_{P_1} = k_1 \mid U_F = \vec{u}_F] = \Pr[K_{P_1} = k_1] > 0,$$

from (2.2). Therefore, for all $k_1 \in K_{P_1}$ there is a $\vec{u} = (u_1, \ldots, u_n)$ such that the key of P_1 reconstructed from $\vec{u}$ is k_1. Now let $\vec{k} = (k_1, \ldots, k_l)$ be any vector in $K_{P_1} \times \cdots \times K_{P_l}$. We claim that there is a $\vec{u}$ such that the key of P_i reconstructed from $\vec{u}$ is k_i for $1 \leq i \leq l$.

Suppose that our claim is false. Let $h(\leq l)$ be the maximum index such that the keys of $\{P_i\}$ are $(k_1, \ldots, k_{h-1}, k'_h, \ldots, k'_l)$ by some $\vec{u}$, where $k'_h \neq k_h$. Then $2 \leq h$ from our discussion. Let

$$F_h \stackrel{\triangle}{=} \{1, 2, \ldots, n\} \setminus P_h.$$

Then from Lemma 2.4.1 (let $Q = P_h$ and $P = P_i$),

$$F_h \cap P_i \neq \emptyset \quad \text{for } 1 \leq i \leq h-1. \tag{2.6}$$

Let $\vec{u}_{F_h}$ be a subvector of $\vec{u}$ which corresponds to F_h. Then $\vec{u}_{F_h}$ can compute $k_1, \ldots, k_{h-1}$ from (2.6). Suppose that

$$\Pr[K_{P_h} = k_h | U_{F_h} = \vec{u}_{F_h}] > 0.$$

This means that there exists a $\vec{u}$ such that the keys are $k_1, \ldots, k_{h-1}, k_h$. This contradicts the maximality of h. Therefore,

$$\Pr[K_{P_h} = k_h | U_{F_h} = \vec{u}_{F_h}] = 0.$$

However, this is against (2.2).

Hence, for any $\vec{k} \in K_{P_1} \times \cdots \times K_{P_l}$, there exists a $\vec{u}$ such that the keys are $\vec{k}$. Remember that user 1 is included in any P_i from our definition of $\tilde{P}$. It follows that u_i must be distinct for each $\vec{k}$. Therefore,

$$|U_1| \geq |K_{P_1}| \times \cdots \times |K_{P_l}| = |K|^l.$$

Hence,

$$\log |U_1| \geq l \log |K| = \delta_1 \log |K|.$$

□

Note that Proposition 2.3.3 is also obtained as a corollary from Theorem 2.4.3. Indeed, all the previous bounds for KPSs are obtained as corollaries to Theorem 2.4.3.

From Theorem 2.4.2 and Theorem 2.4.3, we get the following corollary.

Corollary 2.4.3 *In a $(\mathcal{P}, \mathcal{F})$-OTCAS, if $|C_P| = |M|$ for all $P \in \mathcal{P}$, then*

$$\log |U_i| \geq \delta_i \log |M|,$$

where $\delta_i = |\{P \mid i \in P \in \mathcal{P} \ , \ \{1, 2, \ldots, n\} \backslash P \in \mathcal{F}\}|$.

2.5 Multiple use broadcast encryption

In this section we first show how to construct a computationally secure $(\mathcal{P}, \mathcal{F})$-Multiple use Conditional Access Scheme ($(\mathcal{P}, \mathcal{F})$-MCAS) from a $(\mathcal{P}, \mathcal{F})$-KPS by using the ElGamal cryptosystem. We then prove that our $(\mathcal{P}, \mathcal{F})$-MCAS is secure against chosen (message, privileged subset of users) attacks if the ElGamal cryptosystem is secure and if the original $(\mathcal{P}, \mathcal{F})$-KPS is simulatable. We also show that all the KPSs considered in Section 2.3 are simulatable. This construction is the first $(\mathcal{P}, \mathcal{F})$-MCAS whose security is proved formally. Furthermore, our technique can be generalized to many of the OTCAS presented in [58].

2.5.1 A proposed construction for $(\mathcal{P}, \mathcal{F})$-MCAS

Let $(U_1, \ldots, U_n, K)$ be a $(\mathcal{P}, \mathcal{F})$-KPS. The TA distributes secret information $u_1, \ldots, u_n$ to the users in the same way as for the $(\mathcal{P}, \mathcal{F})$-KPS. Let Q be a prime power such that $|K| \mid Q-1$. Let g be a primitive $|K|$-th root of unity over $GF(Q)$. All the participants agree on Q and g. Let

$$M \triangleq \langle g \rangle = \{m \mid m = g^x \text{ for some } x\}$$

If the TA wishes to send a message $m_p \in M$ to a privileged set $P \in \mathcal{P}$, then the TA broadcasts

$$c_P = (g^r, m_P g^{rk_P}),$$

where k_P is the key of the $(\mathcal{P}, \mathcal{F})$-KPS for P and r is a random number. Each member of P can decrypt c_P by using k_P with the ElGamal cryptosystem.

2.5.2 Security

Let $\vec{u}_F$ be a $\vec{u}_F \in U_F$ with $\Pr[U_F = \vec{u}_F] > 0$. We will show that the proposed construction is secure against chosen message attacks, in which the adversary can target privileged subsets of users adaptively. Informally these attacks are defined as follows. Fix a forbidden subset F (under the control of the adversary) arbitrarily. Suppose that F has obtained a broadcast c_P of a privileged subset P, $P \cap F = \emptyset$. Then F chooses several privileged subsets P_i and messages m_{P_i} adaptively, and can obtain from the TA, by using it as an oracle, the broadcast c_{P_i}, $i = 1, 2, \ldots$.

Definition 2.5.1 *A $(\mathcal{P}, \mathcal{F})$-MCAS is secure against chosen (message, privileged subset of users) attacks if there is no probabilistic polynomial time algorithm (adversary) A_0 such as follows. Give as input to A_0:*

$$Q, g, \tilde{F} \in \mathcal{F}, \vec{u}_{\tilde{F}}, \tilde{P} \in \mathcal{P}, c_{\tilde{P}} \in C_{\tilde{P}}$$

with $\tilde{F} \cap \tilde{P} = \emptyset$. A_0 then chooses $P_i \in \mathcal{P}$ and $m_i \in M$ adaptively, and sends these to the TA as a query for $i = 1, 2, \ldots, l$. The TA gives back $c_{P_i} \in C_{P_i}$ to A_0. Finally, A_0 outputs $m_{\tilde{P}}$ with non-negligble probability for all $(\tilde{F}, \tilde{P})$.

Definition 2.5.2 *We say that the ElGamal cryptosystem is secure if there is no probabilistic polynomial time algorithm A_1 which on input (Q, g, y, g^r, my^r) outputs m with non-negligible probability, where r is a random number and $y \in \langle g \rangle$.*

Definition 2.5.3 *We say that a $(\mathcal{P},\mathcal{F})$-KPS is simulatable if there is a probabilistic polynomial time algorithm (the simulator) B for which the following holds. On input $(Q,g,y,P \in \mathcal{P},\tilde{F} \in \mathcal{F})$ with $P \cap \tilde{F} = \emptyset$, B outputs $\vec{u}_{\tilde{F}}$, $g^{k_{P_1}},\ldots,g^{k_{P_h}}$ with probability*

$$\Pr[K_{P_1} = k_{P_1},\ldots,K_{P_h} = k_{P_h}, u_{\tilde{F}} = \vec{u}_{\tilde{F}} \mid K_P = k_P],$$

where $y = g^{k_P}$ and $\{P_1,\ldots,P_h\} = \{P_i \mid P_i \in \mathcal{P}, P_i \neq P, P_i \cap \tilde{F} = \emptyset\}$.

Theorem 2.5.1 *Suppose that a $(\mathcal{P},\mathcal{F})$-KPS is simulatable. Then the $(\mathcal{P},\mathcal{F})$-MCAS obtained by using this KPS in our construction is secure against chosen (message, privileged subset of users) attacks if the ElGamal cryptosystem is secure.*

(Proof) Suppose that a $(\mathcal{P},\mathcal{F})$-KPS is simulatable and that the proposed $(\mathcal{P},\mathcal{F})$-MCAS is not secure against chosen (message, privileged subset of users) attacks. Then there is a simulator B for the $(\mathcal{P},\mathcal{F})$-KPS, and an adversary A_0 which breaks $c_{\tilde{P}}$ for $\tilde{P} \in \mathcal{P}$ by controlling $\tilde{F} \in \mathcal{F}$ for some $\tilde{P} \cap \tilde{F} = \emptyset$.

We will describe a probabilistic polynomial time algorithm A_1 which breaks the ElGamal cryptosystem by using A_0 and B as subroutines. Let the input to A_1 be (Q,g,y,g^r,my^r). Then there is a $k_{\tilde{P}}$ such that $y = g^{k_{\tilde{P}}}$. A_1 works as follows.

1. A_1 gives $(Q,g,y,\tilde{P},\tilde{F})$ to B. Then B outputs $\vec{u}_{\tilde{F}}$, $g^{k_{P_1}},\ldots,g^{k_{P_h}}$.
2. A_1 gives $(Q,g,\tilde{F},\vec{u}_{\tilde{F}},\tilde{P},g^r,my^r)$ to A_0.
3. Since A_1 has $g^{k_{P_1}},\ldots,g^{k_{P_h}}$, A_1 can answer any query of A_0.
4. Finally, A_0 outputs m with non-negligible probability.

Then A_1 can output m with non-negligible probability. This is a contradiction. □

2.5.3 Simulatable $(\mathcal{P},\mathcal{F})$-KPSs

In what follows, we assume that $\binom{t+w-1}{t-1}$ is polynomial in the length of Q for the Blundo *et al.* scheme, that $\sum_{i=0}^{w}\binom{n-1}{i}$ is polynomial in the length of Q for the Fiat-Naor scheme, and that $2^{n-1}-1$ is polynomial in the length of Q for the Desmedt-Viswanathan scheme.

Theorem 2.5.2 *The Fiat-Naor scheme and the Desmedt-Viswanathan scheme are simulatable.*

(Proof) We give a proof for the Fiat-Naor scheme. The proof for the Desmedt-Viswanathan scheme is obtained in a similar way.

We shall describe a simulator B whose input is $(Q, g, y, P, \tilde{F})$, where $P \cap \tilde{F} = \emptyset$. B chooses s_{F_i} randomly for all $F_i \in \mathcal{F}$. From the $\{s_{F_i}\}$, B can obtain $\vec{u}_{\tilde{F}}$. Note that $s_{\tilde{F}} \not\in \vec{u}_{\tilde{F}}$. On the other hand,

$$k_P = \sum_{F:|F|\leq w, F\cap P=\emptyset} s_F = s_{\tilde{F}} + \sum_{F:F\neq\tilde{F},|F|\leq w, F\cap P=\emptyset} s_F \pmod{q-1}$$

Therefore,

$$\begin{aligned} y &= g^{k_P} = g^{s_{\tilde{F}}} \cdot g^{\sum_{F:F\neq\tilde{F},|F|\leq w,F\cap P=\emptyset} s_F}, \\ g^{s_{\tilde{F}}} &= y/g^{\sum_{F:F\neq\tilde{F},|F|\leq w,F\cap P=\emptyset} s_F}. \end{aligned}$$

Thus B can compute $g^{s_{\tilde{F}}}$ which is consistent with k_P such that $y = g^{k_P}$. Then B can compute $g^{k_{P_i}}$ for all $P_i \in \mathcal{P}$ because B knows $\{s_F \mid F \neq \tilde{F}, F \in \mathcal{F}\}$ and $g^{s_{\tilde{F}}}$. □

Definition 2.5.4 *Let* $A = \{a_{i_1\cdots i_t} \mid 0 \leq i_1 \leq w, \ldots, 0 \leq i_t \leq w\}$. *We say that* A *is symmetric if for any* $a_{i_1\cdots i_t} \in A : a_{i_1\cdots i_t} = a_{\pi(i_1\cdots i_t)}$ *for all permutations* π *of* $(i_1 \cdots i_t)$. *Furthermore, let*

$$f(x_1, \ldots, x_t) = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} a_{i_1\cdots i_t} x_1^{i_1} \cdots x_t^{i_t}.$$

We say that $f(x_1, \ldots, x_t)$ *is symmetric if* $\{a_{i_1\cdots i_t}\}$ *is symmetric.*

Lemma 2.5.1 *For given* $D = \{c_{j_1\cdots j_t} \mid 1 \leq j_1 \leq w+1, \ldots, 1 \leq j_t \leq w+1\}$, *let*

$$a_{i_1\cdots i_t} \triangleq \sum_{j_1=1}^{w+1} \cdots \sum_{j_t=1}^{w+1} c_{j_1\cdots j_t} w_{j_1 i_1} \cdots w_{j_t i_t},$$

where $[w_{ij}] \triangleq C^{-1}$ *and*

$$C \triangleq \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & w+1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^w & \cdots & (w+1)^b \end{pmatrix}.$$

Then

$$c_{j_1,\dots,j_t} = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} a_{i_1\cdots i_t} j_1^{i_1} \cdots j_t^{i_t}.$$

Furthermore, if D is symmetric, then $\{a_{i_1\cdots i_t}\}$ is symmetric.

Theorem 2.5.3 *The Blundo et al. scheme is simulatable.*

(Proof) For simplicity, suppose that the input to the simulator B is

$$\tilde{F} = \{1, 2, \dots, w\},\ P = \{v_1, \dots, v_t\},\ y = g^{k_P},\ Q,\ g.$$

B first chooses a (dummy) symmetric polynomial

$$f(x_1, \dots, x_t) = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} a_{i_1\cdots i_t} x_1^{i_1} \cdots x_t^{i_t},$$

randomly. Then $\vec{u}_{\tilde{F}} = (f(1, x_2, \dots, x_t), \dots, f(w, x_2, \dots, x_t))$. Next we consider a (real) symmetric polynomial

$$f_c(x_1, \dots, x_t) = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} \hat{a}_{i_1\cdots i_t} x_1^{i_1} \cdots x_t^{i_t} \tag{2.7}$$

such that $f_c(i, x_2, \dots, x_t) = f(i, x_2, \dots, x_t)$ for $1 \le i \le w$ and $f_c(v_1, \dots, v_t) = k_P$. We first show that there exists such a polynomial f_c. Let

$$J = \{(j_1 \cdots j_t) \mid 1 \le j_1 \le w+1, \dots, 1 \le j_t \le w+1\} \setminus \{(w+1 \cdots w+1)\}.$$

Then B can compute $c_{j_1\cdots j_t} = f_c(j_1, \dots, j_t)$ for all $(j_1 \cdots j_t) \in J$ by using $\vec{u}_{\tilde{F}}$. Let $c = f_c(w+1, \dots, w+1)$, where c is an unknown variable. From Lemma 2.5.1, B can compute $\{\hat{a}_{i_1\cdots i_t}\}$ from $\{c_{j_1\cdots j_t}\}$ and c. Further, it is easy to see that $\hat{a}_{i_1\cdots i_t}$ has the form

$$\hat{a}_{i_1\cdots i_t} = \alpha_{i_1\cdots i_t} + \beta_{i_1\cdots i_t} c, \tag{2.8}$$

for some constants $\alpha_{i_1\cdots i_t}$ and $\beta_{i_1\cdots i_t}$. Then from (2.7), we have

$$k_P = f_c(v_1, \dots, v_t) = e_0 + e_1 c$$

for some constants e_0 and e_1. This means that there exists such an f_c. Now

$$y = g^{k_P} = g^{e_0}(g^c)^{e_1}.$$

Then $g^c = (y/g^{e_0})^{1/e_1}$. Therefore B can compute $\{g^{\hat{a}_{i_1\cdots i_t}}\}$ from (2.8). Finally B can compute $g^{k_{P_i}}$ for all $P_i \in \mathcal{P}$ by using (2.7) and $\{g^{\hat{a}_{i_1\cdots i_t}}\}$. □

Corollary 2.5.1 *Suppose that the ElGamal cryptosystem is secure. The MCASs obtained from the Blundo et al. scheme, the Fiat-Naor scheme and the Desmedt-Viswanathan scheme by using our construction, are all secure against chosen (message, privileged subset of users) attacks.*

2.5.4 Generalization of our MCAS

We can generalize the MCASs in Corollary 2.5.1 so that anyone can do broadcast encryption. In the Fiat-Naor based MCAS, make each g^{s_F} public. In the Blundo *et al.* based MCAS, make each g^{a_i} public, where a_i is the coefficient of the symmetric polynomial f. Finally in the Desmedt-Viswanathan based MCAS, make each g^{k_P} public.

Chapter 3

Revocation Scheme with Small Overhead and Large Traceability

3.1 Introduction

In such applications, as pay TV, CD-ROM distribution and online databases, data should only be available to authorized users. To prevent unauthorized users from accessing data, the data supplier will encrypt data and provide only the authorized users with personal keys to decrypt it. However, some unauthorized users (*pirates*) may obtain some decryption keys from a group of one or more authorized users (*traitors*). Then the pirate users can decrypt data that they are not entitled to. To prevent this, Chor, Fiat and Naor [19] proposed w-resilient traceability schemes which reveal at least one traitor when a pirate decoder is confiscated if there are at most w traitors (CFN schemes). Their schemes are, however, non constructive. Recently, Stinson and Wei showed some explicit constructions by using combinatorial designs [59]. Although their constructions may not be as good asymptotically as those in [19], they are often better for small values of w and n.

On the other hand, it is often desirable for the center to be able to exclude certain users from recovering the message that is broadcast in encrypted form [27, 38]. We say that a broadcast encryption scheme is a (w, n)-revocation scheme if a center can exclude w or less users among n users. It was recently shown that a (w, n)-revocation scheme is obtained from a cover free family by Kumar et al. [38] and the authors [66] independently. Kumar et al. also presented a construction of cover free families such that

$$\rho_T = O(w^2),$$

which is independent of n, by using algebraic geometry codes, where

$$\rho_T \triangleq \text{(the length of a ciphertext)/(the length of a plaintext)}.$$

In this chapter, we present a construction of cover free families such that not only $\rho_T = O(w^2)$ but also they can be used as w-resilient traceability schemes. We also show an efficient (w, n)-revocation scheme such that

$$\rho_T = O(w^{1+\epsilon})$$

for any $\epsilon > 0$. The proposed constructions of cover free families use almost strongly universal hash functions. Our constructions are conceptually much simpler and much easier than that of Kumar et al. [38], which uses algebraic geometry codes.

The notion of universal classes of hash functions was introduced by Carter and Wegman [18]. It has found numerous applications in cryptography, complexity theory and other areas [18, 63, 57, 56, 7, 41] (see the Introduction in [57]). In particular, ϵ-almost strongly universal (ϵ-ASU) classes of hash functions have been studied and used for authentication codes [56]. The ϵ-ASU hash functions were also used for construction of identification codes via channels [41].

Throughout this chapter, we assume that there exist secure block ciphers.

Related works: Kurosawa and Desmedt found lower bounds on the size of keys and the size of ciphertexts of traceability schemes [39]. They also proposed two schemes, a one-time use (w, n)-traceability scheme (the KD one-time traceability scheme) which meets these bounds and a public key variant for multiple use (the KD public key traceability scheme) [39]. However, Stinson and Wei showed that the tracing algorithm of the KD schemes is subject to a linear attack [60]. Finally, Kurosawa et al. presented a proven secure tracing algorithm for their schemes [43]. However, their multiple use scheme assumes the difficulty of the discrete log problem, not the existence of secure block ciphers.

3.2 Previous works

A set system is a pair $(X, \mathcal{B})$, where $X \triangleq \{1, 2, \ldots, v\}$ and $\mathcal{B}$ is a set of blocks $B_i \subset X$ with $i = 1, 2, \ldots, n$. We consider a set system such that $|B_i| = k$ for $i = 1, 2, \ldots, n$.

Definition 3.2.1 *[25] We say that $\{X, \mathcal{B}\}$ is a (v, n, k, w, D)-cover free family if*

$$|B_{i_0} \setminus \bigcup_{j=1}^{w} B_{i_j}| \geq D$$

for $\forall B_{i_1}, \ldots, \forall B_{i_w}$ and for $\forall B_{i_0} \notin \{B_{i_1}, \ldots, B_{i_w}\}$.

Kumar et al. presented a construction of cover free families such that $\rho_T = O(w^2)$, which is independent of n, by using algebraic geometry codes.

On the other hand, a broadcast encryption scheme is said to have w-traceability if when a set of at most w authorized users (who are not necessarily excluded) pool their keys together to construct a "pirate decoder", at least one of the users (a traitor) involved can be identified from the decoder [19]. Stinson and Wei proposed a w-traceability scheme based on the following set system.

Definition 3.2.2 *[59] We say that $(X, \mathcal{B})$ is a w-(v, n, k) traceable set system if for $\forall B_{i_1}, \ldots, \forall B_{i_w}$ and for $\forall B_{i_0} \notin \{B_{i_1}, \ldots, B_{i_w}\}$,*

$$|F \cap B_{i_0}| < \max_{1 \leq j \leq w} |F \cap B_{i_j}| \tag{3.1}$$

for any $F \subseteq \bigcup_{j=1}^{w} B_{i_j}$ such that $|F| = k$.

In their w-traceability scheme, B_i corresponds to the key of user i and F corresponds to the pirate key as follows. Let $V = \{r_1, \ldots, r_v\}$ be a set of base keys, where r_i is a random element of $GF(p)$. Let $A_i \triangleq \{(j, r_j) \mid j \in B_i\}$ be the secret key of an authorized user i for $1 \leq i \leq n$.

The data supplier T chooses a random polynomial $f(x)$ over $GF(p)$ such that $\deg f(x) < k$ and $f(0) = s$, where s is a secret to be sent. Then T broadcasts the ciphertext

$$C = \{f(i) + r_i \mid i \in X\}.$$

Each authorized user i can compute the secret s from the ciphertext C by using A_i because $\deg f(x) < k$.

A pirate key e_p generated by w traitors $i_1, \ldots, i_w$ must be $e_p \subseteq \bigcup_{j=1}^{w} A_{i_j}$ such that $|e_p| \geq k$. Now if (3.1) is satisfied, then a traitor is detected by computing $\max_i |F \cap B_i|$.

3.3 Relationship among cover free family, Revocation Scheme and traceability scheme

In [38, page 614], it was remarked that cover free families could be used to construct traceability schemes. Actually, we can prove the following theorem.

Theorem 3.3.1 *If there exists a (v, n, k, w, D)-cover free family, then there exists a (w, n)-revocation scheme such that $\rho_T < v/D$. Further, if*

$$k < D + \lceil D/w \rceil, \tag{3.2}$$

then it can be used as a w-traceability scheme as well.

(Proof) Let $\{X, \mathcal{B}\}$ be a (v, n, k, w, D)-cover free family. First, we show a (w, n)-revocation scheme such that $\rho_T < v/D$. Let $V = \{r_1, \ldots, r_v\}$ be a set of base keys, where r_i is a random element of $GF(p)$. Let $A_i \triangleq \{(j, r_j) \mid j \in B_i\}$ be the secret key of an authorized user i for $1 \leq i \leq n$. Let $(m_0, \cdots, m_{D-1})$ be a plaintext to be sent, where $m_i \in GF(p)$. A center T constructs a polynomial $f(x)$ over $GF(p)$ such that

$$f(x) = m_0 + m_1 x + \cdots m_{D-1} x^D.$$

Suppose that w users $i_1, \cdots, i_w$ should be excluded. Then T broadcasts

$$C = \{f(i) + r_i \mid i \in X \setminus \bigcup_{j=1}^{w} B_{i_j}\}.$$

Each user $i_0 \not\in \{i_1, \cdots, i_w\}$ can compute at least D values of $f(i)$ from C by using A_i because

$$|B_{i_0} \setminus \bigcup_{j=1}^{w} B_{i_j}| \geq D.$$

Then he can compute the plaintext $(m_0, \cdots, m_{D-1})$ because $\deg f(x) < D$. On the other hand, the users $i_1, \cdots, i_w$ have no information on $(m_0, \cdots, m_{D-1})$ because each r_i with $i \in X \setminus i \in X \setminus \bigcup_{j=1}^{w} B_{i_j}\}$ is a random number. Finally, it is clear that $\rho_T < v/D$.

We next show that the $\{X, \mathcal{B}\}$ can be used as a w-traceability scheme as well. We consider Stinson-Wei w-traceability scheme such that $\deg f(x) < D$ instead of $\deg f(x) < k$. In this case, a pirate key e_p generated by w traitors $i_1, \ldots, i_w$ must be $e_p \subseteq \bigcup_{j=1}^{w} A_{i_j}$ such that $|e_p| \geq D$. In other words, a pirate key corresponds to $F \subseteq \bigcup_{j=1}^{w} B_{i_j}$ such that $|F| \geq D$. For any such F, it is clear that

$$\max_{1 \leq j \leq w} |F \cap B_{i_j}| \geq \lceil D/w \rceil.$$

Further, for any $i_0 \notin \{i_1, \cdots, i_w\}$, we have that

$$|F \cap B_{i_0}| \leq k - D.$$

because $F \subseteq \bigcup_{j=1}^{w} B_{i_j}$ and

$$|B_{i_0} \setminus \bigcup_{j=1}^{w} B_{i_j}| \geq D.$$

On the other hand, from (3.2),

$$k - D < \lceil D/w \rceil.$$

Hence, it holds that

$$|F \cap B_{i_0}| \leq k - D < \lceil D/w \rceil \leq \max_{1 \leq j \leq w} |F \cap B_{i_j}|.$$

Therefore, at least one traitor is detected by computing $\max_i |e_p \cap A_i|$. □

3.4 Proposed constructions of cover free families

In this section, we present three constructions of cover free families. The first construction yeilds w-revocation schemes such that not only $\rho_T = O(w^2)$ but also can be used as w-resilient traceability schemes. The second construction yeilds w-revocation schemes with $\rho_T = O(w^{1+\epsilon})$ for any $\epsilon > 0$. And the third construction yeilds w-revocation schemes such that $\rho_T = 1 + \epsilon$ for any $\epsilon > 0$ and the size of decryption keys is much smaller than the lower bound for w-revocation schemes with $\rho_T = 1$ obtained in Chapter 2. The first two proposed constructions use almost strongly universal hash functions and the third construction uses t-designs. Our constructions are conceptually much simpler and much easier than that of Kumar et al. [38], which uses algebraic geometry codes.

3.4.1 Construction using ASU hash families

Let S and T be finite sets such that $|S| \geq |T|$. Let H be a set of functions such that $h : S \longrightarrow T$ for each $h \in H$. Let $|H| = v$, $|S| = m$, $|T| = N$.

Definition 3.4.1 *[18] We say that H is an* ϵ-almost strongly universal *(ϵ-$ASU(v, m, N)$) hash function family provided that the following two conditions are satisfied:*

1. *for any $s \in S$ and any $t \in T$, there exist exactly $|H|/|T|$ functions $h \in H$ such that $h(s) = t$.*
2. *for any two distinct elements $s_1, s_2 \in S$ and for any two (not necessarily distinct) elements $t_1, t_2 \in T$, there exist at most $\epsilon|H|/|S|$ functions $h \in H$ such that $h(s_i) = t_i$, $i = 1, 2$.*

Theorem 3.4.1 *If there exists an ϵ-$ASU(v, m, N)$ hash function family H, then there exists a (v, n, k, w, D)-cover free family such that $n = mN$, $k = v/N$ and*

$$D = \frac{v}{N}(1 - w\epsilon).$$

(Proof) Let a universe X be H. Index each block $B \in \mathcal{B}$ by $(s, t) \in S \times T$ and define

$$B_{(s,t)} \stackrel{\Delta}{=} \{h \in H \mid h(s) = t\}.$$

Then it is clear that the number of blocks is $n = mN$. From the definition of ϵ-ASU hash families, we have

$$k = |B_{(s,t)}| = |H|/|T| = v/N,$$

$$|B_{(s,t)} \cap B_{(s',t')}| \leq \epsilon k.$$

Finally, for any $B_{i_0}, B_{i_1}, \ldots, B_{i_w} \in \mathcal{B}$,

$$|B_{i_0} \cap (\bigcap_{j=1}^{w} B_{i_j})| \leq \sum_{j=1}^{w} |B_{i_0} \cap B_{i_j}| \leq w\epsilon k.$$

Therefore,

$$|B_{i_0} \setminus (\bigcap_{j=1}^{w} B_{i_j})| \geq k - w\epsilon k = \frac{n}{N}(1 - w\epsilon).$$

□

Theorem 3.4.2 *There exists a ϵ-$ASU(v, m, N)$ hash function family such that $v = q^{l+2}$, $N = q$, $m = q^{lq^t}$ and*

$$\epsilon = \frac{l}{q} + \frac{1}{q^{l-t}} - \frac{1}{q^l}$$

for $1 \leq \forall t < \forall l < \forall q =$ prime power.

The proof is given in the following section.

Corollary 3.4.1 *There exists a (v, n, k, w, D)-cover free family such that $v = q^{l+2}$, $n = q^{lq^t+1}$, $k = q^{l+1}$ and $D = q^{l+1} - w(lq^l + q^{t+1} - q)$ for $1 \leq \forall t < \forall l < \forall q =$ prime power.*

(Proof) From Theorem 3.4.1 and Theorem 3.4.2. □

Corollary 3.4.2 *There exists a $(q^4, q^{2q+1}, q^3, w, D)$-cover free family such that*

$$D = q^3 - w(3q^2 - q)$$

for $\forall q =$ prime power.

(Proof) In Corollary 3.4.1, let $l = 2$ and $t = 1$. □

Theorem 3.4.3 *There exists a (w, n)-revocation scheme such that $\rho_T = O(w^{1+\epsilon})$ for any $\epsilon > 0$.*

(Proof) Consider the $(q^4, q^{2q+1}, q^3, w, D)$-cover free family such that

$$D = q^3 - w(3q^2 - q)$$

of Corollary 3.4.2, where q is a prime power. For given $\epsilon > 0$, let τ be such that

$$1 - \tau = 1/(1 + \epsilon).$$

Let q be large enough so that

$$q^\tau - 3 \geq q^\tau/2.$$

Let $w = \lfloor q^{1-\tau} \rfloor$. Then we have

$$\begin{aligned} q - 3w &\geq q - 3q^{1-\tau} \\ &= q^{1-\tau}(q^\tau - 3) \\ &\geq q^{1-\tau}(q^\tau/2) \\ &= q/2. \end{aligned}$$

Therefore,

$$\begin{aligned} D &= q^3 - w(2q^2 + q^2 - q) \\ &> q^3 - 3wq^2 \\ &\geq q^3/2. \end{aligned}$$

Hence from Theorem 3.3.1, there exists a (w, n)-revocation scheme such that

$$\rho_T < v/D < q^4/(q^3/2) = 2q = O(w^{1/(1-\tau)}) = O(w^{1+\epsilon}).$$

□

Theorem 3.4.4 *Let q be a prime power and let $w = \lfloor \sqrt{q}/2 \rfloor$. Then there exists a (w, n)-revocation scheme such that $\rho_T = O(w^2)$. Further, it can be used as a w-traceability scheme as well.*

(Proof) Consider the $(q^4, q^{2q+1}, q^3, w, D)$-cover free family such that

$$D = q^3 - w(3q^2 - q)$$

of Corollary 3.4.2. Suppose that $q > 12$. Then as in the proof of Theorem 3.4.3, we can show that

$$D > q^3/2.$$

Hence from Theorem 3.3.1, there exists a (w, n)-revocation scheme such that

$$\rho_T < v/D < q^4/(q^3/2) = 2q = O(w^2).$$

We next show that it can be used as a w-traceability scheme as well. It is enough to show (3.2) from Theorem 3.3.1. Since $q > 12$, it holds that

$$\sqrt{q}/2 - 3 > 0.$$

Since

$$D > q^3 - 3wq^2,$$

we have

$$\begin{aligned} D + \lceil D/w \rceil - k &> (q^3 - 3wq^2) + (q^3/w - 3q^2) - q^3 \\ &= q^2(-3w + q/w - 3) \\ &\geq q^2(-3\sqrt{q}/2 + 2\sqrt{q} - 3) \\ &= q^2(\sqrt{q}/2 - 3) \\ &> 0. \end{aligned}$$

Therefore, (3.2) is satisfied. □

3.4.2 Construction using t-designs

In this section, we show a construction of cover free families which yeild w-revocation schemes such that $\rho_T = 1 + \epsilon$ for any $\epsilon > 0$ and the size of decryption keys is much smaller than the lower bound for w-revocation schemes with $\rho_T = 1$ obtained in Chapter 2. This implies that the key size can be significantly reduced is the transmission overhead is sligtly larger than optimal.

Definition 3.4.2 *A t-(n, m, λ) design is a pair $(X, \mathcal{A})$ where X is an n-element set of points and $\mathcal{A}$ is a collection of m-element subsets of X (blocks) with the property that every t-element subset of X is contained in exactly λ blocks.*

Proposition 3.4.1 *In a t-(n, m, λ)-design $(X, \mathcal{A})$,*

$$|\mathcal{A}| = \lambda \binom{n}{t} / \binom{m}{t}.$$

Proposition 3.4.2 *If $(X, \mathcal{A})$ is a t-(n, m, λ) design and S is any s-element subset of X, with $0 \leq s \leq t$, then the number of blocks containing S is*

$$b_s = |\{A \in \mathcal{A} \mid S \subseteq A\}| = \lambda \binom{n-s}{t-s} / \binom{m-s}{t-s}.$$

Now we show our construction of superimposed distance families from a t-(n, m, λ) design $(X, \mathcal{A})$. Let

$$v = |\mathcal{A}| = \lambda \binom{n}{t} / \binom{m}{t}.$$

Let $X = \{1, 2, \ldots, n\}$ and let $\mathcal{A} = \{A_1, A_2, \ldots, A_v\}$. Define

$$B_i = \{j \mid i \notin A_j\}$$

for $1 \leq i \leq n$.

Example 3.4.1 *Consider a* 2-(9, 3, 1) *design (BIBD) such that*

$$\begin{array}{rclrclrcl} A_1 &=& \{1,2,3\}, & A_2 &=& \{1,4,7\}, & A_3 &=& \{1,5,9\}, \\ A_4 &=& \{1,6,8\}, & A_5 &=& \{2,4,9\}, & A_6 &=& \{2,5,8\}, \\ A_7 &=& \{2,6,7\}, & A_8 &=& \{3,4,8\}, & A_9 &=& \{3,5,7\}, \\ A_{10} &=& \{3,6,9\}, & A_{11} &=& \{4,5,6\}, & A_{12} &=& \{7,8,9\}. \end{array}$$

Then the following (12, 9, 4, 1, 3)*-cover free family is obtained.*

$$\begin{array}{rcl} B_1 &=& \{5,6,7,8,9,10,11,12\}, \\ B_2 &=& \{2,3,4,8,9,10,11,12\}, \\ B_3 &=& \{2,3,4,5,6,7,11,12\}, \\ B_4 &=& \{1,3,4,6,7,9,10,12\}, \\ B_5 &=& \{1,2,4,5,7,8,10,12\}, \\ B_6 &=& \{1,2,3,5,6,8,9,12\}, \\ B_7 &=& \{1,3,4,5,6,8,10,11\}, \\ B_8 &=& \{1,2,3,5,7,9,10,11\}, \\ B_9 &=& \{1,2,4,6,7,8,9,11\}. \end{array}$$

Theorem 3.4.5 *The* $\{B_1, B_2, \ldots, B_n\}$ *of the above construction is a* $(v, n, k, t-1, D)$*-cover free family such that*

$$\begin{aligned} v &= \lambda\binom{n}{t}/\binom{m}{t}, \\ k &= \binom{n-1}{t-1}\cdot\lambda\cdot\frac{n-m}{m}\cdot\frac{1}{\binom{m-1}{t-1}}, \\ D &= \lambda\cdot\frac{n-m}{m-t+1} \end{aligned}$$

(Proof) From Proposition 3.4.1,

$$v = |\mathcal{A}| = \lambda\binom{n}{t}/\binom{m}{t}.$$

From Proposition 3.4.2,

$$\begin{aligned} k &= |\{j \mid i \notin A_j\}| \\ &= |\mathcal{A}| - |\{j \mid i \in A_j\}| \\ &= v - \lambda\binom{n-1}{t-1}/\binom{m-1}{t-1} \\ &= \binom{n-1}{t-1}\cdot\lambda\cdot\frac{n-m}{m}\cdot\frac{1}{\binom{m-1}{t-1}}. \end{aligned}$$

For $\forall B_{i_0}, \forall B_{i_1}, \ldots, \forall B_{i_w}$, from Proposition 3.4.2 and the definition of t-design,

$$\begin{aligned} D = \left|B_{i_0} \setminus \bigcup_{l=1}^{w} B_{i_l}\right| &= |\{j \mid i_0 \notin A_j\} \setminus \bigcup_{l=1}^{t-1}\{j \mid i_l \notin A_j\}| \\ &= \left|\{j \mid i_0 \notin A_j\} \setminus \left(\bigcap_{l=1}^{t-1}\{j \mid i_l \in A_j\}\right)^c\right| \\ &= |\{j \mid i_0 \notin A_j\} \setminus \{j \mid \{i_1, \ldots, i_{t-1}\} \subseteq A_j\}^c| \\ &= |\{j \mid i_0 \notin A_j, \{i_1, \ldots, i_{t-1}\} \subseteq A_j\}| \\ &= |\{j \mid \{i_1, \ldots, i_{t-1}\} \subseteq A_j\} \setminus \{j \mid \{i_0, i_1, \ldots, i_{t-1}\} \subseteq A_j\}| \\ &= |\{j \mid \{i_1, \ldots, i_{t-1}\} \subseteq A_j\}| - |\{j \mid \{i_0, i_1, \ldots, i_{t-1}\} \subseteq A_j\}| \\ &= \lambda(n-t+1)/(m-t+1) - \lambda \end{aligned}$$

□

Corollary 3.4.3 *If there exists a t-(n, m, λ) design, then there exists a $(v, n, k, t-1, D)$-cover free family such that v, k and D are given by Theorem 3.4.5.*

We next show this construction yeilds w-revocation schemes such that $\rho_T = 1+\epsilon$ for any $\epsilon > 0$ and the size of decryption keys is much smaller than the lower bound for w-revocation schemes with $\rho_T = 1$ obtained in Chapter 2.

Theorem 3.4.6 *If there exists a t-(n, m, λ) design, then there exists a w-revocation scheme such that*

$$\begin{aligned} w &= t-1, \\ \rho_T &= \frac{n-w}{n-m}, \\ \rho_I &= \binom{n-1}{w}\frac{m-w}{m} \cdot \frac{1}{\binom{m-1}{w}} \end{aligned}$$

where

$$\rho_I \triangleq \frac{\text{the length of a decryption key}}{\text{the length of a plaintext}}.$$

(Proof) From Theorem 3.3.1 and Corollary 3.4.3,

$$\rho_I = \frac{k}{D} = \binom{n-1}{w}\frac{m-w}{m} \cdot \frac{1}{\binom{m-1}{w}}.$$

Next, from our construction and Proposition 3.4.2,

$$\begin{aligned} \left|\left(\bigcup_{l=1}^{b} B_{i_m}\right)^c\right| &= |\bigcap_{l=1}^{w}(B_{i_l})^c| \\ &= |\bigcap_{l=1}^{w}\{j \mid i_l \in A_j\}| \\ &= |\{j \mid \{i_1, \ldots, i_w\} \subseteq A_j\}| \\ &= \lambda \cdot \frac{n-w}{m-w}. \end{aligned}$$

Therefore,

$$\rho_T = \frac{\left|\left(\bigcup_{l=1}^{b} B_{i_l}\right)^c\right|}{D} = \frac{n-w}{n-m}.$$

□

Now from Theorem 3.4.6, we see that :

1. $\rho_T \approx 1$ if $n \gg w$ and $n \gg m$.
2. It alwasy holds that $\rho_I < \binom{n-1}{w}$ because $t \leq m$ and hence $w \leq m-1$.

This means that the size of decryption keys can be much smaller than the bound for w-revocation schemes with $\rho_T = 1$ obtained in Chapter 2 if $n \gg w$ and $n \gg m$. There exist such t-designs for any t [21, p.51]. Further, $\rho_I \ll \binom{n-1}{w}$ if $t \ll m$.

3.5 Proof of Theorem 3.4.2

Stinson showed a composition construction of an ϵ-ASU class of hash functions such as follows [56, Theorem 5.5].

Definition 3.5.1 *Let $C = (n, |C|, d)$ be an error correcting code over an alphabet S. Let H be an ϵ-ASU class of hash functions from S to T. Then for all i with $1 \leq i \leq n$ and $\forall h \in H$, define a hash function $g_{(i,h)} : \{1, 2, \ldots, |C|\} \to T$ by the rule*

$$g_{(i,h)}(x) = h(\textit{the ith symbol of the xth codeword of } C).$$

Let

$$H * C \triangleq \{g_{(i,h)}\}.$$

Proposition 3.5.1 *[56, Theorem 5.5] $H_C \triangleq H * C$ (defined as above) is an $\tilde{\epsilon}$-ASU$(n|H|, |C|, |T|)$ class of hash functions from $\{1, 2, \ldots, |C|\}$ to B such that*

$$\tilde{\epsilon} = \epsilon + 1 - \frac{d}{n},$$

Remark. In [56, Theorem 5.5], Stinson used the term *AU* class of hash functions. Bierbraur pointed out that it is equivalent to an error correcting code [7].

Let q be a prime power and let $1 < k < q$. Let

$$A \triangleq \{(a_1, \ldots, a_k) \mid a_i \in GF(q)\}.$$

$$B \triangleq \{\text{the elements of } GF(q)\}.$$

In [23], den Boer described the following ϵ-ASU class of hash functions from A to B.

Definition 3.5.2 *For* $\forall(e_0, e_1)$ *such that* $e_0, e_1 \in GF(q)$, *let*

$$h_{(e_0,e_1)}(a_1, \ldots, a_k) = e_0 + a_1e_1 + \cdots + a_ke_1^k.$$

Let

$$G(q, k) \triangleq \{h_{(e_0,e_1)}\}.$$

Proposition 3.5.2 *[23] The above* $G(q, k)$ *is a* (k/q)*-*$ASU(q^2, q^k, q)$ *class of hash functions from* A *to* B *such that* $|G(q, k)| = q^2$.

Then the following corollary is obtained from Proposition 3.5.1.

Corollary 3.5.1 *Let* $G(q, k)$ *be a* (k/q)*-*ASU *class of hash functions from* A *to* B *defined as above. Let* $C = (n, |C|, d)$ *be an error correcting code over* $GF(q^k)$. *Then*

$$G(q, k)_C \triangleq G(q, k) * C$$

is an $\tilde{\epsilon}$*-*$ASU(nq^2, |C|, q)$ *class of hash functions from* $\{1, 2, \ldots, |C|\}$ *to* B *such that*

$$\tilde{\epsilon} = \frac{k}{q} + 1 - \frac{d}{n},$$

$$|G(q, k)_C| = nq^2.$$

A $[q^k, q^t]$ Reed-Solomon code is a code over $GF(q^k)$ such that the length of a codeword is $n = q^k$, the number of codewords is $|C| = (q^k)^{q^t}$ and the minimum Hamming distance is $d = q^k - q^t + 1$. Finally, from Corollary 3.5.1, we obtain Theorem 3.4.2.

Chapter 4

Linear Code Implies Public-Key Traitor Tracing

4.1 Introduction

In such applications as pay TV, CD-ROM distribution and online databases, data should only be available to authorized users. To prevent unauthorized users from accessing data, the data supplier will encrypt data and provide only the authorized users with personal keys to decrypt it. However, some authorized users (*traitors*) may create a pirate decoder.

A (w, n)-traceability scheme is a scheme in which at least one traitor is detected from a confiscated pirate decoder if there are at most w traitors among n authorized users. Chor, Fiat and Naor [19] introduced the first (w, n)-traceability scheme. Their scheme is, however, non-constructive. Stinson and Wei showed some explicit constructions by using combinatorial designs [59]. In the above two schemes, a private-key encryption scheme is used to encrypt a session key.

On the other hand, the first public-key (w, n)-traceability scheme was shown by Kurosawa and Desmedt [39, section 5]. That is, anyone can broadcast encrypted data to authorized users. Although Shamir's $(w+1, n)$-threshold secret sharing scheme was used in their original scheme, we should use Shamir's $(2w - 1, n)$-threshold secret sharing scheme to avoid a linear attack given by [60]. We call such a corrected scheme the corrected KD scheme.

After that, Boneh and Franklin presented another public-key (w, n)-traceability scheme [14]. Only the above two schemes are known as public-key (w, n)-traceability schemes currently.

In this chapter, we first show that three public-key (w, n)-traceability schemes can be derived from an $[n, u, d]$-linear code $\mathcal{C}$ such that $d \geq 2w + 1$. We call them linear coded KD scheme (LC-KD scheme), linear coded BF scheme (LC-BF scheme) and linear coded KD$'$ scheme (LC-KD$'$ scheme), respectively. The previous schemes are obtained as special cases. This observation gives a more freedom and a new insight to the study of this field.

For example, we show that Boneh-Franklin scheme (BF scheme) is equivalent to a slight modification of the corrected KD scheme. (We call it modified KD scheme. It will be given in Section 4.5.3.) This means that BF scheme is redundant or overdesigned because modified KD scheme is much simpler. Indeed, BF scheme must use a public code matrix Γ and $2w$ additional secret random numbers $\beta_1, \cdots, \beta_{2w}$ which modified KD scheme does not require. More generally, we prove the equivalence between LC-BF scheme and LC-KD$'$ scheme.

We also show that LC-KD scheme is better than LC-KD$'$ scheme from a view point of key generation. This implies that the corrected KD scheme is better than modified KD scheme from a view point of key generation. Since modified KD scheme is better than BF scheme as shown above, we see that the corrected KD scheme is the best among them.

We finally prove the secrecy and the black box traceability of LC-KD scheme under the decision Deffie-Hellman assumption. Those of LC-KD$'$ scheme and LC-BF scheme are proved similarly. The tracing algorithm of BF scheme for any pirate decoder is obtained as a special case. (It is not written clearly in the original paper [14]. It is not written at all in their latest version [15].)

Generalized Scheme	Original Scheme
LC-KD scheme	corrected KD scheme
LC-KD$'$ scheme	modified KD scheme
LC-BF scheme	BF scheme

4.2 Preliminaries

4.2.1 Notation

An $[n, u, d]$-linear code is a linear code of length n, dimension u and the minimum Hamming distance d.

Let $q > n$ be a prime. Let G_q be a group of prime order q. Let $g \in G_q$ be a generator of G_q. For example, G_q is a subgroup of Z_p^* of order q, where $q \mid p - 1$. Alternatively, we can use an elliptic curve over a finite field.

$\cdot$ denotes the inner product of two vectors over $GF(q)$.

4.2.2 DDH Assumption

The decision Diffie-Hellman assumption (DDH) assumption says that no polynomial statistical test can distinguish with non negligible advantage between the two distributions $\mathbf{D} = (g, g^r, y, y^r)$ and $\mathbf{R} = (g, g^r, y, v)$, where g, y, v are chosen at random from G_q and r is chosen at random in Z_q.

4.2.3 Model of Traitor Tracing

In the model of traceability schemes, there are a data supplier T, a set of n authorized users and a pirate user. Some authorized users are malicious and they are called *traitors*. The traitors create a pirate key e_p. The pirate key is used in a pirate decoder.

Suppose that there are at most w traitors. Then a (w, n)-traceability scheme is a scheme such that at least one traitor is detected from a confiscated pirate decoder. A (w, n)-traceability scheme has four components.

Key generation: The key generation algorithm $\mathcal{K}$ is a probabilistic polynomial time algorithm that outputs $(e_T, e_1, \cdots, e_n)$ on input 1^l, where l is the security parameter. e_T is the broadcast encryption key of the data supplier T and e_i is the personal decryption key of authorized user i.

T runs $\mathcal{K}$ and sends e_i to authorized user i secretly.

Encryption: The encryption algorithm $\mathcal{E}$ is a probabilistic polynomial time algorithm that takes an encryption key e_T and a session key s to return a header h; we write

$$h \overset{R}{\leftarrow} e_T(s).$$

The data m is encrypted by using a secure symmetric encryption function E with the session key s as $E_s(m)$. Finally, T broadcasts $(h, E_s(m))$.

Decryption: Then decryption algorithm $\mathcal{D}$ is a deterministic algorithm that takes the personal decryption key e_i and a header h to return the session key s; we write

$$s \leftarrow e_i(h).$$

Each authorized user i can recover s from h by using his personal key e_i and then decrypt $E_s(m)$ to obtain the data m.

Tracing: T can detect at least one traitor from a pirate key e_p by using a tracing algorithm.

We have *black box* traceability if the pirate decoder can only be used as an oracle. That is, the tracing algorithm cannot examine the pirate key e_p. For black box tracing, we shall assume that the pirate decoder is resettable to its initial state, as in [20].

In what follows, a session key s is chosen from G_q.

4.3 Previous Public-Key (w, n) Traceability Schemes

4.3.1 Corrected Kurosawa-Desmedt Scheme

Key generation: The data supplier T chooses a uniformly random polynomial $f(x) = a_0 + a_1x + \cdots + a_{2w-1}x^{2w-1}$ over $GF(q)$. Then T gives to each authorized user i the personal decryption key $e_i = f(i)$, where $i = 1, 2, \ldots, n$. He next publishes g and $y_0 = g^{a_0}, y_1 = g^{a_1}, \ldots, y_{2w-1} = g^{a_{2w-1}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as $h = (g^r, sy_0^r, y_1^r, \ldots, y_{2w-1}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using $f(i)$.

$$s = U/(g^r)^{f(i)}, \text{ where } U = sy_0^r \prod_{j=1}^{2w-1} (y_j^r)^{i^j}.$$

4.3.2 Boneh-Franklin Scheme

BF scheme makes use of a public code matrix Γ defined as follows. Consider the following $(n-2w) \times n$ matrix G:

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ 1^2 & 2^2 & 3^2 & \cdots & n^2 \\ 1^3 & 2^3 & 3^3 & \cdots & n^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1^{n-2w-1} & 2^{n-2w-1} & 3^{n-2w-1} & \cdots & n^{n-2w-1} \end{pmatrix} \pmod q$$

Let $w_1, \ldots, w_{2w}$ be a basis of the linear space of vectors satisfying

$$G\mathrm{x} = 0 \bmod q. \tag{4.1}$$

Viewing these $2w$ vectors as the columns of a matrix, we obtain an $n \times 2w$ matrix Γ:

$$\Gamma = \begin{pmatrix} | & | & | & & | \\ w_1 & w_2 & w_3 & \cdots & w_{2w} \\ | & | & | & & | \end{pmatrix}$$

Define the code as the set of rows of the matrix Γ. Hence, it consists of n codewords each of length $2w$.

Key Generation: For $i = 1, \ldots, 2w$, the data supplier chooses a random $a_i \in Z_q$ and compute $y_i = g^{a_i}$. Then T computes $z = \prod_{i=1}^{2w} y_i^{\beta_i}$ for random $\beta_1, \ldots, \beta_{2w} \in Z_q$ and publishes $z, y_1, \ldots, y_{2w}$ as the public key. The personal decryption key of user i is computed as

$$\theta_i = (\sum_{j=1}^{2w} a_j \beta_j) / (\sum_{j=1}^{2w} a_j \gamma_j) \pmod q,$$

where $\gamma^{(i)} = (\gamma_1, \ldots, \gamma_{2w}) \in \Gamma$ is the i'th codeword of Γ.

Encryption: For a session key $s \in G_q$, T computes a header as $h = (sz^r, y_1^r, \ldots, y_{2w}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using θ_i.

$$s = sz^r / U^{\theta_i}, \text{ where } U = \prod_{j=1}^{2w} (y_j^r)^{\gamma_j}.$$

Remark. In the key generation, $a_1, \cdots, a_{2w}$ must be chosen so that $\sum_{j=1}^{2w} a_j \gamma_j \neq 0 \pmod{q}$ for $i = 1, \cdots, n$. This was overlooked in [14].

4.4 Linear Code Implies Public-Key Traitor Tracing

This section shows that if there exists an $[n, u, d]$-linear code C such that $d \geq 2w+1$, then three public-key (w, n)-traceability schemes are derived. We call them linear coded KD scheme (LC-KD scheme), linear coded BF scheme (LC-BF scheme) and linear coded KD′ scheme (LC-KD′ scheme), respectively. The corrected KD scheme and the original BF scheme are obtained as special cases.

Let H be a parity check matrix of an $[n, u, d]$-linear code over $GF(q)$ such that $d \geq 2w + 1$. Any $2w$ columns of H are linearly independent because $d \geq 2w + 1$. This property plays a central role in the proof of traceability of our schemes.

We assume that H is publicly known. Note that H is an $(n - u) \times n$ matrix over $GF(q)$. Let the ith column of H be $\mathbf{b}_i = (b_{1,i}, b_{2,i}, \cdots, b_{n-u,i})^T$.

4.4.1 LC-KD Scheme

Assume that the first row of H is $(1, \cdots, 1)$.

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random Let $(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H$. T gives e_i to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_1 = g^{a_1}, y_2 = g^{a_2}, \ldots, y_{n-u} = g^{a_{n-u}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as $h = (g^r, sy_1^r, y_2^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using e_i.

$$s = U/(g^r)^{e_i}, \text{ where } U = sy_1^r \prod_{j=2}^{n-u} (y_j^r)^{i^j}. \tag{4.2}$$

The tracing algorithm will be given in Section 4.7.

4.4.2 LC-BF Scheme

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random in such a way that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ for $i = 1, \cdots, n$. Let $y_i = g^{a_i}$. Then T computes $z = \prod_{i=1}^{n-u} y_i^{\beta_i}$ for random $\beta_1, \ldots, \beta_{n-u} \in Z_q$ and publishes $z, y_1, \ldots, y_{n-u}$ as the public key. The personal decryption key of user i is computed as

$$\theta_i = (a_1, \ldots, a_{n-u}) \cdot (\beta_1, \ldots, \beta_{n-u}) / (a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i. \tag{4.3}$$

Encryption: For a session key $s \in G_q$, T computes a header as $h = (sz^r, y_1^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using θ_i.

$$s = sz^r / U^{\theta_i}, \text{ where } U = \prod_{j=1}^{n-u} (y_j^r)^{b_{j,i}}. \tag{4.4}$$

4.4.3 LC-KD′ Scheme

This is a slight modification of LC-KD scheme.

Key Generation: The data supplier T chooses $(a_1, \cdots, a_{n-u})$ uniformly at random in such a way that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ for $i = 1, \cdots, n$. Let $(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H$. (Note that $e_i \neq 0$ for $i = 1, \cdots, n$.) T gives e_i to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_1 = g^{a_1}, y_2 = g^{a_2}, \ldots, y_{n-u} = g^{a_{n-u}}$ as the public key.

Encryption: For a session key $s \in G_q$, T computes a header as $h = (sg^r, y_1^r, y_2^r, \ldots, y_{n-u}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows.

$$s = sg^r / U^{1/e_i}, \text{ where } U = \prod_{j=1}^{n-u} (y_j^r)^{b_{j,i}}. \tag{4.5}$$

Remark. In h, s is multiplied to g^r in LC-KD′ scheme while it is multiplied to y_1^r in LC-KD scheme.

4.5 Relationship with the Original Schemes

4.5.1 Corrected KD Scheme

Let $\mathcal{C}$ be an $[n, n-2w, d]$-Reed Solomon code over $GF(q)$, where $d = 2w+1$. Then it is clear that the corrected KD scheme is obtained from LC-KD scheme as a special case.

4.5.2 BF Scheme

In BF scheme, note that G (shown in Section 4.3.2) is a generator matrix of an $[n, n-2w, d]$ Reed-Solomon code over $GF(q)$. Further we see that $G \cdot \Gamma = \mathcal{O}$ from (4.1). Hence Γ^T is a parity check matrix of the Reed-Solomon code $\mathcal{C}$. This implies that the original BF scheme is obtained from LC-BF scheme as a special case.

4.5.3 Modified KD Scheme

In LC-KD$'$ scheme, let $\mathcal{C}$ be an $[n, n-2w, d]$-Reed Solomon code over $GF(q)$, where $d = 2w+1$. Then the following scheme is obtained. We call it modified KD scheme because it is a slight modification of the corrected KD scheme.

Key Generation: The data supplier T chooses a uniformly random polynomial $f(x) = a_0 + a_1x + \cdots + a_{2w-1}x^{2w-1}$ over $GF(q)$ such that $f(i) \neq 0$ for $i = 1, \cdots, n$. Then T gives $f(i)$ to authorized user i as the personal decryption key for $i = 1, 2, \ldots, n$. He next publishes $y_0 = g^{a_0}, y_1 = g^{a_1}, \ldots, y_{2w-1} = g^{a_{2w-1}}$.

Encryption: For a session key $s \in G_q$, T computes a header as $h = (sg^r, y_0^r, y_1^r, \ldots, y_{2w-1}^r)$, where r is a random number. T broadcasts h.

Decryption: Each user i computes s from h as follows by using $f(i)$.

$$s = sg^r / U^{1/f(i)}, \text{ where } U = \prod_{j=0}^{2w-1} (y_j^r)^{i^j}. \tag{4.6}$$

Remark. In h, s is multiplied to g^r in modified KD scheme while it is multiplied to y_1^r in the corrected KD scheme.

4.6 Equivalence

4.6.1 LC-BF Scheme = LC-KD$'$ Scheme

LC-BF scheme is more complicated than LC-KD$'$ scheme because it uses secret random numbers $\beta_1, \cdots, \beta_{n-u}$ which LC-KD$'$ scheme does not use. Nevertheless, we show that they are equivalent. This means that LC-BF scheme is redundant or overdesigned.

Public-key equivalence: In the key generation of LC-BF scheme, let

$$c = \sum_{i=1}^{n-u} a_i \beta_i.$$

For any fixed $(a_1, \cdots, a_{n-u})$, it is easy to see that $\Pr[c \neq 0] = 1 - (1/q)$. Therefore, we assume that $c \neq 0$ in what follows.

The public key of LC-BF scheme is $pk = (z, y_1, \ldots, y_{n-u})$. First since q is a prime and $z \in G_q$, z is a generator of G_q. Next note that

$$z = \prod_{i=1}^{n-u} y_i^{\beta_i} = \prod_{i=1}^{n-u} g^{a_i \beta_i} = z^c.$$

Let $a'_i = a_i / c$. Then we have

$$y_i = g^{a_i} = z^{a_i/c} = z^{a'_i}.$$

Now it is clear that $(a_1, \cdots, a_{n-u}) \cdot \mathbf{b}_i \neq 0$ if and only if $(a'_1, \cdots, a'_{n-u}) \cdot \mathbf{b}_i \neq 0$, where $i = 1, \cdots, n$. Therefore, the public key pk of LC-BF scheme is equivalent to that of LC-KD$'$ scheme.

Header equivalence: Clear.

Decryption equivalence: In LC-BF scheme, from (4.3), we obtain that

$$1/\theta_i = (a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i / c = (a'_1, \cdots, a'_{n-u}) \mathbf{b}_i.$$

On the other hand, in LC-KD$'$ scheme,

$$e_i = (a_1, \ldots, a_{n-u}) \cdot \mathbf{b}_i.$$

Therefore, $1/\theta_i$ of LC-BF scheme is equivalent to e_i of LC-KD$'$ scheme.

Secrecy equivalence: The same public key and the same header are used in both schemes. Therefore, the secrecy of LC-BF scheme against outside enemies is equivalent to that of LC-KD$'$ scheme.

Traceability equivalence: Suppose that there exists a pirate decoder M_0 for LC-BF scheme which is not (black box) traceable. Then we show that there exists a pirate decoder M_1 for LC-KD$'$ scheme which is not (black box) traceable. Let w traitors be $i_1, \cdots, i_w$ in both schemes.

Consider LC-KD$'$ scheme in which a public key is $pk = (g, y_1, \cdots, y_{n-u})$ and the private key of user i is e_i. From the above equivalence, the same pk is used and the private key of user i is $\theta_i = 1/e_i$ in LC-BF scheme.

From our assumption, there exists an algorithm B which creates an untraceable pirate decoder M_0 from pk and $\theta_{i_1}, \cdots, \theta_{i_w}$ for LC-BF scheme.

Now in LC-KD$'$ scheme, our traitors first create M_0 by running B on input pk and $1/e_{i_1}, \cdots, 1/e_{i_w}$. They then use M_0 as their pirate decoder M_1.

Finally it is easy to show that if there is a tracing algorithm which detects some traitor from M_1, then M_0 is also traceable. This contradicts our assumption. Hence, M_1 is not traceable.

The converse part is proved similarly.

Now we have proved the following theorem.

Theorem 4.6.1 *LC-BF scheme is equivalent to LC-KD$'$ scheme.*

4.6.2 BF Scheme = Modified KD Scheme

From Theorem 4.6.1, we have the following equivalence.

Corollary 4.6.1 *BF scheme is equivalent to modified KD scheme.*

However, BF scheme is more complicated than the modified KD scheme because it must use a public code matrix Γ and $2w$ additional secret random numbers $\beta_1, \cdots, \beta_{2w}$. This means that BF scheme is redundant or overdesigned.

4.6.3 Comparison

We compare three schemes, LC-KD scheme, LC-BF scheme and LC-KD$'$ scheme. We have seen that LC-BF scheme is equivalent to LC-KD$'$ scheme, and hence redundant.

Now in LC-KD$'$ scheme, $a_1, \cdots, a_{n-u}$ must be chosen in such a way that $e_i \neq 0$ for $i = 1, \cdots, n$, which LC-KD scheme does not require. This check is very inefficient if n is large. Therefore. LC-KD scheme is better than LC-KD$'$ scheme from a view point of key generation.

Similarly, the corrected KD scheme is better than modified KD scheme from a view point of key generation. Further, modified KD scheme is better than BF scheme as shown in Section 4.6.2. As a conclusion, we see that the corrected KD scheme is the best among them.

4.7 Secrecy and Traceability

In this section, we prove the secrecy and the traceability of LC-KD scheme, LC-KD$'$ scheme and LC-BF scheme.

Note that any $2w$ columns of H are linearly independent because $d \geq 2w + 1$.

4.7.1 Secrecy of LC-KD Scheme

Theorem 4.7.1 *LC-KD scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption.*

(Proof) Similarly to the proof of [39, Theorem 14], we can show that the secrecy of LC-KD scheme is reduced to that of ElGamal encryption scheme. It is well known that ElGamal encryption scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption. □

4.7.2 Black Box Tracing Algorithm for LC-KD Scheme

Let BAD be the set of at most w traitors who created a confiscated pirate decoder. Let A be a subset of at most w users. We first describe a procedure TEST which checks whether $A \cap BAD \neq \emptyset$.

Suppose that $(e_T, e_1, \cdots, e_n)$ is being used as the key. For a random encryption key $e'_T = (a'_1, \cdots, a'_{n-m})$, let the corresponding private decryption keys be $(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H$. We say that e'_T matches with A if $e'_i = e_i$ for all $i \in A$.

TEST(A)

Step 1. T chooses e'_T which matches A randomly. (We can do this because any $2w$ columns of H are linearly independent.) He chooses a random session key s' and computes an *illegal* header

$$h' \overset{R}{\leftarrow} e'_T(s'). \tag{4.7}$$

Step 2. T gives h' to the pirate decoder. Let the output of the pirate decoder be s_A.

Output:

$$TEST(A) = \begin{cases} 1 & \text{if } s_A = s' \\ 0 & \text{otherwise} \end{cases}$$

We next describe a procedure $TEST2(A, m)$ which runs $TEST(A)$ m times independently, where m is a sufficiently large positive integer.

TEST2(A, m)
Set *counter* $:= 0$. For $i = 1, 2, \ldots, m$, do

Step 1. Run $TEST(A)$ randomly.

Step 2. Let *counter* $:=$ *counter* $+ TEST(A)$. Reset the pirate decoder.

Output: $TEST2(A, m)$, the final value of *counter*.

We say that a set of users A is *marked* if $TEST2(A, m) = m$. We now present our tracing algorithm.

Black box tracing algorithm
Find a marked set $A = \{i_1, i_2, \ldots, i_w\}$ by exhaustive search. Suppose that $i_1 < i_2 < \cdots < i_w$. For $j = 1, 2, \ldots, w$, do:

Step 1. Let $B := A \setminus \{i_1, i_2, \ldots, i_j\}$. Run $TEST2(B, m)$.

Step 2. Let $m_j = TEST2(B, m)$.

Output: i_j such that $m_{j-1} - m_j$ is the maximum. (If there are more than one such j, choose one of them arbitrarily.)

User i_j is a traitor.

4.7.3 Validity of Our Tracing Algorithm

We can show the validity of our tracing algorithm by using the following three *test conditions.*

(1) If $A \supseteq BAD$, then $\Pr[TEST(A) = 1]$ is overwhelming.

(2) If $A \cap BAD = \emptyset$, then $\Pr[TEST(A) = 1]$ is negligible.

(3) If $A \cap BAD \neq \emptyset$ and $A \setminus BAD \neq \emptyset$, then for any $i \in A \setminus BAD$,

$$|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{i\}) = 1]|$$

is negligible.

Theorem 4.7.2 *If the above three conditions are satisfied, then our black box tracing algorithm succeeds with overwhelming probability. That is user i_j is a traitor.*

(Proof) If $A \supseteq BAD$, then $TEST2(A, m) = m$ with overwhelming probability from (1). Therefore, there exists at least one marked A. On the other hand, from (2), if $A \cap BAD = \emptyset$, then $TEST2(A, m) \ll m$. This means that if A is marked, then $A \cap BAD \neq \emptyset$.

Now suppose that A is marked. Let $m_0 = m$. It is easy to see that $m_w = 0$. If $m_{j-1} - m_j$ is the maximum, then $m_{j-1} - m_j \geq m/w$. On the other hand, if $j \in A \setminus BAD$, then $m_{j-1} - m_j \ll m/w$ from (3). Therefore, if $m_{j-1} - m_j$ is the maximum, then $i_j \in BAD$. □

We finally show that LC-KD scheme satisfies the above three test conditions under the DDH assumption. We assume that a pirate decoder decrypts valid headers with overwhelming probability.

Theorem 4.7.3 (Test Condition (1)) *In LC-KD scheme, if $A \supseteq BAD$, then $\Pr[TEST(A) = 1]$ is overwhelming under the DDH assumption.*

Theorem 4.7.4 (Test Condition (2)) *In LC-KD scheme, if $A \cap BAD = \emptyset$, then $\Pr[TEST(A) = 1]$ is negligible under the DDH assumption.*

Theorem 4.7.5 (Test Condition (3)) *In LC-KD scheme, if $A \cap BAD \neq \emptyset$ and $A \setminus BAD \neq \emptyset$, then for any $i \in A \setminus BAD$,*

$$|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{i\}) = 1]|$$

is negligible under the DDH assumption.

The proofs will be given in the following sections.

4.7.4 Secrecy and Traceability of LC-KD′ Scheme

The secrecy and the traceability of LC-KD′ Scheme are proved similarly.

Theorem 4.7.6 *LC-KD′ scheme is indistinguishably secure against chosen plaintext attack under the DDH assumption.*

Theorem 4.7.7 (Test Condition (1)) *In LC-KD′ scheme, if $A \supseteq BAD$, then* $\Pr[TEST(A) = 1]$ *is overwhelming under the DDH assumption.*

Theorem 4.7.8 (Test Condition (2)) *In LC-KD′ scheme, if $A \cap BAD = \emptyset$, then* $\Pr[TEST(A) = 1]$ *is negligible under the DDH assumption.*

Theorem 4.7.9 (Test Condition (3)) *In LC-KD′ scheme, if $A \cap BAD \neq \emptyset$ and $A \setminus BAD \neq \emptyset$, then for any $i \in A \setminus BAD$,*

$$|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{i\}) = 1]|$$

is negligible under the DDH assumption.

We show the proof of Theorem 4.7.8 in the following section. The other theorems are proved similarly to those of LC-KD scheme.

4.7.5 Secrecy and Traceability of LC-BF Scheme

The secrecy and the traceability of LC-BF Scheme are equivalent to those of LC-KD′ scheme as shown in Section 4.6.1.

4.7.6 Proof of Theorem 4.7.3

By extending the result of Stadler [55, in the proof of Proposition 1] and Naor and Reingold [49, lemma 3.2], Bellare et al. proved the following proposition [5].

Proposition 4.7.1 *[5] There is a probabilistic algorithm Σ such that on input g^a, g^b, g^c, Σ outputs $g^{b'}, g^{c'}$, where b' is random and*

$$c' = \begin{cases} ab' \bmod p & \text{if } c = ab \bmod p \\ random & \text{if } c \neq ab \bmod p \end{cases}$$

Σ runs in $O(T^{exp})$ time, where T^{exp} is the time needed to perform an exponentiation.

Now we show that

$$p_0 = |\Pr[P \text{ decrypts valid headers correctly}] - \Pr[TEST(A) = 1]|$$

is negligible for any pirate decoder P.

Suppose that $p_0 \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder such that $p_0 \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2w})$ and the private keys of BAD.

Now on input $d = (g, g'y, y')$, M works as follows.

1. Choose e_i at random for each $i \in A$ and let $e'_i = e_i$ for each $i \in A$.

2. Let $OUT = \{i_1, i_2, \ldots, i_w\}$ be a w-subset of users such that $OUT \cap A = \emptyset$.

3. For $j = 1, 2, \ldots, w$, M runs Σ of Proposition 4.7.1 w times independently on input $d = (g, g', y, y')$. Let the output of Σ be $g^{e_{i_j}}, (g')^{e'_{i_j}}$.

4. Compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$ from $\{g^{e_i} \mid i \in OUT \cup A\}$, where
$$(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H.$$
Each a_i is written as a linear combination of $\{e_i \mid i \in A \cup BAD\}$ because any $2w$ columns of H are linearly independent and $|A \cup BAD| \leq 2w$. Therefore, we can do this.

5. Compute $(g')^{a'_1}, (g')^{a'_2}, \ldots, (g')^{a'_{n-u}}$ from $\{(g')^{e'_i} \mid i \in OUT \cup A\}$, where
$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$

6. Select a random session key s' and compute h' as follows.
$$h' = (g', s'(g')^{a'_1}, (g')^{a'_2}, \ldots, (g')^{a'_{n-u}}).$$

7. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.

8. Give h' to the pirate decoder P. Let the output of P be s_A.

9. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

For $OUT = \{i_1, i_2, \ldots, i_w\}$, it holds that

$$e'_{i_j} = \begin{cases} e_{i_j} \bmod p & \text{if } d \leftarrow \mathbf{D} \\ \text{random} & \text{if } d \leftarrow \mathbf{R}. \end{cases}$$

from Proposition 4.7.1. Therefore, if d is chosen from $\mathbf{D}$, h' is a legal header. On the other hand, if d is chosen from $\mathbf{R}$, h' is an illegal header used in $TEST(A)$. Hence, we have

$$\begin{aligned} &|\Pr[M(d) = 1 \mid d \in \mathbf{D}] - \Pr[M_w(d) = 1 \mid d \in \mathbf{R}]| \\ &= p_0 \\ &\geq \epsilon. \end{aligned}$$

from our assumption.

This means that M can distinguish $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

4.7.7 Proof of Theorem 4.7.4

Suppose that $\Pr[TEST(A) = 1] \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder P such that $\Pr[TEST(A) = 1] \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2w})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose $a'_2, \ldots, a'_{n-u}$ at random. Let a'_1 be such that $g^{a'_1} = y$.

2. Select a random session key s' and compute h' as follows.

$$h' = (g', s'y', y^{a'_2}, y^{a'_3}, \ldots, y^{a'_{n-u}}).$$

3. Compute $g^{e'_1}, g^{e'_2}, \ldots, g^{e'_n}$ from $g^{a'_1}, g^{a'_2}, \ldots, g^{a'_{n-u}}$, where

$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$

4. Choose e_i at random for each $i \in BAD$. Let $g^{e_i} = g^{e'_i}$ for each $i \in A$.

5. From $\{g^{e_i} \mid i \in A \cup BAD\}$, compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$, where

$$(e_1, \ldots, e_n) = (a_1, \cdots, a_{n-u}) \cdot H$$

   Each a_i is written as a linear combination of $\{e_i \mid i \in A \cup BAD\}$ because any $2w$ columns of H are linearly independent and $|A \cup BAD| \leq 2w$. Therefore, we can do this.

6. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.

   Give h' to the pirate decoder P. Let the output of P be s_A.

7. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

Then we obtain that

$$\begin{aligned} &|\Pr[M(d) = 1 \mid d \leftarrow \mathbf{D}] - \Pr[M(d) = 1 \mid d \leftarrow \mathbf{R}]| \\ &= \ |\Pr[s_A = s' \mid d \leftarrow \mathbf{D}] - \Pr[s_A = s' \mid d \leftarrow \mathbf{R}]| \end{aligned}$$

First we see that $\Pr[s' = s \mid d \leftarrow \mathbf{R}]$ is negligible because y' is random. Next it is easy to see that if d is chosen from $\mathbf{D}$, then h' is a testing header used in $TEST(A)$. Therefore,

$$\Pr[s_A = s' \mid d \leftarrow \mathbf{D}] = Pr[TEST(A) = 1] \geq \epsilon$$

from our assumption.

This means that M can distinguishes $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

4.7.8 Proof of Theorem 4.7.5

Suppose that

$$|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{\tilde{i}\}) = 1]| \geq \epsilon$$

for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^r, y, y^r)$ and $\mathbf{R} = (g, g^r, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and r is chosen at random from Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder P such that

$$|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{i\}) = 1]| \geq \epsilon$$

from a public key $pk = (g, y_1, \dots, y_{n-u})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose e_i for each $i \in BAD \cup (A \setminus \{\tilde{i}\})$. Let $e_{\tilde{i}}$ be such that $g^{e_{\tilde{i}}} = y$.

2. Compute $g^{a_1}, g^{a_2}, \dots, g^{a_{n-u}}$ from $\{g^{e_i} \mid i \in BAD \cup A\}$, where

$$(e_1, \cdots, e_n) = (a_1, \cdots, a_{n-u})H.$$

3. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \dots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.

4. Next let $e'_i = e_i$ for each $i \in A \setminus \{\tilde{i}\}$ and $e'_{\tilde{i}}$ be such that $y^{e'_{\tilde{i}}} = y'$.

5. Compute $y^{a'_1}, y^{a'_2}, \dots, y^{a'_{n-u}}$ from $\{y^{e'_i} \mid i \in BAD \cup A\}$, where

$$(e'_1, \cdots, e'_n) = (a'_1, \cdots, a'_{n-u})H.$$

6. Select a random session key s' and compute h' as follows.

$$h' = (g', s'y^{a'_1}, y^{a'_2}, \dots, y^{a'_{n-u}}).$$

   Give h' to the pirate decoder P. Let the output of P be s_A.

7. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

It is easy to see that if d is chosen from $\mathbf{D}$, then h' is an illegal header used in $TEST(A)$. On the other hand, if d is chosen from $\mathbf{R}$, then h' is an illegal header used in $TEST(A \setminus \{\tilde{i}\})$.

Therefore,

$$
\begin{aligned}
&|\Pr[M(d) = 1 \mid d \leftarrow \mathbf{D}] - \Pr[M(d) = 1 \mid d \leftarrow \mathbf{R}]| \\
&= \left|\Pr[TEST(A) = 1] - \Pr[TEST(A \setminus \{\tilde{i}\}) = 1]\right| \\
&\geq \epsilon
\end{aligned}
$$

from our assumption.

This means that M can distinguish $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

4.7.9 Proof of Theorem 4.7.8

Suppose that $\Pr[TEST(A) = 1] \geq \epsilon$ for some nonnegligible probability ϵ. Then we show that there exists a probabilistic polynomial time Turing machine M which can distinguish $\mathbf{D} = (g, g^a, y, y^a)$ and $\mathbf{R} = (g, g^a, y, v)$ with nonnegligible probability, where g, y, v are chosen at random from G_q and a is chosen at random in Z_q.

From our assumption, there is an algorithm B which creates a pirate decoder such that $\Pr[TEST(A) = 1] \geq \epsilon$ from a public key $pk = (g, y_1, \cdots, y_{2w})$ and the private keys of BAD.

Now on input $d = (g, g', y, y')$, M works as follows.

1. Choose e_i at random for each $i \in BAD$.

2. For each $i \in A$, choose t_i at random and compute y^{t_i}. Define e_i as $g^{e_i} = y^{t_i}$.

3. From $\{g^{e_i} \mid i \in A \cup BAD\}$, compute $g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}}$, where
$$(e_1, \ldots, e_n) = (a_1, \cdots, a_{n-u}) \cdot H.$$

4. Create a pirate decoder P by running B on input a public key $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{n-u}})$ and the private keys of BAD, $\{e_i \mid i \in BAD\}$.

5. For each $i \in A$, compute $\beta_i = (y')^{t_i}$. For each $i \in BAD$, choose a random element β_i.

6. Suppose that $y' = y^r$. Define e'_i as $\beta_i = g^{re'_i}$ for each $i \in (A \cup BAD)$.

7. From $\{\beta_i \mid i \in A \cup BAD\}$, compute $g^{ra'_1}, g^{ra'_2}, \ldots, g^{ra'_{n-u}}$, where $\beta_i = g^{re'_i}$ and

$$(e'_1, \ldots, e'_n) = (a'_1, \cdots, a'_{n-u}) \cdot H.$$

8. Select a random session key s' and compute h' as follows.

$$h' = (s'g', g^{ra'_1}, g^{ra'_2}, \ldots, g^{ra'_{n-u}}).$$

   Give h' to the pirate decoder P. Let the output of P be s_A.

9. Finally M outputs 1 if $s_A = s'$ or 0 otherwise.

Then we obtain that

$$\begin{aligned} &|\Pr[M(d) = 1 \mid d \leftarrow \mathbf{D}] - \Pr[M(d) = 1 \mid d \leftarrow \mathbf{R}]| \\ = \;& |\Pr[s_A = s' \mid d \leftarrow \mathbf{D}] - \Pr[s_A = s' \mid d \leftarrow \mathbf{R}]| \end{aligned}$$

First we see that $\Pr[s' = s \mid d \leftarrow \mathbf{R}]$ is negligible because g' is random.

Next we will show that if d is chosen from $\mathbf{D}$, then h' is an illegal header used in $TEST(A)$. In this case, $y' = y^r$ and $g' = g^r$ for some r. We need to show that $e'_i = e_i$ for each $i \in A$. Assume that $y = g^x$. Then

1. $e_i = xt_i$ since $y^{t_i} = g^{e_i}$.

2. On the other hand,

$$g^{re'_i} = \beta_i = (y')^{t_i} = y^{rt_i} = g^{xrt_i}.$$

   Therefore, $e'_i = xt_i$.

Hence, $e'_i = e_i$. Therefore, h' is an illegal header used in $TEST(A)$. Consequently,

$$\Pr[s_A = s' \mid d \leftarrow \mathbf{D}] = Pr[TEST(A) = 1] \geq \epsilon$$

from our assumption.

This means that M can distinguishes $\mathbf{D}$ and $\mathbf{R}$ with nonnegligible probability.

4.8 Black Box Full Traceability of Proposed Tracing Algorithm

In this section, we generalize the proposed tracing algorithm of Section 4.7.2 and discuss its traceability and error probabilities in detail.

We first introduce a few notations.

Definition 4.8.1 *$C(m, A)$ is a set of ciphertexts given which the decryption algorithm outputs m if the decryption key of a member of A is used as a key while no algorithm can obtain any information on m without any of the decryption keys of A.*

p_A for a pirate decoder is the probability that given a ciphertext $c \in C(m, A)$ the decoder outputs m. The probability is taken over the choices of m, c and the random tape of the pirate decoder.

Before describing the generalized tracing algorithm, we will restate the test conditions using the above notations.

Test Condition (1)

If $A \supseteq BAD$, then $|p_U - p_A|$ is negligible where U is the set of all users. Note that p_U is not necessarily overwhelming.

Test Condition (2)

If $A \cap BAD = \emptyset$, then p_A is negligible.

Test Condition (3)

If $A \cap BAD \neq \emptyset$ and $A \backslash BAD \neq \emptyset$, then for any $i \in A \backslash BAD$, $|p_A - p_{A\backslash\{i\}}|$ is negligible.

Then the tracing algorithm of Section 4.7.2 is generalized as follows.

Black Box Tracing Algorithm

Find all i such that $|p_A - p_{A\backslash\{i\}}|$ is non-negligible for some set of users A.

We discuss the traceability and error probabilities of this algorithm in the following sections.

4.8.1 Black Box Full Traceability

In this section, we prove the proposed black box tracing algorithm can trace *all* the traitors whose keys are *actively* used to construct a pirate decoder if it is used in any of the six public key (w, n)-traceability schemes discussed in previous sections. Note that it also works in the trivial scheme.

Even though a traitor provides his decryption key for constructing a pirate decoder, it is difficult to trace the traitor if his key is not effectively used in the decoder. Therefore, we consider *an active set of traitors.* For a pirate decoder, the active set of traitors is defined as a minimal set of traitors who can construct the same pirate decoder. We aim to trace all members of the active set.

It is obvious that tracing all traitors is impossible if there exist two or more active sets which can construct the confiscated pirate decoder. Hence, we first need to prove the following theorem.

Theorem 4.8.1 *The active set is uniquely determined from a confiscated pirate decoder.*

(Proof) Suppose there exist two active sets of traitors BAD_1 and BAD_2. We demonstrate $BAD_1 \cap BAD_2$ can construct the same pirate decoder as the confiscated pirate decoder and therefore contradicts the minimality of the active sets.

$BAD_1 \cap BAD_2$ can construct the following pirate decoder.

1. Given an input $c \in C(m, A)$, determine $A \cap BAD_1 \cap BAD_2$ as follows.

   Decrypt the input using every keys of $BAD_1 \cap BAD_2$ and let $A \cap BAD_1 \cap BAD_2$ be the set of users from whose keys the same message is obtained.

   It is easy to see this step works correctly with high probability. Moreover, the probability can be increased if MAC or signature of m is attached to the ciphertext.

2. Output the message with probability $p_{A \cap BAD_1 \cap BAD_2}$.

We then prove that the above decoder correctly simulates the confiscated pirate decoder by showing $p_A = p_{A \cap BAD_1 \cap BAD_2}$ for any A. Since the test conditions hold, from view point of BAD_1, $p_A = p_{A \cap BAD_1}$. On the other hand, from view point of BAD_2, $p_{A \cap BAD_1} = p_{(A \cap BAD_1) \cap BAD_2}$. □

We now state the main result on black box full traceability.

Theorem 4.8.2 *The proposed tracing algorithm can trace all members of active set.*

(Proof) We first prove that users detected by the tracing algorithm are members of the active set. Let BAD be the active set. Suppose the tracing algorithm detects user $i \notin BAD$ as a traitor. This implies there exists A such that $|p_A - p_{A\setminus\{i\}}|$. This contradicts Test Condition (3).

We then prove that users who were not detected by the algorithm are not members of the active set. Let BAD be the active set. Suppose the tracing algorithm does not detect user $i \in BAD$ as a traitor. This means $|p_A - p_{A\setminus\{i\}}|$ is negligible for any A. Hence, $BAD\setminus\{i\}$ can construct the same pirate decoder in a similar way to the construction in the proof of Theorem 4.8.1. □

4.8.2 Error Probabilities

In this section, we deal with two error probabilities of the proposed tracing algorithm. They include the probability that the algorithm outputs innocent users as traitors and the probability that the algorithm cannot trace members of the active set.

If the tracer has unlimited computational power, then the value of p_A for any A can be explicitly computed and therefore both probabilities are zero. However, there usually is a limit on computational resources and the tracing algorithm can get only approximate values of p_A.

We evaluate the error probabilities when the number of ciphertexts that the algorithm is allowed to give to the confiscated pirate decoder is t for each A. Let the approximate value $\hat{p}_A$ of p_A be the number that the pirate decoder outputs m given a ciphertext $c \in C(m, A)$ divided by t. If the tracing algorithm considers $|p_A - p_{A\setminus\{i\}}|$ is negligible if and only if $|\hat{p}_A - \hat{p}_{A\setminus\{i\}}| < \alpha$ for some probability α. Then the error probabilities are bounded by the following theorems.

Theorem 4.8.3 *For any A, i and any probabilities α and ϵ,*

$$\Pr[|\hat{p}_A - \hat{p}_{A\setminus\{i\}}| > \alpha \ : \ |p_A - p_{A\setminus\{i\}}| < \epsilon] \leq \frac{2}{(\alpha - \epsilon)^2 t}.$$

Theorem 4.8.4 *For any A, i and any probabilities α and $\delta > \alpha$,*

$$\Pr[|\hat{p}_A - \hat{p}_{A\setminus\{i\}}| < \alpha \ : \ |p_A - p_{A\setminus\{i\}}| > \delta] \leq \frac{2}{(\delta - \alpha)^2 t}.$$

We first derive a lemma to prove the above thereoms.

Proposition 4.8.1 (Chebyshev's inequality) *Let X be a random variable with mean $\mu = E(X)$ and variance $\sigma^2 = Var(X)$. Then for any $\lambda > 0$,*

$$\Pr[|X - \mu| > \lambda\sigma] \leq \frac{1}{\lambda^2}.$$

Lemma 4.8.1 *For any A and $\lambda > 0$,*

$$\Pr[|\hat{p}_A - p_A| > \frac{\lambda\sqrt{p_A(1-p_A)}}{\sqrt{t}}] \leq \frac{1}{\lambda^2}.$$

Moreover,

$$\Pr[|\hat{p}_A - p_A| > \frac{\lambda}{2\sqrt{t}}] \leq \frac{1}{\lambda^2}.$$

(Proof) In binomial distribution, $E(X) = t\theta$ and $\sigma^2(X) = t\theta(1-\theta)$. Apply proposition 4.8.1. Then since $p_A(1-p_A) \leq \frac{1}{4}$ for any probability p_A, the second inequality is obtained. □

(Proof) [Theorem 4.8.3]

$$\begin{aligned}
& \Pr[|\hat{p}_A - \hat{p}_{A\setminus\{i\}}| > \alpha \ : \ |p_A - p_{A\setminus\{i\}}| < \epsilon] \\
\leq \ & \Pr[|\hat{p}_A - p_A| \geq \frac{\alpha-\epsilon}{2} \vee |\hat{p}_{A\setminus\{i\}} - p_{A\setminus\{i\}}| \geq \frac{\alpha-\epsilon}{2} \ : \ |p_A - p_{A\setminus\{i\}}| < \epsilon] \\
\leq \ & \Pr[|\hat{p}_A - p_A| \geq \frac{\alpha-\epsilon}{2} \ : \ |p_A - p_{A\setminus\{i\}}| < \epsilon] \\
& + \Pr[|\hat{p}_{A\setminus\{i\}} - p_{A\setminus\{i\}}| \geq \frac{\alpha-\epsilon}{2} \ : \ |p_A - p_{A\setminus\{i\}}| < \epsilon] \\
\leq \ & 2 \times \frac{1}{4\left(\frac{\alpha-\epsilon}{2}\right)^2 t} \quad \text{(From Lemma 4.8.1)} \\
= \ & \frac{2}{(\alpha-\epsilon)^2 t}
\end{aligned}$$

□

(Proof) [Theorem 4.8.4]

$$\begin{aligned}
& \Pr[|\hat{p}_A - \hat{p}_{A\setminus\{i\}}| < \alpha \ : \ |p_A - p_{A\setminus\{i\}}| > \delta] \\
\leq \ & \Pr[|\hat{p}_A - p_A| > \frac{\delta-\alpha}{2} \vee |\hat{p}_{A\setminus\{i\}} - p_{A\setminus\{i\}}| > \frac{\delta-\alpha}{2} \ : \ |p_A - p_{A\setminus\{i\}}| > \delta] \\
\leq \ & \Pr[|\hat{p}_A - p_A| > \frac{\delta-\alpha}{2} \ : \ |p_A - p_{A\setminus\{i\}}| > \delta]
\end{aligned}$$

$$
\begin{aligned}
& + \Pr[|\hat{p}_{A\setminus\{i\}} - p_{A\setminus\{i\}}| > \frac{\delta - \alpha}{2} \ : \ |p_A - p_{A\setminus\{i\}}| > \delta] \\
\leq & \ 2 \times \frac{1}{4\left(\frac{\delta-\alpha}{2}\right)^2 t} \quad \text{(From Lemma 4.8.1)} \\
= & \ \frac{2}{(\delta - \alpha)^2 t}
\end{aligned}
$$

□

Chapter 5

How to Break Some Revocation and Tracing Schemes

5.1 Introduction

In such applications as pay TV, CD-ROM distribution and online databases, data should only be available to authorized users. To prevent unauthorized users from accessing data, the data supplier will encrypt data and provide only the authorized users with personal keys to decrypt it. However, some authorized users (*traitors*) may create a pirate decoder.

A (w, n)-traceability scheme is a scheme in which at least one traitor is detected from a confiscated pirate decoder if there are at most w traitors among n authorized users. Chor, Fiat and Naor [19] introduced the first (w, n)-traceability scheme. The data supplier also wants to exclude some subset of users time to time. Such a broadcast encryption scheme is called a revocation scheme.

Let $\mathcal{U}$ be the set of users such that $|\mathcal{U}| = n$ and let $\mathcal{R} \subseteq \mathcal{U}$ be a group of $|\mathcal{R}| = r$ users whose decryption privileges should be revoked. The goal of a revocation scheme is to allow a center to transmit a message M to all users such that any user $u \in \mathcal{U} \setminus \mathcal{R}$ can decrypt the message correctly, while even a coalition consisting of all members of $\mathcal{R}$ cannot decrypt it.

In this chapter, we show attacks and comments on some of revocation and tracing schemes. They include Chor, Fiat and Naor(CFN) traceability scheme[19, 20], Naor, Naor and Lotspiech(NNL) revocation schemes with traceability[48], Matsuzaki, Anzai and Matsumoto(MAM) revocation scheme[47],

M. Yoshida and Fujiwara(YF) revocation scheme with traceability[65] and Tzeng and Tzeng(TT) revocation scheme with traceability[61].

For example, Naor, Naor and Lotspiech showed two revocation schemes and a traitor tracing algorithm for them at Crypto 2001[48]. However, we show that NNL revocation schemes cannot be traceable.

5.2 Attack on NNL Schemes

At Crypto 2001, Naor, Naor and Lotspiech proposed two revocation schemes which can revoke any subset of users [48]. The authors call the two schemes the complete subtree scheme and the subset difference scheme. They also provided a traitor tracing algorithm for each revocation scheme.

In this section, however, we show that none of these schemes has traceability. Our attack succeeds not only in the sense of black box traceability, but also in the sense of weak traceability, i.e. there exists a strategy of traitors to create a pirate key from which no traitor is correctly detected.

Their complete subtree revocation scheme is described as follows. Let E_L denote a symmetric cryptosystem keyed with L. For simplicity, assume that n is a power of 2. Imagine the users as the leaves in a rooted full binary tree $\mathcal{T}$ with n leaves.

Key Generation: Assign an independently and random long-lived key L_i to every node v_i of the complete tree $\mathcal{T}$. The transmission key e_T is the set of all L_i.

The personal decryption key of user u is the set of $\log_2 n + 1$ keys assigned to the nodes along the path from the root to leaf u, i.e.

$$d_u = \{L_i \mid v_i \text{ is a node on the path from the root to the leaf } u\}.$$

Encryption: For a given set $\mathcal{R}$ of revoked users, let $S_{i_1}, \ldots, S_{i_m}$ be the maximal subtrees of $\mathcal{T}$ which hang off $\mathcal{R}$. That is, the set of the leaves of $S_{i_1}, \ldots, S_{i_m}$ is equal to $\mathcal{U} \setminus \mathcal{R}$. We use i_j to denote the identity of the root of subtree S_{i_j}.

The center then chooses a session key s and broadcast a header

$$H = (i_1, \ldots, i_m, E_{L_{i_1}}(s), \ldots, E_{L_{i_m}}(s)).$$

Decryption: For each user $u \notin \mathcal{R}$, there exists exactly one ancestor $i \in \{i_1, \ldots, i_m\}$. Hence, user u can compute s from H by using d_u.

Tracing algorithm: See [48].

Example

Suppose that the set of user is $\mathcal{U} = \{1,2,3,4,5,6,7,8\}$ and the center uses the complete tree described below.

Then $e_T = \{L_1, L_2, \ldots, L_{15}\}$ and the personal decryption keys are $d_1 = \{L_1, L_9, L_{13}, L_{15}\}$, $d_2 = \{L_2, L_9, L_{13}, L_{15}\}$, $d_3 = \{L_3, L_{10}, L_{13}, L_{15}\}$ and so on.

When the center wants to exclude $\mathcal{R} = \{5,6\}$, the header is

$$h = (12, 13, E_{L_{12}}(s), E_{L_{13}}(s)).$$

Given the header, each of the valid users can compute s because he has either L_{12} or L_{13}. On the contrary, user 5 and 6 cannot obtain any information on s.

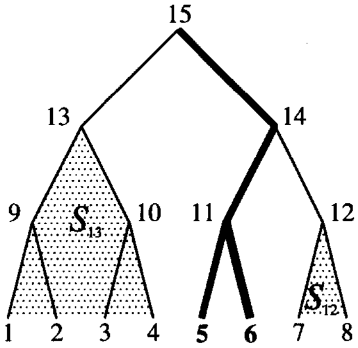

Fig.1. The complete subtree scheme.

We now show that the complete subtree scheme cannot be traceable although the authors showed a traitor tracing algorithm for it. Therefore, their tracing algorithm is broken.

Theorem 5.2.1 *The complete subtree scheme has no traceability.*

(Proof) Suppose user u is a traitor. He constructs a pirate key d_p such that

$$d_p = d_u \setminus \{L_u\}$$

where L_u is the key assigned to leaf u. Let u' be a sibling leaf of u. (For example, $u = 1$ and $u' = 2$ in Fig.1.) Then d_p can decrypt headers if $u \notin \mathcal{R}$ and $u' \notin \mathcal{R}$.

However, given the pirate key d_p, no tracing algorithm can determine if user u is a traitor or user u' is a traitor. □

Theorem 5.2.2 *The subset difference method has no traceability.*

(Proof) The proof is given similarly. □

5.3 CFN Scheme Cannot Have Full Traceability

Chor, Fiat and Naor showed the first traceability scheme. In their scheme, at least one traitor is detected from a confiscated pirate decoder. In this section, we show that their scheme cannot have full traceability, i.e., it is impossible to trace all traitors.

CFN scheme [19, 20] is described as follows.

Key Generation Algorithm: A $q \times N$ sub-key matrix A is chosen at random.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,q} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,q} \end{pmatrix}$$

A set of N hash functions $h_1, h_2, \ldots, h_N$ is chosen. Each hash function h_i maps $\{1, 2, \ldots, n\}$ into $\{1, 2, \ldots, q\}$. The encryption key e is the entire sub-key matrix A and the personal decryption key of user u is the sub-key vector

$$d_u = (a_{1,h_1(u)}, a_{2,h_2(u)}, \ldots, a_{N,h_N(u)}).$$

Encryption Algorithm: Given a session key s, split it into N shares $s_1, s_2, \ldots, s_N$ with an (N, N)-threshold secret sharing scheme. For each $j = 1, 2, \ldots, N$, s_j is encrypted under a secure symmetric encryption scheme E using each sub-key of jth row of A as a key. Then the center broadcasts the header

$$H = \begin{pmatrix} E_{a_{1,1}}(s_1) & E_{a_{1,2}}(s_1) & \cdots & E_{a_{1,q}}(s_1) \\ E_{a_{2,1}}(s_2) & E_{a_{2,2}}(s_2) & \cdots & E_{a_{2,q}}(s_2) \\ \vdots & \vdots & \ddots & \vdots \\ E_{a_{N,1}}(s_N) & E_{a_{N,2}}(s_N) & \cdots & E_{a_{N,q}}(s_N) \end{pmatrix}.$$

Decryption Algorithm: Each authorized user has one sub-key from every row of A, and can decrypt $s_1, s_2, \ldots, s_N$, and thus compute s.

Tracing Algorithm: See [19, 20].

Theorem 5.3.1 *CFN Scheme cannot have full traceability.*

(Proof) Suppose user u and user u' are traitors and construct a pirate decoder d_p as follows.

Let the personal decryption keys of u and u' are

$$d_u = (a_{1,h_1(u)}, a_{2,h_2(u)}, \ldots, a_{N,h_N(u)})$$

and

$$d_{u'} = (a_{1,h_1(u')}, a_{2,h_2(u')}, \ldots, a_{N,h_N(u')})$$

respectively. Let j be the position such that $a_{j,h_j(u)} \neq a_{j,h_j(u')}$. Note that such j always exists, otherwise the personal decryption keys of u and u' are identical and tracing traitors is impossible.

The pirate key d_p is

$$d_p = (a_{1,h_1(u)}, a_{2,h_2(u)}, \ldots, a_{j-1,h_{j-1}(u)}, a_{j,h_j(u')}, a_{j+1,h_{j+1}(u)}, \ldots, a_{N,h_N(u)}).$$

As it contains one sub-key from each row of the key matrix, decryption can be performed correctly using the pirate key d_p User u alone cannot construct d_p and neither can user u'. Thus $\{u, u'\}$ is an active set with respect to the pirate decoder d_p. Since user u' contributes in constructing d_p only by giving a sub-key $a_{j,h_j(u')}$, u' cannot be traced from other information but $a_{j,h_j(u')}$. However, in general , there exist users who have the same sub-key of A as $a_{j,h_j(u')}$.

Therefore, no tracing algorithm can accuse u' as a traitor. □

5.4 Attack on MAM Scheme

Anzai, Matsuzaki and Matsumoto[2] and Naor and Pinkas[51] independently proposed a revocation scheme (AMM-NP scheme) which can revoke up to w out of n users. This scheme is more efficient than any of NNL schemes because the size of revocable users is limited to up to w while NNL schemes can revoke any subset of users.

Matsuzaki, Anzai and Matsumoto [47] recently showed an improved scheme (MAM scheme) of their original AMM-NP revocation scheme [2, 51].

In this section, however, we show some attacks on MAM scheme.

5.4.1 AMM-NP Scheme [2, 51]

AMM-NP revocation scheme [2, 51] is described as follows.

Let $q > n$ be a prime. Let G_q be a group of prime order q. Let $g \in G_q$ be a generator of G_q. Let s be a session key to be broadcast.

Key Generation Algorithm: The center chooses a random polynomial $f(x) = \sum_{j=0}^{w} a_j x^j$ over $GF(q)$. The transmission key is $e_T = f(x)$ and the personal decryption key of user u is $d_u = (u, f(u))$.

Encryption Algorithm: Suppose that $t \leq w$ users should be excluded, say $\mathcal{R} = \{j_1, \cdots, j_t\}$. Choose another $w - t$ random points $(j_{t+1}, f(j_{t+1})), , \ldots, (j_w, f(j_w))$ which are not assigned to any user.

The center then chooses a random number r and broadcasts a header

$$H = \langle sg^{ra_0}, g^r, (j_1, g^{rf(j_1)}), (j_2, g^{rf(j_2)}), \ldots, (j_w, g^{rf(j_w)})\rangle.$$

Decryption Algorithm: Only an unrevoked authorized user can compute g^{ra_0} by performing Lagrange interpolation formula for $f(x)$ implicitly in the exponent of g^r. Then he can decrypt the session key s.

5.4.2 MAM Scheme [47]

In practical applications such as pay-TV, a set of users whom the center wants to exclude may change time after time. For example, the center wants to exclude users who did not pay the fee for the contents for a certain period. Hence the center needs to change the excluded set and broadcast session keys for each round.

In the original AMM-NP revocation scheme [2, 51], the center can arbitrarily select up to w excluded users independently of the round.

On the contrary, in MAM scheme, once a user is excluded at a certain round, he is to be excluded after the round during the lifetime of the system and his personal decryption key cannot be reused. If the center allows excluded users to compute a new session key, the center needs to assign a pair of new ID and corresponding personal decryption key to each of such users [47, section 7].

Nevertheless, MAM scheme is computationally much lighter than the original scheme. Further, the center can *add* any $w - 1$ users to the set of excluded users *each round.* As a result, the center can eventually exclude all users while in AMM-NP scheme, the center can exclude up to w users in each round.

In order to clarify a round number, we will specify the number by putting it to superscript. For example, $s^{(t)}$ is the session key of round t, $\mathcal{R}^{(t-1)}$ is a set of revoked users in round $t - 1$ and so on.

MAM revocation scheme is described as follows.

Key Generation: Let p and q be large prime powers where $q|p - 1$ and g be a primitive qth root of unity in Z_p. These parameters are public knowledge.

The center chooses two uniformly random polynomials $F(x) = \sum_{i=0}^{w+1} a_i x^i$ and $G(x) = \sum_{i=0}^{w+1} b_i x^i$ over $GF(q)$.

The transmission key is $e_T = (F(x), G(x))$. The personal decryption key of user u is $d_u = (F(u), g^{G(u)/F(u)})$.

Encryption: Suppose the center wants to exclude a set of users $\Lambda^{(t)}$ from round t, where each member of $\Lambda^{(t)}$ was not excluded in the last round $t - 1$ and $|\Lambda^{(t)}| < w$. The center computes and broadcasts a header $H^{(t)}$ as follows.

The center chooses a random number $r^{(t)} \in Z_q$ and $\Theta^{(t)} \subseteq \{n + w \times (t-1) + 1, \ldots, n + w \times t\}$ so that $|\Lambda^{(t)}| + |\Theta^{(t)}| = w$.

The header of the round is

$$H^{(t)} = E_{s^{(t-1)}}(g^{r^{(t)}} \bmod p, \{(j, M_j^{(t)}) \mid j \in \Lambda^{(t)} \cup \Theta^{(t)}\})$$

where

$$M_j^{(t)} = r^{(t)} \times F(j) + G(j) \bmod q$$

and $E_s(\cdot)$ is an encryption with a secure symmetric encryption scheme E under key s.

The session key of the round is $s^{(t)} = g^{r^{(t)} \times a_0 + b_0}$ and the set of all excluded users $\mathcal{R}^{(t)}$ in round t is

$$\mathcal{R}^{(t)} = \bigcup_{i \leq t} \Lambda^{(i)}.$$

$\Lambda^{(t)}$ corresponds to the set of users who were valid in round $t-1$ and are excluded in round t and thereafter.

Decryption: Each valid users $u \notin \mathcal{R}^{(t)}$ can compute the session key $s^{(t)}$ from the session key of the last round $s^{(t-1)}$, his secret decryption key d_u, and the header $H^{(t)}$ of round t as follows.

First, u decrypts the header $H^{(t)}$ with the key $s^{(t-1)}$ and obtains $g^{r^{(t)}}$ and $\{(j, M_j^{(t)}) \mid j \in \Lambda^{(t)} \cup \Theta^{(t)}\}$. He then compute

$$s^{(t)} = (g^{r^{(t)}} \times g^{G(u)/F(u)})^{W_1} \times g^{W_2} \bmod p$$

where

$$\begin{aligned} W_1 &= F(u) \times L(u) \bmod q, \\ W_2 &= \sum_{j \in \Lambda^{(t)} \cup \Theta^{(t)}} (M_j^{(t)} \times L(j)) \bmod q \end{aligned}$$

and L is a Lagrange's interpolation polynomial

$$L(j) = \prod_{t \in \Lambda^{(t)} \cup \Theta^{(t)} \cup \{u\} \setminus \{j\}} t/(t-j).$$

5.4.3 Attacks on MAM scheme

MAM scheme is designed so that only valid users can compute the session keys. Hence, an attack is considered to succeed if someone who is not a valid user can get the session keys. At first, we will show in this section that there is a serious flaw in MAM scheme. Then, we will present four attacks on the scheme. The first two attacks are considered in [47] and the authors claim that the scheme is secure against these attacks. However, we will show that it is incorrect. The later two attacks are not considered in [47] but they are very likely to happen in many practical applications.

The flaw

We show a flaw which makes the system vulnerable not only in a few rounds but during the lifetime of the system.

Theorem 5.4.1 *Suppose that user u is excluded in round t, i.e. $u \in \mathcal{R}^{(t)}$. Then anyone who knows $s^{(t-1)}$ and d_u can compute all of the session keys after round t.*

(Proof) It is obvious that anyone can obtain the session key $s^{(t)}$ from $s^{(t-1)}$, d_u and $H^{(t)}$ by performing the same computation as a valid user does (Remember that anyone can get $H^{(t)}$ because it is broadcast).

Next, consider round $t+1$. $u \in \mathcal{R}^{(t)}$ implies $u \in \Lambda^{(t')}$ for some round t' ($\leq t$). Therefore, $u \notin \Lambda^{(t+1)} \cup \Theta^{(t+1)}$ because $\Lambda^{(t')} \cap \Lambda^{(t+1)} = \emptyset$ and $\mathcal{U} \cap \Theta^{(t+1)} = \emptyset$. He thus can compute $s^{(t+1)}$ from $s^{(t)}$, d_u and $H^{(t+1)}$.

Similarly, he can compute all of the session keys after round t by himself.

□

The attacks

As a result of the above theorem, the following attacks are possible.

Secret publishing attack

Some malicious users may publish their personal decryption keys to destruct the system. We will discuss the security of MAM scheme in this case.

The authors analyzed the security in [47, section 4.2 and section 6]. They claim that any valid user cannot compute the system secret a_0 even if all of the excluded users publish their personal decryption keys.

However, their security analysis is not sufficient. The goal of an attack is not to obtain the system secret but to obtain the session keys. In fact, MAM scheme is not secure when *one* user (valid or excluded) publishes his personal decryption key. The detail is as follows.

Suppose an excluded user u publishes his personal decryption key d_u in round t. It is straightforward from Theorem 5.4.1 that any valid user in round $t-1$ can compute all of the session keys even after he is excluded. If a valid user u publishes his personal decryption key d_u in round t, then he should be excluded from the next round $t+1$ Hence, the attack succeeds by the similar argument as above.

Rejoining attack

Consider the case when the center allows users who were excluded in previous rounds to join the system again. For example, it happens in practical settings when some users did not pay for a while but start to pay the fee again.

In [47], the authors considered this case in section 7. As mentioned earlier, a set of excluded users is increasing and the personal decryption keys of excluded users cannot be reused in MAM scheme. They deal with this issue by assigning a pair of new ID and corresponding personal decryption key to each of those users who want to join again.

However, if the center allows an excluded user to join again, the user can get session keys even after he is excluded again.

Suppose a user u is once excluded before round t. He requests to join again from round t, so the center gives him the session key $s^{(t)}$ (or some information to compute it).

Now he has $s^{(t)}$ and d_u where $u \in \mathcal{R}^{(t)}$. Hence, he can compute all of the succeeding session keys even after he is excluded again.

Temporary leakage of a session key

Here, we consider the security of MAM scheme when the session key of some round is leaked to an excluded user. Even though this scenario is not discussed in [47], it is very likely to happen in many practical applications. For example, if the key management of some valid user is not strict enough, an excluded user may be able to obtain a session key or some malicious user may publish a session key before he is excluded. It is desirable that even if excluded users get some of the session keys, they cannot obtain any information about other session keys.

However, in MAM scheme, if *one* session key is leaked, then any excluded user can compute all of the succeeding session keys by himself.

It is easy to see from Theorem 5.4.1 that if a session key $s^{(t)}$ is leaked, then any excluded user $u \in \mathcal{R}^{(t)}$ can compute all of the succeeding session keys.

A coalition of users who are excluded in different rounds

Some malicious users may conspire to get session keys which they are not supposed to have. Even though it is not explicitly mentioned in [47] if this type of conspiracy is allowed or not, it is very likely to happen. In fact, if users who are not excluded conspire and withdraw from the system in different rounds, this conspiracy can be easily made.

As mentioned in [47, section 3], since it is obvious that a coalition can obtain the session keys while at least one member of the coalition is not excluded, it is worthless to consider the security against this attack. However, we should consider the security if the coalition can get session keys even after all members of the coalition are excluded.

We will show that MAM scheme is not secure against this conspiracy.

For simplicity, we assume the number of colluders is two. Suppose that user u_1 and u_2 are colluders and u_1 and u_2 are excluded from round t_1 and t_2 ($> t_1$) respectively. u_2 can compute $s^{(t_1)}$ because he is a valid user in round t_1. He gives $s^{(t_1)}$ to u_1. Then from Theorem 5.4.1, u_1 can compute the session keys even after round t_1 by himself. Consequently, u_1 can give $s^{(t_2)}$ to u_2 in round t_2. Now u_2 can get the session keys after the round.

Note that between two colluders, only one session key is sent for each direction. Each of them can obtain the session keys by himself and does not need to work together except two rounds.

5.5 Attack on YF Scheme and TT Scheme

For the AMM-NP revocation scheme shown in Section 5.4.1, M. Yoshida and Fujiwara [65] and then Tzeng and Tzeng [61] showed a black box tracing algorithm.

In this section, however, we show that their tracing algorithms are broken.

5.5.1 Attack on YF Tracing Algorithm

For AMM-NP revocation scheme, M. Yoshida and Fujiwara showed black box tracing algorithm as follows.

For every possible t-subset of users $\mathcal{R} = \{u_1, u_2, \ldots, u_t\}$ ($t \leq w$), compute a testing header

$$H' = \langle s'g^{ra_0}, g^r, (u_1, g^{rf(u_1)}), \ldots, (u_t, g^{rf(u_t)}), (j_{t+1}, g^{rf(j_{t+1})}), \ldots, (j_w, g^{rf(j_w)})\rangle.$$

Feed H' to the confiscated pirate decoder. If the decoder does not output the correct s', set $\mathcal{R}$ as a possible set of traitors. Output the smallest of all possible sets of traitors found in Step 4.

We now show our attack on the above black box tracing algorithm. Suppose that user u is a traitor. He chooses an innocent user u' randomly. He then creates a pirate decoder P such as follows. On input a header H, P can find the set of revoked users $\mathcal{R}$ from H. P then outputs $D_{d_u}(H)$ where D is the decryption algorithm if and only if $u \notin \mathcal{R}$ and $u' \notin \mathcal{R}$. Then the tracing algorithm cannot determine if user u is a traitor or user u' is a traitor.

Our attack is generalized as follows.

Theorem 5.5.1 *In a traitor tracing scheme, let D be the decryption algorithm. If there exists a pair of users u and u' each of whom can determine whether $D_{d_u}(H) = D_{d_{u'}}(H)$ with overwhelming probability for any (valid or invalid) header H, then the scheme does not have black box traceability.*

(Proof) Suppose user u constructed a pirate decoder $\tilde{D}$ which outputs $D_{d_u}(H)$ if and only if $D_{d_u}(H) = D_{d_{u'}}(H)$.

Then no black box tracing algorithm can accuse u as a traitor since u' can also construct a pirate decoder which behaves same as $\tilde{D}$. □

5.5.2 Attack on TT scheme

TT scheme [61] is the same as YF scheme except that $\deg f(x) = z \geq 2w-1$. They presented two tracing algorithms and the first one is the same as that of YF scheme. Hence, our attack succeeds for their first tracing algorithms.

We next show an attack on their second tracing algorithm.

Their second tracing algorithm is as follows.

If $\mathcal{S} = \{u_1, \ldots, u_m\}$, $m \leq w$ are suspects, choose a random polynomial $f'(x) = \sum_{i=0}^{z} a'_i x^i$ such that $f'(u) = f(u)$ for each $u \in \mathcal{S}$. Then compute a testing header

$$H' = \langle s' g^{r a'_0}, g^r, (j_1, g^{r f'(j_1)}), \ldots, (j_z, g^{r f'(j_z)}) \rangle$$

where $\{j_1, \ldots, j_z\}$ are indices other than $\{u_1, \ldots, u_m\}$.

Feed H' to the confiscated pirate decoder. If the decoder outputs the correct s', $\mathcal{S}$ are traitors.

It is possible for traitors to compute the correct s' if at least one traitor is in the suspect set $\mathcal{S}$. Therefore, this algorithm may accuse innocent users as traitors.

Moreover, even though their proof of the traceability of TT scheme is based on the following lemma[61, Lemma 1], it does not apply if the suspect set $\mathcal{S}$ does not include the set of traitors BAD.

Lemma 1 of [61] *For polynomials $f(x)$ and $f'(x)$ of degree z, the distributions of the headers constructed by $f(x)$ and $f'(x)$ are computationally indistinguishable assuming that the DDH problem is hard.*

As a matter of fact, BAD can detect whether the input is a valid or a testing header. BAD first decrypts the input using each of the personal decryption keys of BAD and compares the results. The input is deduce to be a valid header if all of the results are the same or a testing header otherwise.

Chapter 6

Universal Hasing and Identification Codes via Channels

6.1 Introduction

Suppose that a transmitter sends a message a to a receiver through a communication channel with Shannon capacity C_S by encoding message a. An (n, W, λ_1) transmission code is a code which satisfies

$$\Pr[a \text{ is selected } \mid a \text{ is transmitted}] \geq 1 - \lambda_1, \tag{6.1}$$

for each message a, where each codeword has length n and there are W messages. The rate of the transmission code is defined as

$$R_1 \triangleq \log W/n.$$

Shannon proved that $\max R_1$ is equal to C_S for any arbitrarily small λ_1. This model implicitly assumes that:

1. A bijection from messages to codewords exists (deterministic encoding).

2. Also, the decoding regions of messages are disjoint and the receiver selects one message after receiving a noisy version of the transmitted codeword.

Ahlswede and Dueck [1] introduced a new model called identification codes via channels (ID codes). In this model:

1. There are many codewords for each message and the transmitter chooses a codeword from among them probabilistically (probabilistic encoding).
2. The decoding regions of messages are not disjoint and the receiver chooses a list of messages after receiving a noisy version of the transmitted codeword.

An $(n, M, \lambda_1, \lambda_2)$ ID code is a code which satisfies (6.1) and

$$\Pr[b \text{ is selected } \mid a \text{ is transmitted }] \leq \lambda_2 \text{ for all } b \neq a, \tag{6.2}$$

for each message a, where each codeword has length n and there are M messages. The probabilities are taken over the coin tosses of the transmitter (to choose a codeword) as well as over the noise of the channel. The rate of an ID code is defined as

$$R_2 \triangleq \log\log M/n$$

(which is a double log !!). It was proven that $\max R_2$ is equal to C_S for any arbitrarily small (λ_1, λ_2) [1, 31, 32].

Han and Verdú subsequently introduced the model of identification plus transmission codes (IT codes) [31], where a central station wishes to transmit one of W messages to one of M terminals. Upon receiving a codeword, each terminal decides whether it is the intended recipient of the message and if so it decodes the message. The decoding reliability is measured by (λ_1, λ_2) as follows:

1. For each terminal i,

$$E[\Pr(c \text{ is selected by } i \mid c \text{ is transmitted to } i)] \geq 1 - \lambda_1. \tag{6.3}$$

2. For any pair of terminals $j \neq i$,

$$E[\Pr(j \text{ decides that it is the intended recipient}$$

$$\mid c \text{ is transmitted to } i)] \leq \lambda_2. \tag{6.4}$$

E is taken over all codewords c for terminal i in both equations. An $(n, M, W, \lambda_1, \lambda_2)$ IT code is a code which satisfies (6.3) and (6.4), where each codeword has length n and there are W codewords and M terminals. The rate pair of an IT code is defined as

$$(R_1, R_2) \triangleq (\log W/n, \log\log M/n).$$

Han and Verdú proved that (C_S, C_S) is the maximum achievable rate pair for any arbitrarily small (λ_1, λ_2) [31]. This is a noticeable improvement over encoding the address and the message separately.

Explicit constructions of IT codes and ID codes which achieve the above limits were given by Verdú and Wei [62]. They first showed that an IT code is obtained by concatenating a transmission code with a binary constant weight code. (An $(n, M, \lambda_1, \lambda_2)$ ID code is obtained from an $(n, M, W, \lambda_1, \lambda_2)$ IT code easily.) They then showed explicit constructions of a sequence of binary constant weight codes which is *optimum for identification* (a SBCOI).

Their basic idea of constructing a binary constant weight code is to concatenate an error correcting code C over $GF(q)$ with a $[q]$ PPM code. Then their first explicit construction of SBCOIs is obtained by using an algebraic geometry code AG [35] as C. Algebraic geometry codes, however, require a nontrivial background on algebraic geometry and may not be directly accessible to most readers. Therefore, they then showed their second explicit construction of SBCOIs which uses a two-layer Reed-Solomon code. This is a conceptually simpler, more practical and even more explicit construction.

On the other hand, the notion of universal classes of hash functions was introduced by Carter and Wegman [18]. It has found numerous applications in cryptography, complexity theory and other areas [18, 63, 57, 56, 7] (see the Introduction in [57]). In particular, ϵ-almost strongly universal (ϵ-ASU) classes of hash functions have been studied and used for authentication codes [56].

This chapter shows that ϵ-ASU classes of hash functions can yield better explicit constructions of SBCOIs than the previous explicit constructions of Verdú and Wei. We first show that the incidence matrix of an ϵ-ASU class of hash functions naturally yields a binary constant weight code. Then we show two explicit constructions of SBCOIs. Our first explicit construction combines the algebraic geometry code AG with an ϵ-ASU class of hash functions. This yields a better SBCOI than the first construction of Verdú and Wei. Our second explicit construction combines a Reed Solomon code with an ϵ-ASU class of hash functions. This construction is not only as practical as the second construction of Verdú and Wei but also yields a better SBCOI. After all, ϵ-ASU classes of hash functions enable us to obtain better explicit constructions of IT codes and ID codes.

This chapter is organized as follows. The previous constructions of SBCOIs are summarized in Section 6.2. The background of ϵ-ASU classes of hash functions is given in Section 6.3. In Section 6.4, we show our explicit constructions of SBCOIs using ϵ-ASU classes of hash functions. In Section 6.5, we illustrate how to construct a practical IT code and a practical ID code by combining a Reed Solomon code with an ϵ-ASU class of hash functions.

6.2 Binary constant weight code and IT code [62]

6.2.1 Binary constant weight code

Definition 6.2.1 *An (L, M, W, K) binary constant weight code is a set of binary constant weight codewords such that*

- *the length of each codeword is L,*
- *the number of codewords is M,*
- *the Hamming weight of each codeword is W and*
- *the overlap (the number of coincident 1s) of any pair of codewords is at most K.*

Define

$$\beta \triangleq \log W / \log L, \ \ \rho \triangleq \log\log M / \log L, \ \ \mu \triangleq K/W. \tag{6.5}$$

β is called the weight factor, ρ is called the second order rate and μ is called the overlap factor of the binary constant weight code.

Verdú and Wei showed that an IT code can be obtained by concatenating a transmission code with a binary constant weight code [62].

Proposition 6.2.1 *Suppose that there exists an $(L, M, W, \mu W)$ binary constant weight code and an (n, L, λ) transmission code. Then there exists an $(n, M, W, \lambda, \lambda + \mu)$ IT code with the rate pair $(\beta R, \rho R)$, where R is the rate of the transmission code.*

It holds that [62]

$$\beta \leq 1, \ \rho \leq 1.$$

To make $(\beta R, \rho R)$ approach (C_S, C_S), R needs to approach C_S and (β, ρ) approach $(1,1)$. To make $\lambda_2 = \lambda + \mu$ small, μ needs to approach zero. This motivates the following definition.

Definition 6.2.2 *Consider a sequence $\{C_i\}$, where C_i is a (L_i, M_i, W_i, K_i) binary constant weight code with weight factor β_i, second order rate ρ_i, and pairwise overlap fraction μ_i. We say that the sequence of codes $\{C_i\}$ is optimal for identification if*

$$\beta_i \to 1,$$

$$\rho_i \to 1,$$

$$\mu_i \to 0.$$

(β and ρ should be as large as possible and μ should be as small as possible.)

6.2.2 Previous construction (I)

A binary constant weight code is obtained by concatenating an error correcting code with a PPM code.

Definition 6.2.3 *We denote an error correcting code by $C = (n, |C|, d)$, where n is the length of a codeword, $|C|$ is the number of codewords and d is the minimum Hamming distance.*

Definition 6.2.4 *Let $C_1 = (n_1, |C_1|, d_1)$ and $C_2 = (n_2, |C_2|, d_2)$ be error correcting codes with alphabets A_1 and A_2, respectively, such that $|A_2| = |C_1|$. Then $C = C_1 \circ C_2$ denotes a concatenated code $(n_1 n_2, |C_2|, d_{12})$ with alphabet A_1 such that*

$$C = \{(h(y_1), \ldots, h(y_{n_2})) \mid (y_1, \ldots, y_{n_2}) \in C_2\},$$

where h is a bijection from A_2 to C_1.

Definition 6.2.5 *A $(q, q, 1, 0)$ binary constant weight code (which consists of all binary q-vectors of unit weight) is called a $[q]$* PPM code.

Proposition 6.2.2 *For an error correcting code $C = (n, |C|, d)$ with alphabet size q and a $[q]$* PPM *code C_1, $C_1 \circ C$ is an $(nq, |C|, n, n-d)$ binary constant weight code.*

Verdú and Wei [62] showed that a sequence of binary constant weight codes which is optimum for identification (see Definition 6.2.2) is obtained by using a sequence of algebraic geometry codes $\{AG_i\}$ of the following Proposition as C of Proposition 6.2.2.

Proposition 6.2.3 *[35] Let $q = q_0^{2m}$ where q_0 is a prime. Suppose that*

$$R = 1 - (q^{1/2} - 1)^{-1} - \delta > 0.$$

Then, for any $\epsilon > 0$, there exists a sequence of q-ary codes of $\{AG_i\}$ increasing length whose asymptotic rate is greater than or equal to $R - \epsilon$ and whose asymptotic ratio of minimum (Hamming)distance to length is greater than or equal to $\delta - \epsilon$.

6.2.3 Previous construction (II)

The above construction, however, requires a nontrivial background on algebraic geometry and may not be directly accessible to most readers. Verdú and Wei [62] next presented a conceptually simpler, more practical and even more explicit construction of a sequence of binary constant weight codes which is optimum for identification. It can be understood without much background. This construction uses a two layer Reed-Solomon code as C of Proposition 6.2.2.

Definition 6.2.6 *Let q be a prime power and denote the elements of $GF(q)$ by $\{a_1, \ldots, a_q\}$. A $[q, k]$ Reed-Solomon code ($k < q$) is the set of q-vectors over $GF(q)$:*

$$\{(p(a_1), \ldots, p(a_q)) \mid p(x) \textit{ is a polynomial of degree } < k \\ \textit{with coefficients from } GF(q)\}.$$

In a $[q, k]$ Reed-Solomon code, the blocklength is q, the number of codewords is q^k, and the minimum distance is $q - k + 1$ because if two polynomials of degree $< k$ coincide at k or more places, then they are identical.

Proposition 6.2.4 *The $[q,k,t]$ three layer concatenated code $C_1 \circ C_2 \circ C_3$, with $C_1 = [q]$ PPM, $C_2 = [q,k]$ Reed-Solomon and $C_3 = [q^k, q^t]$ Reed-Solomon, with $t < k < q =$ prime power is a $(q^{k+2}, q^{kq^t}, q^{k+1}, kq^k + q^{1+t})$ binary constant weight code.*

Proposition 6.2.5 *Let C_i be a $[q_i, k_i, t_i]$ three layer concatenated code as in Proposition 6.2.4. The sequence of codes $\{C_i\}$ is optimal for identification if $t_i = i$, $k_i = i+1$, and q_i any increasing sequence of prime powers.*

6.3 Universal hash functions

Let X and Y be finite sets such that $|X| \geq |Y|$. Let H be a set of hash functions such that $h : X \rightarrow Y$ for each $h \in H$.

Definition 6.3.1 *[56] We say that H is an ϵ-*almost strongly universal *(ϵ-ASU) class of hash functions provided that the following two conditions are satisfied:*

1. *for any $x \in X$ and any $y \in Y$, there exists exactly $|H|/|Y|$ functions $h \in H$ such that $h(x) = y$.*

2. *for any two distinct elements $x_1, x_2 \in X$ and for any two (not necessarily distinct) elements $y_1, y_2 \in Y$, there exists at most $\epsilon|H|/|Y|$ functions $h \in H$ such that $h(x_i) = y_i$, $i = 1, 2$.*

Stinson showed a composition construction of an ϵ-ASU class of hash functions such as follows [56, Theorem 5.5].

Definition 6.3.2 *Let $C = (n, |C|, d)$ be an error correcting code over an alphabet X. Let H be an ϵ-ASU class of hash functions from X to Y. Then for $1 \leq \forall i \leq n$ and $\forall h \in H$, define a hash function $g_{(i,h)} : \{1, 2, \ldots, |C|\} \rightarrow Y$ by the rule*

$$g_{(i,h)}(x) = h(\textit{the } i\textit{th symbol of the } x\textit{th codeword of } C)$$

Let

$$H * C \triangleq \{g_{(i,h)}\}.$$

Proposition 6.3.1 *[56, Theorem 5.5] Let $C = (n, |C|, d)$ be an error correcting code over an alphabet X. Let H be an ϵ-ASU class of hash functions from X to Y. Then $H_C \triangleq H * C$ (defined as above) is an $\tilde{\epsilon}$-ASU class of hash functions from $\{1, 2, \dots, |C|\}$ to Y such that*

$$\tilde{\epsilon} = \epsilon + 1 - \frac{d}{n},$$

$$|H_C| = n|H|.$$

(Remark) In [56, Theorem 5.5], Stinson used the term *AU* class of hash functions. Bierbraur pointed out that it is equivalent to an error correcting code [7].

Let q be a prime power and let $1 < k < q$. Let

$$X \triangleq \{(a_1, \dots, a_k) \mid a_i \in GF(q)\}.$$

$$Y \triangleq \{\text{the elements of } GF(q)\}.$$

Boer showed a ϵ-ASU class of hash functions from X to Y such as follows [23].

Definition 6.3.3 *For $\forall(e_0, e_1)$ such that $e_0, e_1 \in GF(q)$, let*

$$h_{(e_0,e_1)}(a_1, \dots, a_k) = e_0 + a_1e_1 + \dots + a_ke_1^k.$$

Let

$$G(q, k) \triangleq \{h_{(e_0,e_1)}\}.$$

Proposition 6.3.2 *[23] The above $G(q, k)$ is a (k/q)-ASU class of hash functions from X to Y such that $|G(q, k)| = q^2$.*

Then the following corollary is obtained from Proposition 6.3.1.

Corollary 6.3.1 *Let $G(q, k)$ be a (k/q)-ASU class of hash functions from X to Y defined as above. Let $C = (n, |C|, d)$ be an error correcting code over $GF(q^k)$. Then $G(q, k)_C \triangleq G(q, k) * C$ is an $\tilde{\epsilon}$-ASU class of hash functions from $\{1, 2, \dots, |C|\}$ to Y such that*

$$\tilde{\epsilon} = \frac{k}{q} + 1 - \frac{d}{n},$$

$$|G(q, k)_C| = nq^2.$$

(Proof) Let $g_{(i,e_0,e_1)}$ be a hash function of $G(q,k)_C$, where $1 \le i \le n$ and $e_0, e_1 \in GF(q)$. Let the xth codeword of C over $GF(q^k)$ be $(A_1,\ldots,A_n)$. Let the vector representation of $A_i \in GF(q^k)$ be $(a_1,\ldots,a_k)$, where $\forall a_i \in GF(q)$. Then

$$\begin{aligned} g_{(i,e_0,e_1)}(x) &= h_{(e_0,e_1)}(\text{the } i\text{th symbol of the } x\text{th codeword of } C) \\ &= h_{(e_0,e_1)}(a_1,\ldots,a_k) \\ &= e_0 + a_1e_1 + \cdots + a_ke_1^k. \end{aligned}$$

Fix $x \in \{1,2,\ldots,|C|\}$ and $y \in Y$ arbitrarily. Let's compute the number N_0 of $g_{(i,e_0,e_1)}$ such that $g_{(i,e_0,e_1)}(x) = y$. That is,

$$e_0 + a_1e_1 + \cdots + a_ke_1^k = y. \tag{6.6}$$

Choose (i,e_1) arbitrarily. Then there exists unique e_0 which satisfies (6.6). Therefore

$$N_0 = |\{(i,e_0)\}| = nq = \frac{|G(q,k)_C|}{q}.$$

Next, fix $x_1, x_2 \in \{1,2,\ldots,|C|\}$ such that $x_1 \ne x_2$ and $y_1, y_2 \in Y$ arbitrarily. Let's compute the number N_1 of $g_{(i,e_0,e_1)}$ such that

$$g_{(i,e_0,e_1)}(x_1) = y_1, \quad g_{(i,e_0,e_1)}(x_2) = y_2.$$

Let the vector representation of the ith symbol of the x_1th codeword of C be $(a_1,\ldots,a_k)$ and that of x_2th codeword be $(b_1,\ldots,b_k)$. Then

$$e_0 + a_1e_1 + \cdots + a_ke_1^k = y_1,$$

$$e_0 + b_1e_1 + \cdots + b_ke_1^k = y_2.$$

Now we have

$$(a_k - b_k)e_1^k + \cdots + (a_1 - b_1)e_1 + (y_2 - y_1) = 0. \tag{6.7}$$

First suppose that $y_1 \ne y_2$. Then (6.7) has at most k solutions on e_1. Therefore,

$$N_1 \le |\{(i,e_1\}| = nk$$

Next, suppose that $y_1 = y_2$. There are at most $n - d$ positions i such that $(a_1,\ldots,a_k) = (b_1,\ldots,b_k)$ since the minimum Hamming distance of C is

d. For such i, (6.7) is satisfied by any e_1. For i such that $(a_1, \ldots, a_k) \neq (b_1, \ldots, b_k)$, (6.7) is satisfied by at most $(k-1)$ e_1. Therefore,

$$N_1 \leq (n-d)q + n(k-1).$$

In each case, N_1 is upper bounded by

$$\begin{aligned} N_1 &\leq (n-d)q + nk \\ &= nq\left(\frac{k}{q} + 1 - \frac{d}{n}\right) \\ &= \left(\frac{k}{q} + 1 - \frac{d}{n}\right) \cdot |G(q,k)_C|/q \end{aligned}$$

□

6.4 Proposed construction of binary constant weight codes

In this section, we show explicit constructions of a sequence of binary constant weight codes which is optimum for identification (a SBCOI for short) by using ϵ-ASU class of hash functions. The proposed constructions give better binary SBCOIs than the previous constructions.

6.4.1 ϵ-ASU implies binary constant weight code

Let H be an ϵ-ASU class of hash function from X to Y. The incidence matrix of H is a $|H| \times |X||Y|$ binary matrix such that each row is indexed by $h \in H$ and each column is indexed by $(x, y) \in X \times Y$ as follows.

$$(h, (x, y)) = \begin{cases} 1 & \text{if } h(x) = y \\ 0 & \text{otherwise} \end{cases}$$

Theorem 6.4.1 *Let H be an ϵ-ASU class of hash functions from X to Y. Then the set of column vectors of the incidence matrix of H is a $(|H|, |X||Y|, |H|/|Y|, \epsilon|H|/|Y|)$ binary constant weight code.*

(Proof) It is clear that the length of each codeword is $|H|$ and the number of codewords is $|X||Y|$. The last two parameters come from Definition 6.3.1.

□

Then from Corollary 6.3.1, we obtain the following explicit construction of a binary constant weight code.

Corollary 6.4.1 *Suppose that there exists an error correcting code* $C = (n, |C|, d)$ *over* $GF(q^k)$*. Then there exists a* $(nq^2, q|C|, nq, nq(\frac{k}{q} + 1 - \frac{d}{n}))$ *binary constant weight code. This is obtained as the set of column vectors of the incidence matrix of* $G(q,k)_C = G(q,k) * C$*. (See Definition 6.3.3 for* $G(q,k)$ *and see Definition 6.3.2 for* $G(q,k) * C$*.).*

6.4.2 Proposed construction (I)

We show that Corollary 6.4.1 yields a better binary constant weight code than Proposition 6.2.2 from the same error correcting code C over $GF(q^k)$. Let β_1, ρ_1 and μ_1 be the weight factor, the second order rate and the overlap factor, respectively, of the binary constant weight code of Corollary 6.4.1. Then

$$\beta_1 = \frac{\log nq}{\log nq^2}, \quad \rho_1 = \frac{\log\log |C|q}{\log nq^2}, \quad \mu_1 = 1 - \frac{d}{n} + \frac{k}{q}.$$

Let β_2, ρ_2 and μ_2 be those of Proposition 6.2.2. Then

$$\beta_2 = \frac{\log n}{\log nq^k}, \quad \rho_2 = \frac{\log\log |C|}{\log nq^k}, \quad \mu_2 = 1 - \frac{d}{n},$$

Consider small fixed k such that $k \geq 2$ (For example, $k = 2$). Then for sufficiently large q, we have

$$\beta_1 > \beta_2, \quad \rho_1 > \rho_2, \quad \mu_1 \approx \mu_2. \tag{6.8}$$

This shows that Corollary 6.4.1 yields a better binary constant weight code than Proposition 6.2.2 because β and ρ should be as large as possible.

Therefore, a SBCOI is obtained if we use a sequence of the algebraic geometry codes $\{AG_i\}$ of Proposition 6.2.3. This SBCOI is better than that of Section 6.2.2 from (6.8).

6.4.3 Proposed construction (II)

A $[q^k, q^t]$ Reed-Solomon code is a code over $GF(q^k)$ such that the length of a codeword is $n = q^k$, the number of codewords is $|C| = (q^k)^{q^t}$ and the minimum Hamming distance is $d = q^k - q^t + 1$. Then from Corollary 6.4.1, we have the following corollary.

Corollary 6.4.2 *There exists a* $(q^{k+2}, q^{kq^t+1}, q^{k+1}, kq^k + q^{1+t} - q)$ *binary constant weight code, where* $t < k < q =$ *prime.*

We denote the above binary constant weight code by $RB(q, k, t)$.

Let β_3, ρ_3 and μ_3 be the weight factor, second order rate and the overlap factor, respectively, of the binary constant weight code of $RB(q, k, t)$. Then

$$\beta_3 = \frac{\log q^{k+1}}{\log q^{k+2}}, \quad \rho_3 = \frac{\log \log q^{kq^t+1}}{\log q^{k+2}}, \quad \mu_3 = \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}. \tag{6.9}$$

Let β_4, ρ_4 and μ_4 be those of the $[q, k, t]$ three layer concatenated code of Proposition 6.2.4. Then

$$\beta_4 = \frac{\log q^{k+1}}{\log q^{k+2}}, \quad \rho_4 = \frac{\log \log q^{kq^t}}{\log q^{k+2}}, \quad \mu_4 = \frac{k}{q} + \frac{1}{q^{k-t}}.$$

Therefore,

$$\beta_3 = \beta_4, \quad \rho_3 > \rho_4, \quad \mu_3 < \mu_4.$$

This shows that $RB(q, k, t)$ is a better binary constant weight code than the $[q, k, t]$ three layer concatenated code of Proposition 6.2.4 because ρ should be as large as possible and μ should be as small as possible.

Therefore, the sequence of binary constant weight codes $\{RB(q_i, k_i, t_i)\}$ is optimal for identification if $t_i = i$, $k_i = i + 1$, and q_i any increasing sequence of prime powers from Proposition 6.2.5. This is a better SBCOI than the SBCOI obtained from the $[q, k, t]$ three layer concatenated code.

6.5 Practical IT code and ID code

By using Corollary 6.4.2, we can construct a practical IT-code and a practical ID-code. Suppose that there exists an (n, q^{k+2}, λ) transmission code. Index each codeword of this code by (y, e_0, e_1), where y is an element of $GF(q^k)$, e_0 and e_1 are elements of $GF(q)$.

6.5.1 Practical IT code

By combining Corollary 6.4.2 with Proposition 6.2.1, we can construct a practical IT-code such as follows.

Corollary 6.5.1 *Suppose that there exists an (n, q^{k+2}, λ) transmission code. Then there exists an $(n, q^{kq^t+1}, q^{k+1}, \lambda, \lambda + \mu_3)$ IT-code with the rate pair $(\beta_3 R, \rho_3 R)$, where R is the rate of the transmission code, $t < k < q =$ prime and β_3, ρ_3, μ_3 are defined by (6.9).*

This IT code is described as follows. Suppose that there are $M \triangleq q^{kq^t+1}$ terminals and the size of messages is $W \triangleq q^{k+1}$.

1. Index each terminal by $(f(x), \alpha)$, where $f(x)$ is a polynomial over $GF(q^k)$ such that $\deg f(x) < q^t$ and α is an element of $GF(q)$.
2. Index each message by (y, e_1), where y is an element of $GF(q^k)$ and e_1 is an element of $GF(q)$.

Now suppose that a central station T wishes to transmit the (y, e_1)th message to the $(f(x), \alpha)$th terminal. Then T first computes $f(y)$. Let the vector representation of $f(y) \in GF(q^k)$ be $(a_1, \ldots, a_k)$, where a_i is an element of $GF(q)$. Next T computes e_0 such that

$$e_0 + a_1 e_1 + \cdots + a_k e_1^k = \alpha. \tag{6.10}$$

Finally, T broadcasts the (y, e_0, e_1)th codeword of the transmission code.

6.5.2 Practical ID code

From the sentences between Theorem 2 and Definition 5 of [62], we can construct a practical ID-code such as follows.

Corollary 6.5.2 *Suppose that there exists an (n, q^{k+2}, λ) transmission code. Then there exists an $(n, q^{kq^t+1}, \lambda, \lambda+\mu_3)$ ID-code with the rate pair $(\beta_3 R, \rho_3 R)$, where R is the rate of the transmission code, $t < k < q =$ prime and β_3, ρ_3, μ_3 are defined by (6.9).*

This ID code is described as follows. Suppose that there are $M \triangleq q^{kq^t+1}$ messages. Index each message by $(f(x), \alpha)$, where $f(x)$ is a polynomial over $GF(q^k)$ such that $\deg f(x) < q^t$ and α is an element of $GF(q)$.

Now suppose that a transmitter T wishes to transmit the $(f(x), \alpha)$th message to a receiver. Then T first chooses (y, e_1) at random, where y is an element of $GF(q^k)$ and e_1 is an element of $GF(q)$. Then T computes $f(y)$.

Let the vector representation of $f(y) \in GF(q^k)$ be $(a_1, \ldots, a_k)$, where a_i is an element of $GF(q)$. Next T computes e_0 such that

$$e_0 + a_1 e_1 + \cdots + a_k e_1^k = \alpha.$$

Finally, T transmits the (y, e_0, e_1)th codeword of the transmission code.

Chapter 7

Conclusion

In this chapter, we summarize the results we have obtained in this work. and provide suggestions for future research.

In Chapter 2, we first presented two tight lower bounds on the size of the secret keys of each user in an unconditionally secure one-time use conditional access scheme (OTCAS). Then we have shown how to construct a computationally secure multiple-use conditional access scheme (MCAS) from a key predistribution scheme (KPS) by using the ElGamal cryptosystem. We have proven that our MCAS is secure against chosen (message, privileged subset of users) attacks if the ElGamal cryptosystem is secure and if the original KPS is simulated. This is the first MCAS with security that is proven formally.

In Chapter 3, we have shown an efficient construction of a class of conditional access schemes. We say that a conditional access scheme is a (w, n)-revocation scheme if a center can exclude w or less users among n users. In this chapter, we have presented efficient (w, n)-revocation schemes such that $\rho_T = O(w^{1+\epsilon})$ and $\rho_T = 1 + \epsilon$ for any $\epsilon > 0$ where transmission rate ρ_T is defined as

$$\rho_T \overset{\triangle}{=} \frac{\text{the length of a ciphtertext}}{\text{the length of a plaintext}}$$

by showing new constructions of cover free families. We have also shown a construction of cover free families which yields a (w, n)-revocation scheme such that not only $\rho_T = O(w^2)$ but it can also be used as a w-resilient traceability scheme.

In Chapter 4, we have shown that three public-key (k, n)-traceability schemes can be derived from a $[n, u, d]$-linear code C such that $d \geq 2k + 1$. The previous schemes are obtained as special cases. This observation provides more freedom and new insight into this field. For example, we have demonstrated that Boneh-Franklin scheme[14] is equivalent to a slight modification of the corrected Kurosawa-Desmedt scheme[39]. This means that BF scheme is redundant or overdesigned because the modified KD scheme is much simpler. It was also shown that the corrected KD scheme is the best among them. In addition, we have shown a tracing algorithm which can detect *all* traitors by using a confiscated pirate decoder as a black box. This algorithm is applicable to all the public-key traceability schemes discussed in this chapter and the trivial scheme. This is the first black box full tracing algorithm with traceability that is formally proven.

In Chapter 5 we have provided attacks and comments on some of the revocation and tracing schemes. They include Chor, Fiat and Naor(CFN) traceability scheme[19, 20], Naor, Naor and Lotspiech(NNL) revocation schemes with traceability[48], Matsuzaki, Anzai and Matsumoto(MAM) revocation scheme[47], Yoshida and Fujiwara(YF) revocation scheme with traceability[65] and Tzeng and Tzeng(TT) revocation scheme with traceability[61]. For example, Naor, Naor and Lotspiech showed two revocation schemes and a traitor tracing algorithm for them at Crypto 2001[48]. However, we have illustrated that NNL revocation schemes cannot be traceable.

In Chapter 6, we have shown that ϵ-almost strongly universal classes of hash functions can yield better explicit constructions of identification codes via channels (ID codes) and identification plus transmission codes (IT codes) than the previous explicit constructions of Verdú and Wei.

Finally, we provide suggestions for future research. We have proposed an efficient conditional access scheme with traceability. However, not many such schemes are known so far. In particular, conditional access schemes with black box traceability should be studied further. We have introduced the first black box full tracing algorithm with traceability that is formally proven. However, it is not efficient and improvement of the algorithm must be considered. There may or may not exist better algorithms. But explicit description of better algorithms or the proof that there exists no better algorithm should be provided. In all of the previous works on black box traceability, it is assumed that pirate decoders are "resettable." Whether or not this assumption is essential is still open. Although it seems to be critical in black box *full* traceability, it will be interesting to consider if it is possible to weaken the assumption.

Acknowledgements

I would like to express my greatest appreciation to my supervisor, Professor Kaoru Kurosawa. His patience, interest, intuition and support were of the upmost value. In addtion, I would like to thank my other supervisor, Associate Professor, Wakaha Ogata for many fruitful discussions and her intellegent instruction; Professor Yvo Desmedt and Mike Burmester for their helpful comments; Professor Tomohiko Uematsu for introducing and inspiring us with "identification via channels;" Tetsu Iwata, a talented collegue and dear freind, for his helpful advice; the other members and ex-menbers of Ogata-Kurosawa Laboratory for their encouragements and many insightful discussions; members of The Centre for Applied Cryptographic Research at the University of Waterloo, for offering me a valuable seven-month stay; and all my friends in Canada for improving my english, especially Fatima Camara for proofreading portions of this thesis. Finally, I wish to thank my family for thier constant support.

Bibliography

[1] R. Ahlswede and G. Dueck, "Identification via Channels," *IEEE Transactions on Information Theory*, vol. 35, 15–29, 1989.

[2] J. Anzai, N. Matsuzaki, and T. Matsumoto, "A Quick Group Key Distribution Scheme with "Entity Revocation"," *Advances in Cryptology – ASIACRYPT '99*, vol. LNCS 1716, 333–347, 1999.

[3] L. A. Bassalygo and M. S. Pinsker, "Limited multiple-access of a nonsynchronous channel" (in Russian), *Problemy Peredachi Informatsii*, vol. 19, No. 8, 92–96, 1983.

[4] A. Beimel and B. Chor, "Communication in Key Distribution Schemes," *IEEE Transactions on Information Theory*, vol. 42, 19–28, 1996.

[5] M. Bellare, A. Boldyreva, and S. Micali, "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements," *Advances in Cryptology – EUROCRYPT 2000*, vol. LNCS 1807, 259–274, 2000.

[6] S. Berkovits, "How to Broadcast a Secret," *Advances in Cryptology – EUROCRYPT '91*, vol. LNCS 0547, 535–541, 1991.

[7] J. Bierbrauer, "Universal Hashing and Geometric Codes," *Designs, Codes and Cryptography*, vol. 11, 207–221, 1997.

[8] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Advances in Cryptology – EUROCRYPT '84*, vol. LNCS 209, 335–338, 1984.

[9] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," *Advances in Cryptology – CRYPTO '92*, vol. LNCS 740, 471–486, 1992.

[10] C. Blundo and A. Cresti, "Space Requirements for Broadcast Encryption," *Advances in Cryptology - EUROCRYPT '94*, vol. LNCS 950, 287–298, 1994.

[11] C. Blundo, L. A. Frota Mattos, and D. R. Stinson, "Trade-offs Between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution," *Advances in Cryptology - CRYPTO '96*, vol. LNCS 1109, 387–400, 1996.

[12] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Information and Computation*, vol. 146, 1–23, 1998.

[13] C. Blundo, L. A. Frota Mattos, and D. R. Stinson, "Generalized Beimel-Chor schemes for broadcast encryption and interactive key distribution," *Theoretical Computer Science*, vol. 200, 313–334, 1998.

[14] D. Boneh and M. Franklin, "An Efficient Public Key Traitor Tracing Scheme (Extended Abstract)," *Advances in Cryptology - CRYPTO '99*, vol. LNCS 1666, 338–353, 1999.

[15] D. Boneh and M. Franklin, "An efficient public key traitor tracing scheme *(full-version of [14])*," `http://crypto.stanford.edu/~dabo/`, 2001.

[16] E. Bresson and J. Stern, "Efficient Revocation in Group Signatures," *Public Key Cryptography 2001*, vol. LNCS 1992, 190–206, 2001.

[17] R. Canetti, T. Malkin, and K. Nissim, "Efficient Communication-Storage Tradeoffs for Multicast Encryption," *Advances in Cryptology - EUROCRYPT '99*, vol. LNCS 1592, 459–474, 1999.

[18] J. L. Carter and M. N. Wegman, "Universal Classes of Hash Functions," *Journal of Computer and System Sciences*, vol. 18, 143–154, 1979.

[19] B. Chor, A. Fiat, and M. Naor, "Tracing Traitors," *Advances in Cryptology - CRYPTO '94*, vol. LNCS 0839, 257–270, 1994.

[20] B. Chor, A. Fiat, M. Naor, and B. Pinkas, "Tracing Traitors," *IEEE Transactions on Information Theory*, vol. 46, 893–910, 2000.

[21] C. J. Colbourun and J. H. Dinitz (Ed.), Handbook of Combinatorial Designs, CRC press, 1996.

[22] Y. Desmedt and V. Viswanathan, "Unconditionally secure dynamic conference key distribution," *Proceedings 1998 IEEE International Symposium on Information Theory*, 1998.

[23] B. Den Boer, "A Simple and Key-Economical Unconditional Authentication Scheme," *Journal of Computer Security*, vol. 2, 65–71, 1993.

[24] A. G. D'yachkov, V. V. Rykov, and A. M. Rashad, "Superimposed distance codes," *Problems of Control and Information Theory*, vol. 18, 237–250, 1989.

[25] A. G. D'yachkov and V. V. Rykov, "Unconditionally secure dynamic conference key distribution," *Proceedings 1998 IEEE International Symposium on Information Theory*, 1998.

[26] A. G. D'yachkov and V. V. Rykov, "Combinatorial pooling designs for clone-library screening," *Workshop at Los Alamos National Laboratory*, 1998.

[27] A. Fiat and M. Naor, "Broadcast Enctyption," *Advances in Cryptology - CRYPTO '93*, vol. LNCS 0773, 480–491, 1993.

[28] A. Fiat and T. Tassa, "Dynamic Traitor Tracing," *Advances in Cryptology - CRYPTO '99*, vol. LNCS 1666, 354–371, 1999.

[29] E. Gafni, J. Staddon, and Y. L. Yin, "Efficient Methods for Integrating Traceability and Broadcast Encryption," *Advances in Cryptology - CRYPTO '99*, vol. LNCS 1666, 372–387, 1999.

[30] J. A. Garay, J. Staddon, and A. Wool, "Long-Lived Broadcast Encryption," *Advances in Cryptology - CRYPTO 2000*, vol. LNCS 1880, 333–352, 2000.

[31] T. S. Han and S. Verdú, "New Results in the Theory of Identification via Channels," *IEEE Transactions on Information Theory*, vol. 38, 14–25, 1992.

[32] T. S. Han and S. Verdú, "Approximation Theory of Output Statistics," *IEEE Transactions on Information Theory*, vol. 39, 752–772, 1993.

[33] J. Hastad, "Solving simultaneous modular equations of low degree," *SIAM Journal of Computing*, vol. 17, 336–341, 1988.

[34] H. D. L. Hollmann, J. H. van Lint, J. Linnartz, and L. M. G. M. Tolhuizen, "On Codes with the Identifiable Parent Property," *Journal of Combinatorial Theory, Series A*, vol. 82, 121–133, 1998.

[35] G. L. Katsman, M. A. Tsfasman, and S. G. Vladut, "Modular Curves and Codes with a Polynomial Construction," *IEEE Transactions on Information Theory*, vol. 30 part II, 353–355, 1984.

[36] A. Kiayias and M. Yung, "Self Protecting Pirates and Black-Box Traitor Tracing," *Advances in Cryptology - CRYPTO 2001*, vol. LNCS 2139, 63–79, 2001.

[37] H. Komaki, Y. Watanabe, G. Hanaoka, and H. Imai, "Efficient Asymmetric Self-Enforcement Scheme with Public Traceability," *Public Key Cryptography 2001*, vol. LNCS 1992, 225–239, 2001.

[38] R. Kumar, S. Rajagopalan, and A. Sahai, "Coding Constructions for Blacklisting Problems without Computational Assumptions," *Advances in Cryptology - CRYPTO '99*, vol. LNCS 1666, 609–623, 1999.

[39] K. Kurosawa and Y. Desmedt, "Optimum Traitor Tracing and Asymmetric Schemes," *Advances in Cryptology - EUROCRYPT '98*, vol. LNCS 1403, 145–157, 1998.

[40] K. Kurosawa, T. Yoshida, Y. Desmedt, and M. Burmester, "Some Bounds and a Construction for Secure Broadcast Encryption," *Advances in Cryptology - ASIACRYPT '98*, vol. LNCS 1514, 420–433, 1998.

[41] K. Kurosawa and T. Yoshida, "Strongly universal hashing and identification codes via channels," *IEEE Transactions on Information Theory*, vol. 45, 2091–2095, 1999.

[42] K. Kurosawa, T. Yoshida, and Y. Desmedt, "Inherently Large Traceability of Broadcast Encryption Scheme," *Proceedings 2000 IEEE International Symposium on Information Theory*, 464–464, 2000.

[43] K. Kurosawa, M. Burmester, and Y. Desmedt, "A Proven Secure Tracing Algorithm for the Optimal KD Traitor Tracing Scheme," *DIMACS*

workshop on management of digital intellectual property (April, 2000), Rump session of EUROCRYPT 2000 (May, 2000), 2000.

[44] K. Kurosawa, T. Yoshida, "Linear Code Implies Public-Key Traitor Tracing," *Public Key Cryptography 2002*, vol. LNCS 2274, 172–187, 2002.

[45] M. Luby and J. Staddon, "Combinatrial Bounds for Broadcast Encryption," *Advances in Cryptology – EUROCRYPT '98*, vol. LNCS 1403, 512–526, 1998.

[46] T. Matsumoto and H. Imai, "On the Key Predistribution System: A Practical Solution to the Key Distribution Problem," *Advances in Cryptology – CRYPTO '87*, vol. LNCS 293, 185–193, 1987.

[47] N. Matsuzaki, J. Anzai, and T. Matsumoto, "Light Weight Broadcast Exclusion Using Secret Sharing," *Proceedings of ACISP 2000*, vol. LNCS 1841, 313–327, 2000.

[48] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," *Advances in Cryptology – CRYPTO 2001*, vol. LNCS 2139, 41–62, 2001.

[49] M. Naor and O. Reingold, "Number theoretic constructions of efficient pseudo-random functions," *Proceedings of 38th IEEE Symposium on Foundations of Computer Science*, 458–467, 1997.

[50] M. Naor and B. Pinkas, "Threshold Traitor Tracing," *Advances in Cryptology – CRYPTO '98*, vol. LNCS 1462, 502–517, 1998.

[51] M. Naor and B. Pinkas, "Efficient Trace and Revoke Schemes," *Financial Cryptography*, vol. LNCS 1962, 1–20, 2000.

[52] R. Safavi-Naini and Y. Wang, "Sequential Traitor Tracing," *Advances in Cryptology – CRYPTO 2000*, vol. LNCS 1880, 316–332, 2000.

[53] A. Silverberg, J. Staddon, and J. L. Walker, "Efficient Traitor Tracing Algorithms Using List Decoding," *Advances in Cryptology – ASIACRYPT 2001*, vol. LNCS 2248, 175–192, 2001.

[54] J. Staddon, D. R. Stinson, and R. Wei, "Combinatrial Properties of Frameproof and Traceability Codes," *IEEE Transactions on Information Theory*, vol. 47, 1042–1049, 2000.

[55] M. A. Stadler, "Publicly Verifiable Secret Sharing," *Advances in Cryptology - EUROCRYPT '96*, vol. LNCS 1070, 190–199, 1996.

[56] D. R. Stinson, "Universal hashing and authentication codes," *Advances in Cryptology - CRYPTO '91*, vol. LNCS 576, 74–85, 1991.

[57] D. R. Stinson, "Combinatorial Techniques for Universal Hashing," *Journal of Computer and System Sciences*, vol. 48, 337–346, 1994.

[58] D. R. Stinson, "On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption," *Designs, Codes and Cryptography*, vol. 12, 215–243, 1997.

[59] D. R. Stinson and R. Wei, "Combinatorial properties and constructions of traceability schemes and frameproof codes," *SIAM Journal on Discrete Mathematics*, vol. 11, 41–53, 1998.

[60] D. R. Stinson and R. Wei, "Key Preassigned Traceability Schemes for Broadcast Encryption," *Selected Areas in Cryptography '98*, vol. LNCS 1556, 144–156, 1998.

[61] W.-G. Tzeng and Z.-J. Tzeng, "A Public-Key Traitor Tracing Scheme with Revocation Using Dynamic Shares," *Public Key Cryptography 2001*, vol. LNCS 1992, 207–224, 2001.

[62] S. Verdú and V. K. Wei, "Explicit Construction of Optimal Constant-Weight Codes for Identification via Channels," *IEEE Transactions on Information Theory*, vol. 39, 30–36, 1993.

[63] M. N. Wegman and J. L. Carter, "New Hash Functions and Their Use in Authentication and Set Equality," *Journal of Computer and System Sciences*, vol. 22, 265–279, 1981.

[64] J. J. Yan and Y. Wu, "An Attack on A Traitor Tracing Scheme," *IACR Cryptology ePrint Archive*, 2001.

[65] M. Yoshida and T. Fujiwara, "An Efficient Traitor Tracing Scheme for Broadcast Encryption," *Proceedings 2000 IEEE International Symposium on Information Theory, 463-463*, 2000.

[66] T. Yoshida, Y. Desmedt, and K. Kurosawa, "How to Break the Bound of Broadcast Encryption," *Submitted to CRYPTO '99*, 1999.

[67] T. Yoshida, K. Kurosawa, "How to Break Some Revocation and Tracing Schemes," *Submitted to EUROCRYPT 2002*, 2001.

Author's Contributions

Journals

1. K. Kurosawa and T. Yoshida, "Strongly universal hashing and identification codes via channels," *IEEE Transactions on Information Theory*, vol. 45, 2091–2095, 1999.

Conferences

1. K. Kurosawa, T. Yoshida, Y. Desmedt, and M. Burmester, "Some Bounds and a Construction for Secure Broadcast Encryption," *Advances in Cryptology - ASIACRYPT '98*, vol. LNCS 1514, 420–433, Beijing, China, October 18–22, 1998.

2. K. Kurosawa, T. Yoshida, and Y. Desmedt, "Inherently Large Traceability of Broadcast Encryption Scheme," *2000 IEEE International Symposium on Information Theory*, 464–464, Sorrento, Italy, June 25–30, 2000.

3. K. Kurosawa, T. Yoshida, "Linear Code Implies Public-Key Traitor Tracing," *Public Key Cryptography 2002*, vol. LNCS 2274, 172–187, Paris, France, February, 12–14, 2001.

Submitted

1. T. Yoshida, K. Kurosawa, "How to Break Some Revocation and Tracing Schemes," Submitted to EUROCRYPT 2002.