國立交通大學

資訊科學與工程研究所

碩士論文

低儲存量及常數計算量的
公開金鑰廣播加密系統

Public Key Broadcast Encryption with Low Number of Keys and Constant Decryption Time

研究生：劉易儒

指導教授：曾文貴　教授

中華民國九十六年六月

低儲存量及常數計算量的公開金鑰廣播加密系統
Public Key Broadcast Encryption with Low Number of Keys and Constant Decryption Time

研究生：劉易儒　　Student：Yi-Ru Liu
指導教授：曾文貴　　Advisor：Wen-Guey Tzeng

國立交通大學
資訊科學與工程研究所
碩士論文

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

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

低儲存量及常數計算量的公開金鑰廣播加密系統

學生：劉易儒　　　　　　　　指導教授：曾文貴博士

資訊科學與工程研究所

國立交通大學

摘要

在此篇論文中，我們提出了兩個公開金鑰廣播加密系統；第一種方法，我們把它稱為 BE-PI，它能夠達到$O(r)$的表頭長度(Header Size)、$O(r)$的計算量、$O(1)$的公開金鑰儲存量及$O(\log n)$ 的私密金鑰儲存量，其中r代表註銷使用者個數，n代表使用者個數，這是首次在傳輸量 $O(r)$ 的狀況下達到儲存量$O(\log n)$ 的公開金鑰廣播加密系統。另一種方法，我們把它稱做為 PK-SD-PI 系統，它可以做到 $O(r)$ 的表頭長度，$O(1)$的公開金鑰儲存量及$O(\log^2 n)$ 的私密金鑰儲存量，但它只需要 $O(1)$的計算量；此外，藉由和 LSD 類似的方式，我們可以再把它改變成 PK-LSD-PI，並得到$O(1)$的公開金鑰儲存量、$O(\log^{1+1/k})$ 的私密金鑰儲存量及$O(kr)$表頭長度之間的取捨。另外，利用我們的方法，也能降低之前公開金鑰背叛者追蹤方法中，公開金鑰數量至$O(1)$。我們廣播加密的方法，在選擇明文攻擊模式下(CPA)，具有抵擋完全共謀的安全性，經過些許改變，我們可以使我們系統達到抵擋選擇密文攻擊(CCA)的安全性。

關鍵字：廣播加密、撤銷機制、背叛者追蹤

Public Key Broadcast Encryption with Low Number of Keys and Constant Decryption Time

Student: Yi-Ru Liu Advisor: Dr. Wen-Guey Tzeng

Institute of Computer Science and Computer Engineering

National Chiao Tung University

Abstract

We proposed two public-key broadcast encryption schemes. The first scheme, called the **BE-PI** scheme, has $O(r)$ header size, $O(r)$ computation cost, $O(1)$ public keys and $O(\log n)$ private keys, where r is the number of revoked users and n is the number of users. This is the first public-key BE(broadcast encryption) scheme that with $O(\log n)$ private keys under $O(r)$ header size. The other scheme, we call it **PK-SD-PI** scheme, has $O(r)$ header size, $O(1)$ public keys, $O(\log^2 n)$ private keys and only $O(1)$ computation cost. By using similar technique in LSD. We can convert it to PK-LSD-PI scheme, has $O(1)$ public keys and $O(\log^{1+1/k})$ private keys with $O(kr)$ header size tradeoff. Using our method, it also can reduce public key size to $O(1)$ in public traitor tracing scheme. Our BE system is static full-collusion resistant secure under chosen plain attack (CPA). With little modification, it can also against chosen cipher attack (CCA).

Key words: Broadcast Encryption, Revocation Scheme, Traitor Tracing

誌　　　謝

我首先要感謝我的指導老師曾文貴教授，在我碩士班的學習過程中，帶領我深入了解密碼學及資訊安全的領域，老師認真的研究精神，亦使我受益良多。另外我要感謝口試委員，交大資工蔡錫鈞教授、清大資工孫宏民教授及中央研究院資訊科學研究所呂及人教授，在論文上給我許多的建議與指導，讓我的論文更加完善。除此之外，我要感謝實驗室學長朱成康和學姐林孝盈的指導，實驗室同學陳仕烽、周昆逸、實驗室學弟沈宣佐、陳宏達、高翎恩、曾智揚及學妹廖怡翔，和你們一起討論和學習是一件愉快的事。

最後，我要感謝我的父母，不論在精神或物質上都給我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，以此文獻給所有我想感謝的人。

Contents

Chapter1

Introduction

Broadcast Encryption schemes enable a center to deliver encrypted data to a large set S of N users. For any set S , we can deliver an encrypted message to users $u \in S$, while the users $u \notin S$ cannot get information about the message. Such schemes are useful in pay-TV systems, the distribution of copyrighted material on encrypted CD/DVD disks, internet multicasting of video, music and magazines, etc..

In 1993, Noar and Fiat [1] formalized the basic definitions and paradigms of this field. A broadcasted message M is usually sent in the form $< Hdr(S,k), E_k(M) >$, where k is a session key for encrypting M via a symmetric encryption method E . An authorized user in S can use his private keys to decrypt the session key k from $Hdr(S,k)$, then use k to decrypt message M .The performance measures of a broadcast encryption scheme are the header size, the size of private keys held by each user, the size of public keys and the time for decryption. A broadcast encryption scheme should be able to resist the collusion attack from revoked users. A scheme is fully *collusion-resistant* if even all revoked users collude, they get no information about the broadcasted message.

Broadcast encryption schemes can be static or dynamic. For a dynamic broadcast encryption scheme, the private keys of a user can be update from time to time, while the private keys of a user in a static broadcast encryption scheme remain the same through the lifetime of the system. Broadcast encryption schemes can be public-key or secret-key. For a public key broadcast encryption scheme, any one can broadcast a message to an arbitrary group of authorized users by using the public system parameters, while for a secret-key broadcast encryption scheme, only the special dealer, who knows all secrets of the system can broadcast a message.

Our scheme is stateless public-key broadcast encryption system. The first stateless broadcast encryption system was proposed by Naor and Litspiech in 2001 [12]. They regard the problem of “designing stateless broadcast encryption as” “solving subset cover problem”. They present two methods .One is “Complete Subtree”(CS) technique. In the CS algorithm, everyone needs to store $O(\log n)$ keys with $O(r\log(n/r))$ header size, where n is the number of users and r is the number of revoked users. Another major improvement of there idea was the “subset diffenrence”(SD) technique. They use pseudorandom function to reduce the number of keys for an user needs to store. They also break the lower bound based on the information security. In the SD algorithm, everyone needs to store $O(\log^2 n)$ keys with O(r) header size. In 2003, Dodis and Fazio [6] introduce how to transform a secret-key to a public-key BE system. They use “Identity-Based Encryption (IBE)” and “ Hierarchical Identity-Based Encryption (HIBE)”technique to transform CS and SD to public-key broadcast encryption scheme with O(1) public key and keeps private key parameter. In 2005, Boneh [4] proposed a pulic-key broadcast encryption system with short ciphertext and private key. However, this scheme needs very large public-key storage space ($O(n)$) and receivers need to use all public keys for decryption message.

First of our public BE scheme has $O(\log n)$ private keys and only $O(r)$ header size. The other one can reduce PK-SD computation cost to $O(1)$. Our scheme is static secure based on CBDH problem under CPA mode. We can build it with CCA secure by applying Fujisaka and Okamoto method[9].

Chapter2

Relative Work

Consider two trivial BE systems. If we assign everyone with only one key*, and we revoke r users .Then we will need to send n-r encrypted message to each user. The header size is $O(n)$. On the other hand, if we assign 2^n to each user **. The header size will become $O(1)$ but there are too many keys ($O(2^{n-1})$)need to be stored. How to get good performance between header size and storage size is the major problem.

Following table is the performance of stateless broadcast encryption schemes.

Private-key Broadcast Encryption：

Method	Header Size	Priv. key Size	Comp. cost
*	$O(n)$	$O(n-r)$	$O(1)$
**	$O(1)$	$O(2^{n-1})$	$O(n)$
CS[12]	$O(r\log(n/r))$	$O(\log n)$	$O(\log\log n)$
PRSG or OWF- based			
SD[12]	$O(r)$	$O(\log^2 n)$	$O(\log n)$
LSD[13]	$O(kr)$	$O(\log^{1+1/k} n)$	$O(\log n)$
Jho[14]	$O(\frac{r}{p+1}+\frac{N-(p+2)r/(p+1)}{c})$	$O(c^p)$	$O(c)$
SIC[15]	$O(kr)$	$O(\log n)$	$O(n^{1/k})$
RSA Accumulator-based			
Asano[16]	$O(r\log_a(n/r)+r)$	$O(1)$	$O(2^a\log_a^2 n)$

SIC[15]	$O(kr)$	$O(1)$	$O((n^{1/k}\log^2 n)/k)$

Public-key Broadcast Encryption:

Method	Header Size	Priv.key Size	Pub. Key Size	Decryption Comp.cost
CS-PK[6]	$O(r\log(n/r))$	$O(\log n)$	$O(1)$	$O(1)$
SD-PK[6]	$O(r)$	$O(\log^2 n)$	$O(1)$	$O(\log n)$
LSD-PK[6]	$O(kr)$	$O(\log^{1+1/k} n)$	$O(1)$	$O(\log n)$
BGW(i) [4]	$O(1)$	$O(1)$	$O(n)$	$O(n)$
BGW(ii)[4]	$O(\sqrt{n})$	$O(1)$	$O(\sqrt{n})$	$O(\sqrt{n})$
Our BE-PI	$O(r)$	$O(\log\ \ n)$	$O(1)$	$O(r)$
Our PK-SD-PI	$O(r)$	$O(\log^2 n)$	$O(1)$	$O(1)$

For designing stateless broadcast encryption schemes. We can regard it as an subset-cover problem:

For a set N={1,2,3,…,n} How to set subset $S_1,S_2,...S_w \subset N$ such that for any $R \subset N$ we can find $S_{i_1},S_{i_2},...,S_{i_t}$ where $S_{i_1} \cup S_{i_2} \cup ... \cup S_{i_t} = N \setminus R$

In the above system. We can regard *N* as users. All set S_i has an unique key K_i , elements in *S* are the users who have key K_i. *t* is the header size for sending message to subset $N \setminus R$. Key size for each user is the number of subsets a user belongs to.

2.1Private Key Broadcast Encryption

In private key BE system, only the server who knows all secrets can broadcast encrypted message. Here we introduce CS, SD, LSD and SIC schemes.

2.1.1 Complete Subtree (CS) Scheme

This scheme was proposed by Noar[12] in 2001. The collection of subsets $S_1,S_2,...S_w \subset N$ in this scheme corresponds to all complete subtrees in the full binary tree with N leaves. For any node v_i in the full binary tree , the subset S_i is the collection of receivers u that correspond to the leaves of the subtree rooted at node v_i .Following picture (Fig1) is an example

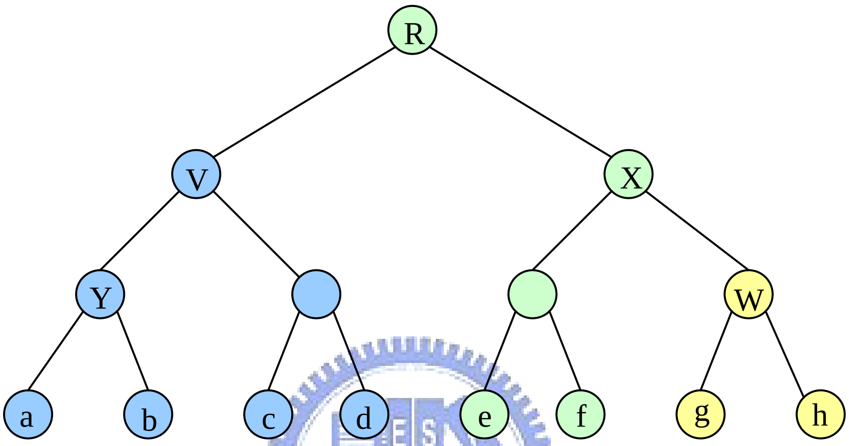


Figure 1 CS scheme

a,b,c,d,e,f ,g,h are users. $S_V = \{a.b.c.d\}$ $S_W = \{g,h\}$

The key assignment method simply assign each subset S_i an independent and random value K_i. It is easy to see that each user only needs to store value K_i where i is nodes on the path from root to user. For example, user b needs to store K_b, K_Y, K_V, K_R. In a full binary tree, we know that the height of the tree is $\log n$, so the key size for each user is $O(\log n)$.

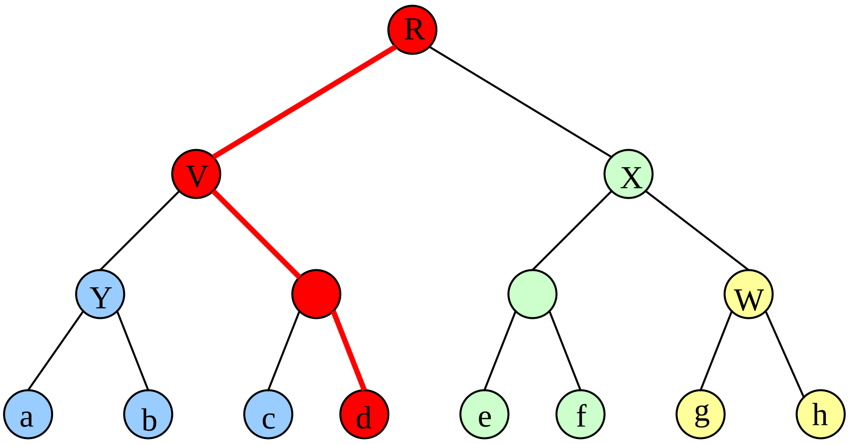


Figure 2 Revoke d in CS scheme

For a given set R of revoked receivers, we remove the edges and nodes from revoked receivers to root, and we get subtree $S_{i_1},\ldots,S_{i_t}$. If we revoke user d in figure 1. we get S_Y, S_c, S_X (Fig 2.). The header will be like :

$Hdr = < S_Y, S_c, S_X, E_{K_Y}(k), E_{K_c}(k), E_{K_X}(k) >$.

The cover size of CS scheme is at most $r\log(n/r)$.

2.1.2 Subset Difference(SD) Scheme

Disadvantage of the Complete Subtree method is that $N \setminus R$ may be partitioned into $r\log(n/r)$ subsets. It is large. Now we want to reduce the partition size. Consequently, we needs to increase subsets. A Subset $S_{i,j}$ in SD scheme is the $S_i - S_j$ in CS scheme (Fig3.).

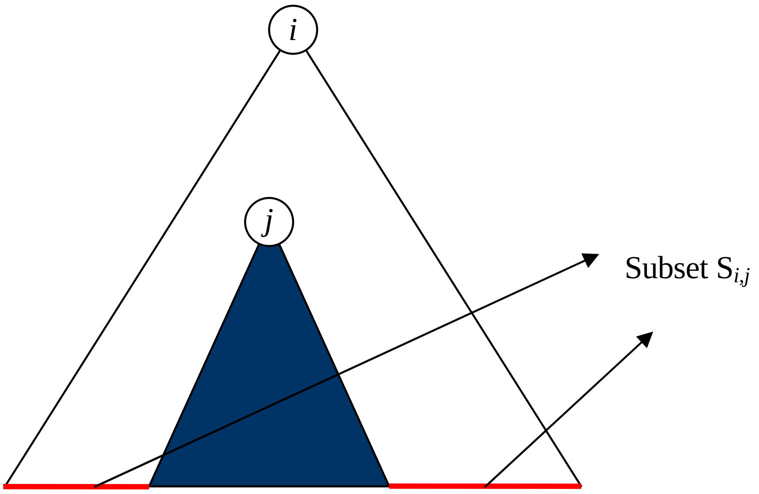
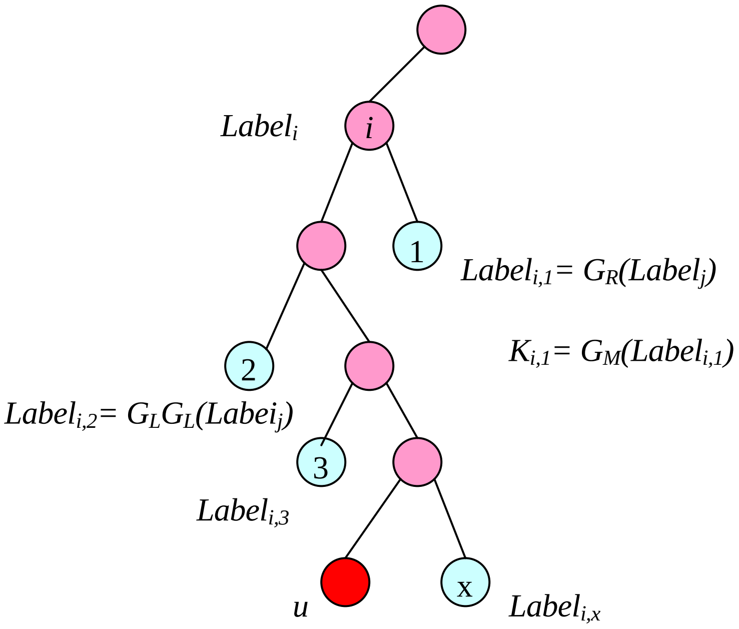


Figure 3 Subset $S_{i,j}$ in SD scheme

SD scheme partitions the non-revoked receivers into at most *2r-1* subsets. However, there are $O(n)$ subsets for an user belongs in. It means that everyone needs to store $O(n)$ keys. It is very impractical, so we use pseudorandom function to derive keys from parent's label. Let $G:\{0,1\}^n \rightarrow \{0,1\}^{3n}$ be a pseudo-random sequence generator that triples the input, whose output length is three times the length of the input; For each node v_i. We assign it a $label_i = \{0,1\}^{3n}$. Let $G_R(S)$ denote the left third of the output of G on seed S, $G_R(S)$ the right third and $G_M(S)$ the middle third. We say that $G\{0,1\}^n \rightarrow \{0,1\}^{3n}$ is a pseudo-random sequence generator if no polynomial-time adversary can distinguish the output of G on a randomly chosen seed from a truly random string of similar length. Now, consider the subtree T_i (root at v_i). j_L and j_R are $i's$ left and right child. We will use the following top-down labeling process. The root is assigned a label L_i.The label L_{i,j_L} is computed from $G_L(L_i)$ and L_{i,j_R} is computed from$G_R(L_i)$.The key $K_{i,j}$ of set $S_{i,j}$ is derived from $G_M(L_{i,j})$. Therefore, if we get the label value of v_i , we can derive all keys$K_{i,j}$={ $K_{i,j}$ | *j* is an descendant of *i*} of subset $S_{i,j}$. Now, each user only needs to store *Label*$_{i,j}$= { $label_{i,j}$ | *i* is ancestor of u , and for each *i* , *j* is the sibling of nodes on the path from user to *i* }.For example :



Label$_i$
i
1
Label$_{i,1}$= G$_R$(Label$_j$)
K$_{i,1}$= G$_M$(Label$_{i,1}$)
2
Label$_{i,2}$= G$_L$G$_L$(Labei$_j$)
3
Label$_{i,3}$
u
x
Label$_{i,x}$

Figure 4 Key generation in SD scheme

In Figure 4, for node *i*, user *u* needs to store $L_{i,1}, L_{i,2}, L_{i,3}, L_{i,x}$. We can discover that an user in a *n*-users SD-BE system needs to store :

$$1+\sum_{k=1}^{\log n+1} k-1=\frac{1}{2}\log^2 n+\frac{1}{2}\log n+1=O(\log^2 n)$$ values.

The Cover. For a set R of revoked receivers, we find Steiner Tree $ST(R)$ with the property that any $u \in N \setminus R$ that is below a leaf of tree has been covered. We start by making $T = ST(R)$ and then iteratively remove nodes from T until T consists of just a single node:

1. Find two leaves v_i and v_j in T such that there least-common-ancestor v does not contain any other leaf of T. Let v_l and v_k be the two children of v .v_k is ancestor of v_j and v_l is ancestor of v_i. ($v_l = v_k = v$ when there is only one leaf left)

2. If $v_l \neq v_i$ then add the subset $S_{l,i}$ to the collection; likewise, if $v_k \neq v_j$ add the subset $S_{k,j}$ to the collection.

3. Remove from T all the descendants of v and make it a leaf

A cover in SD scheme contains at most *2r-1* subsets for any set of *r* revocations.

2.1.3 LSD Scheme

In 2002, Halevy and Shamir propose Layer Subset Difference (LSD) method which can reduce key size to $O(\log^{1+1/k})$ with header size $O(kr)$.

Here we describe the simplest version of the Layered Subset Difference scheme where *k*=2.

In LSD, set partition is the same as SD scheme. A set $S_{i,j}$ we can split it into $S_{i,k} \cup S_{k,j}$ (*k* is a descendant of *i* and *j* is a descendant of *k*). Figure 5 demonstrates the set $S_{i,k} \cup S_{k,j} = S_{i,j}$.

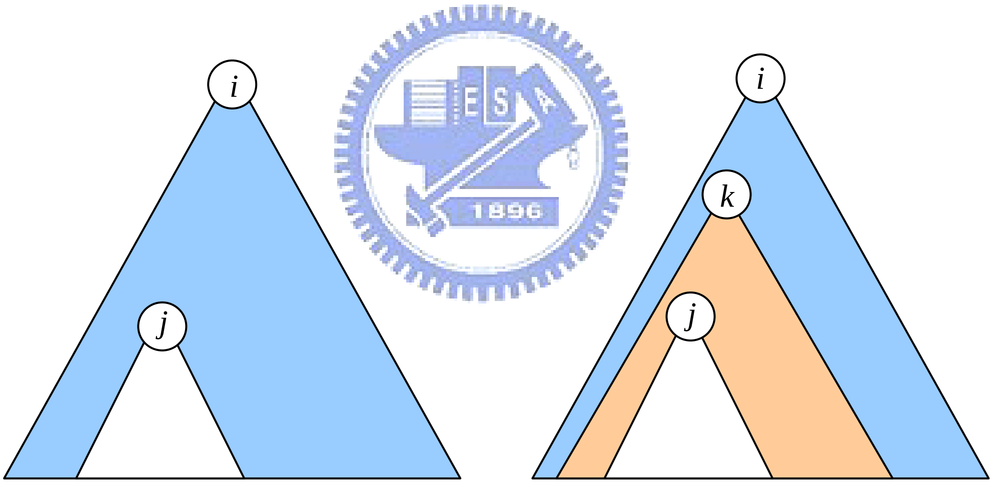


Figure 5 Subset in LSD

We define some of the $\sqrt{\log n}$ levels as "special". The root is considered to be at special level, and in addition we consider every level of depth $t \cdot \sqrt{\log(n)}$ *for* $t = 1 ... \sqrt{\log(n)}$ as special. We define set $S_{i,j}$ is an useful set if i and j belong to the same layer or i is at a special layer. Any set in SD we can present by at most two useful sets. The keys need to be stored for each user *u* is similar to SD scheme, but it only need to store $L_{i,j}$ where $S_{i,j}$ is an useful set. For example, user *u* in Figure6 for ascendant *i*. The labels he needs to store is the same in SD scheme, but for ascendant V, he only needs to store $L_{V,2}$.

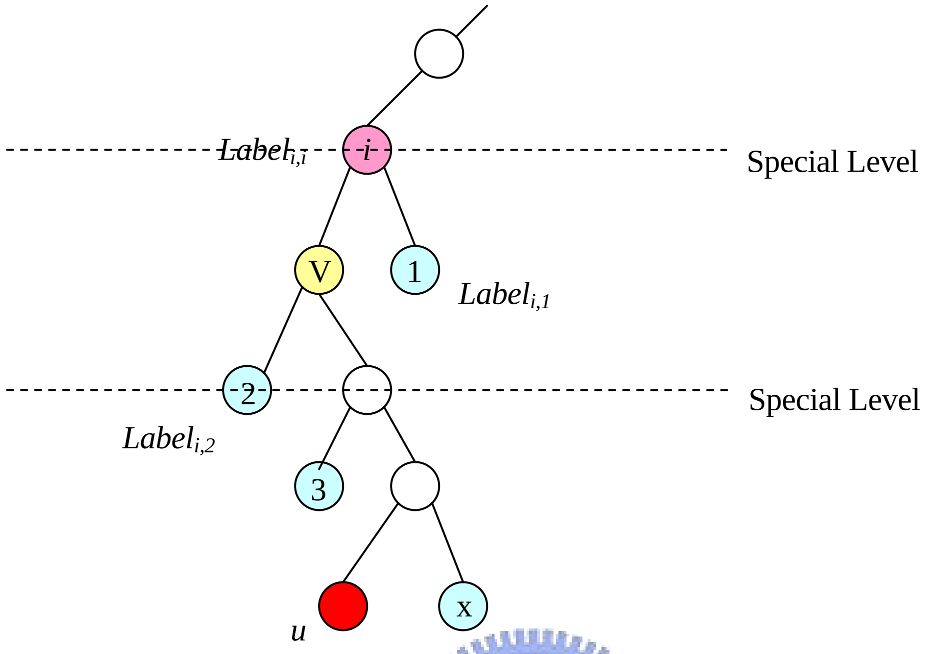


Figure 6 LSD scheme.

The total number of keys an user needs to store for each layered is $O(\sqrt{\log n}^2) = O(\log n)$. There are $\sqrt{\log n}$ layers. The total storage size is $O(\log n)O(\log^{1/2} n) = O(\log^{3/2} n)$.

Any subset in SD scheme is at most divided into two subsets in this scheme. So header size is at most 4r-2.

Using the similar method, we can divide a subset $S\backslash R$ into more subsets and get $O(\log^{1+1/k} n)$ storage size with $O(kr)$ header size tradeoff.

2.1.4 SIC Scheme

Addrapadung proposed Subset Incremental Chain (SIC) [15] scheme in 2005. This scheme improves storage size to $O(\log n)$ and header size to $O(r)$ with $O(n)$ computation cost. Using RSA-Accumulator technique, it can reduce storage size to O(1), but it needs more computation for finding primes. This scheme also can be layered. We introduce no layered situation.

Graph-decomposition.

This paper's authors give a method to analyze the relationship between keys. For any set *S*, we can regard it as a node . If set $A \subset B$ there is a direct path from A to B . In following example $S_{toy} = \{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{2,3\},\{2,4\},\{3,4\},\{1,2,3\}\}$

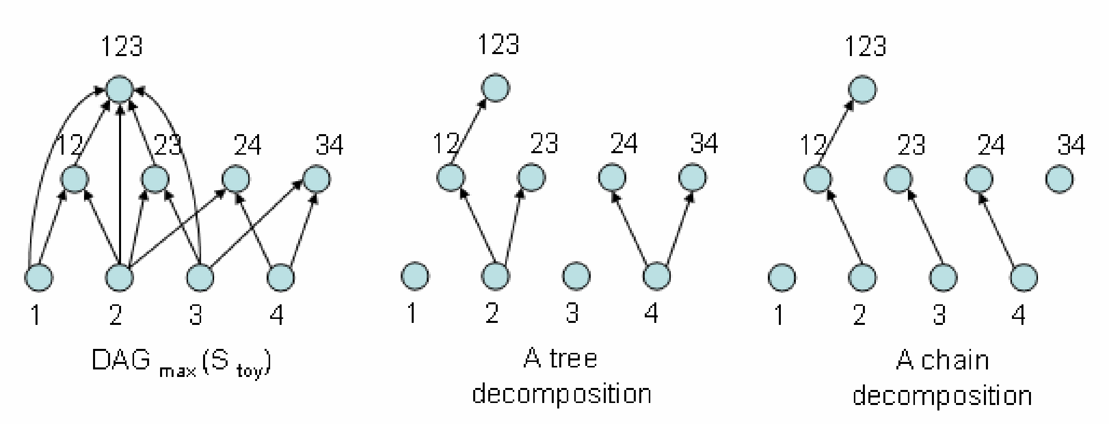


Figure 7 Graph-decomposition.

Using DAG graph, we can easily reduce it to chain decomposition and find that when using pseudorandom function ,we can derive all keys from five independent values $K_1, K_2, K_3, K_4, K_{34}$.

The Cover. In this scheme, we define following notaions

For $i, j \in N = \{1,2,...,n\}$ and $i < j$ denote：

$$i \rightarrow j := \{\{i\},\{i,i+1\},...,\{i,..., j\}\},$$
$$i \leftarrow j := \{\{j\},\{j, j-1\},...,\{j,...,i\}\},$$

l_v ：The leftmost leaf under v

r_v ：The rightmost leaf under v

BT_L ：The set of internal nodes which are left children

BT_R ：The set of internal nodes which are right children

For root, we assign $1 \rightarrow n$ and $2 \leftarrow n$. For each internal node, if $v \in BT_L$ we assign it $l_v + 1 \leftarrow r_v$, otherwise $v \in BT_R$ we assign it $l_v \rightarrow r_v - 1$ .A 16 users example was shown in Fig8.

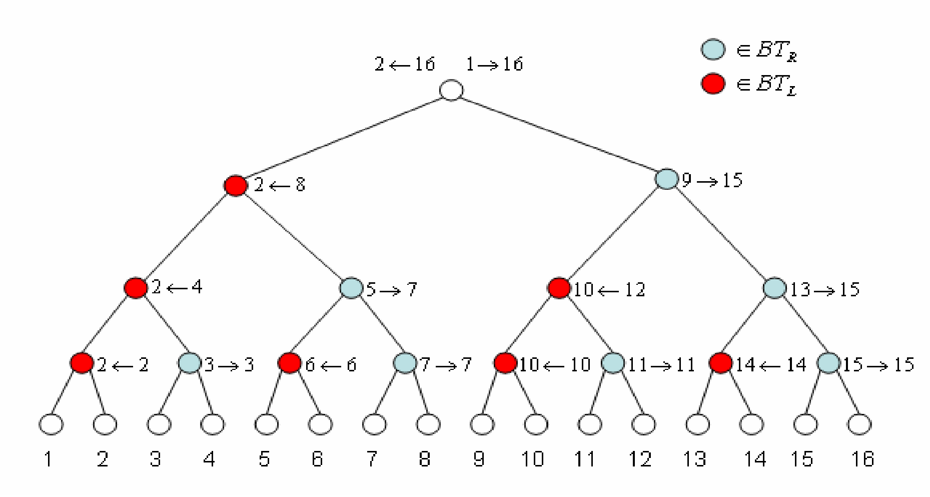


Figure 8 Sets of SIC scheme

All sets in SIC scheme are:

$$S_{SIC} = \bigcup_{v \in BT_L} (l_v + 1 \leftarrow r_v) \cup \bigcup_{v \in BT_R} (l_v \rightarrow r_v - 1) \cup (1 \rightarrow n) \cup (2 \leftarrow n)$$

Using previous graph decomposition method. We can arrange all sets into chain decomposition graph. For instance, we can arrange all sets in Figure8 into Figure 9.

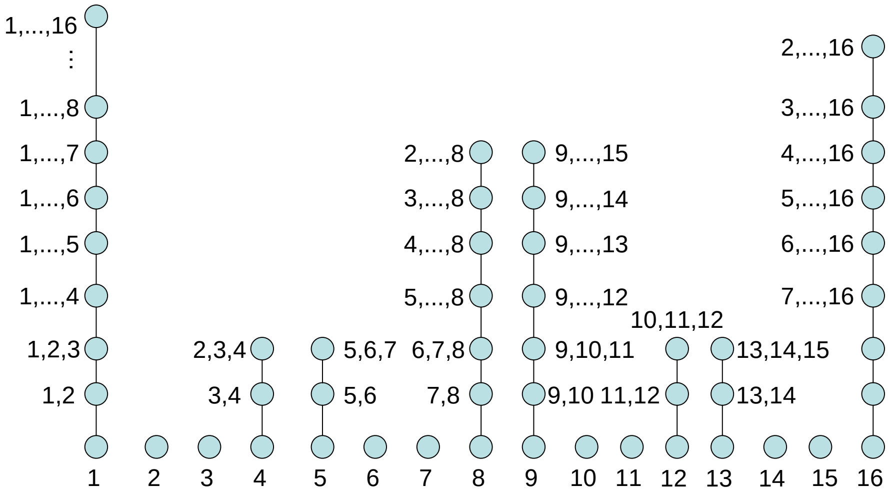


Figure 9 Chain decomposition.

Then, we have two ways to derive keys：

1. Based on PRSG：

Choosing an PRSG $G:\{0,1\}^n \rightarrow \{0,1\}^n$. For each chain, we only need to generate an independent value for root, then all nodes above it can be derived from root. For example, $K_{123} = G(K_{12}) = G^2(K_1)$. In each chain, user stores keys for subsets which he belongs to and nearest to the root. For example in the Fig.8, our paradigm with the chain decomposition in the Figure9 point out user2 needs to store the keys $K_2, K_{234}, K_{2\sim8}, K_{2\sim16}, K_{12}$ .Since one user is at most in $\log n + 1$ chains, the storage size for each user is $O(\log n)$.

2. Based on RSA-Accumulator

We construct a Maximin Matrix $A_{n\times m}$.

n = The number of users.

m = The number of chains.

Maximin Matrix Definition：

For a set system X, for all $S \in S_X$ there exist $j : 1 \le j \le m$ where $\max_{i\in S} a_{ij} < \min_{i\in N\setminus S} a_{ij}$

Consider a chain decomposition $\{G_1,...,G_M\} \in S_X$

For each chain $G_j : S_1 \rightarrow ... \rightarrow S_l$ we construct $j's$ column vector

$$a_{ij} = \begin{cases} 0 & if \quad i \in S_1 \\ w & if \quad i \in S_{w+1} \setminus S_w \\ l & otherwise \end{cases}$$

Then we choose a random number s, a big integer $N = pq$ and n distinct primes p_j .

Compute secret value $p(u)$ and assign to each user.

$$p(u) = \prod_{j=1}^{m} s^{p_j^{a_{uj}}} \bmod N$$

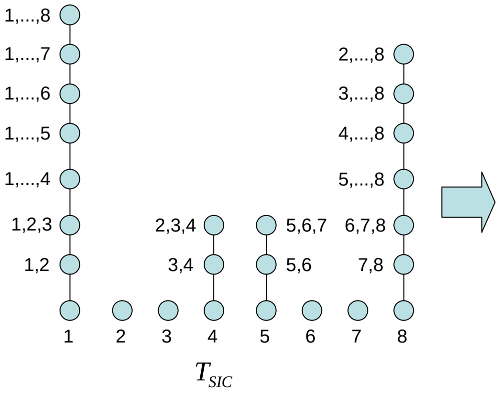
The key $k(S)$ for each set S ：

$$k(S)=\prod_{j=1}^{m} s^{p_j^{(\max_{i\in S} a_{ij})}} \bmod N$$

User can derive key $k(S)$ if $u\in S$ ：

$$k(S)=\prod_{j=1}^{m} p(u)^{p_j^{(\max_{i\in S} a_{ij}-a_{uj})}} \bmod N$$

If $u\notin S$, $(\max_{i\in S} a_{ij}-a_{uj})<0$.User does not know the factor of *N*, so he can not compute it on exponentiation. Following figure is an exhibition：



$$p(1)=s^{(p_1^0 p_2^1 p_3^1 p_4^3 p_5^3 p_6^1 p_7^1 p_8^7)}$$
$$p(8)=s^{(p_1^7 p_2^1 p_3^1 p_4^3 p_5^3 p_6^1 p_7^1 p_8^0)}$$

$$\begin{bmatrix} 0 & 1 & 1 & 3 & 3 & 1 & 1 & 7 \\ 1 & 0 & 1 & 2 & 3 & 1 & 1 & 6 \\ 2 & 1 & 0 & 1 & 3 & 1 & 1 & 5 \\ 3 & 1 & 1 & 0 & 3 & 1 & 1 & 4 \\ 4 & 1 & 1 & 3 & 0 & 1 & 1 & 3 \\ 5 & 1 & 1 & 3 & 1 & 0 & 1 & 2 \\ 6 & 1 & 1 & 3 & 2 & 1 & 0 & 1 \\ 7 & 1 & 1 & 3 & 3 & 1 & 1 & 0 \end{bmatrix}$$

Maximin matrix

Figure 10 Translate decomposition chain into Maximin matrix.

User 1 can derive subset key K_{123} by:

$$k(123)=s^{p_1^2 p_2^1 p_3^1 p_4^3 p_5^3 p_6^1 p_7^1 p_8^7} = p(1)^{p_1^2 p_2^0 p_3^0 p_4^0 p_5^0 p_6^0 p_7^0 p_8^0}$$

Security of this scheme is based on RSA Assumption. Using this scheme , everyone only needs to store $O(1)$ keys , but needs to compute Maximin matrix and find *n* primes. The header size of SIC scheme is at most *2r-1*. It is same as SD scheme.

2.2Public Key Broadcast Encryption

In Public key BE system, everyone can broadcast encrypted message without any secret. Dodis and Fazio [6] showed how to translate the SD and LSD methods to the public key setting, while having a fixed constant size public key.

The main idea is Identity Based Encryption (IBE) [8] and Hierarchical Identity Based Encryption (HIBE) [2]. IBE is a public encryption system. In this system we can input any string and generate a pair of keys. Public key can be generated by everyone and the correspondent private key for user only can be generated by private key generator (PKG) .Advantage of this technique is that we can use our e-mail address or phone number to be our public key. Such that people do not need to store any public key. It saves a lot of storage space. HIBE is an enhancement of IBE. In HIBE, everyone has a unique Hierarchical ID and each user can derive his descendant users. We can send encrypted message to any user and only receiver's ascendants can decrypt it (include receiver). For example, if an user's HID is tw.nctu.cs . He can decrypt all message which pattern is like tw.nctu.cs.* , but he can not decrypt messages like tw.nctu.ee or tw.gov …etc.

2.2.1 Public key -Complete Subtree (PK-CS) Scheme

The main idea of this method is rename all sets (nodes) with an unique ID, then we can use IDE scheme. First, we let ID(root)= R . Then , the ID of left child we concatenate 0 after parent's ID, and ID of right child we concatenate 1 after parent's ID.

In Figure11 ID(Y)=R00 , ID(b)=R001 , ID(W)=R11

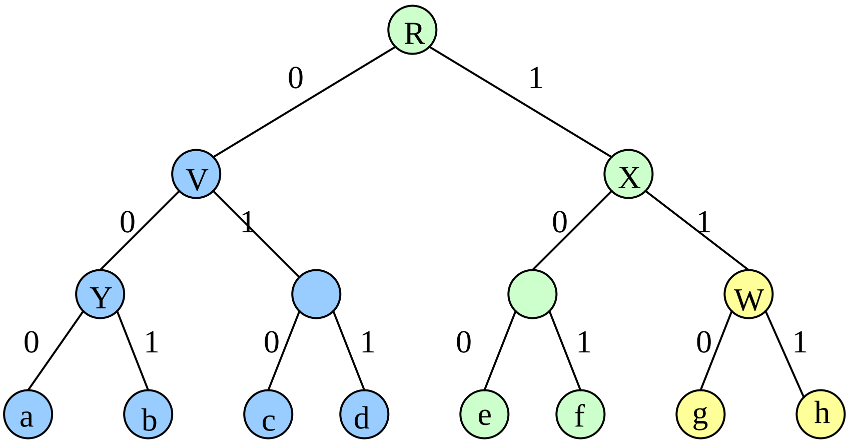


Figure 11 ID of PK-CS scheme

The size of private key for each user needs to store is the same as CS scheme. In figure10 , user b needs to store the private keys where ID=R001 , ID=R00 , ID=R0 and ID=R . The public key size is $O(1)$ .This is equal to IBE system. Header size and computation cost are equal to CS scheme. For any people want to send encrypted message, he can find set covers of users, then uses public keys of these ID .

2.2.2 Public key -Subset Different (PK-SD) Scheme

Transfer SD scheme is similar with CS scheme. The hard problem is that, in this system all values of sets are not independent. Parent's value can derive child's value. The answer to solve this question is HIBE system. We define following notations:

v_i ：Node i

$S_{i,j}$ ：Set $S_i - S_j$. It is same as SD scheme.

L_i ： Label of node i.

$L_{i,j}$ ： Label of set $S_{i,j}$. It can be derived from L_i .

$K_{i,j}^{PRI}$ ：private key of set $S_{i,j}$ $K_{i,j}^{PUB}$ ：public key of set $S_{i,j}$

ID(x),HID(x)： The object’s name of x

The ID of each node v_i is the same as PK-CS scheme. We need to give an unique HID for each $L_{i,j}$.

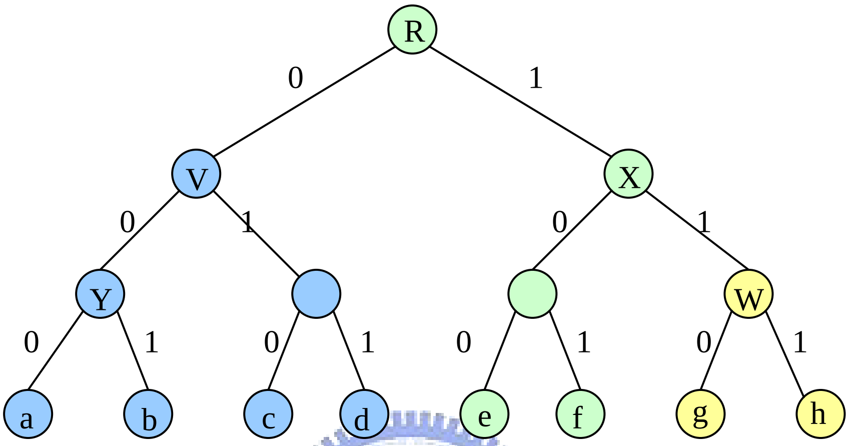


Figure 12 HID of PK-SD scheme

In above Figure ID(v_v)=R0 ID(v_b)=R001

For root , ID(root)= R

For label L_i , HID(L_i)=ID(v_i)

For label $L_{i,j}$ $i \neq j$, HID($L_{i,j}$)=(ID(L_i),[ID(v_j)\ID(v_i)])

[ID(v_j)\ID(v_i)]： The different between ID(v_j) with ID(v_i). For each different symbol. We use “,” to separate it.

Now, each label has an unique HID and same relationship in SD-scheme. For example, In above figure:

HID($L_{V,b}$)=(R0, 0, 1) HID($L_{R,e}$)=(R,1,0,0)

HID for keys $K_{i,j}$, HID($S_{i,j}$)=(HID($L_{i,j}$),2)

We can use HIBE system to derive all value properly. The number of values for each user need to store in PK-SD is equal to SD scheme. Each user stores the private keys of

HID($L_{i,j}$), where $L_{i,j}$ is the label needs to be stored in SD scheme. However, in HIBE system , private key length is linear to max level which is $O(\log n)$. By sharing some parameter, the private key size is about twice of SD scheme ($O(\log^2 n)$).By using the HIBE technique in [2]. The header size is still $O(r)$, the public key size is $O(1)$, and the computation cost is $O(\log n)$ exponentiations for key derivation.

2.3Tracing Traitor

In a BE system, message was encrypted and only the subscribers can decrypt the ciphertext. However, a traitor (malicious subscriber) may clone his decoder and sell the pirate decoder for profits. A traitor tracing scheme is a scheme with capability to find these pirate users. A traitor tracing scheme is fully *k*-resilient if it can point out all traitors where the number of traitors is less than *k*.

2.3.1 Dynamic Shares Scheme

Tzeng and Tzeng[21] proposed an efficient fully *k*-resilient Public-Key Traitor Tracing scheme by using dynamic shares. There scheme needs $O(k)$ header size, $O(1)$ private keys, $O(k)$ public keys and $O(k)$ computation cost.

System setup. Center select a large prime *q* and select a degree-*z* polynomial $z \geq 2k$ $f(x) = \sum_{t=0}^{z} a_t x^t \pmod q$ with coefficient over Z_q. The user's secret key is $f(x)$ and public keys are $\left\langle g, g^{a_0}, g^{f(1)}, \ldots, g^{f(z)} \right\rangle$

Registration. When a receiver *i*, *i*>*z* registers, the center give the receiver *i* a decoder with the share $(i, f(i))$ (private key)

We call $(j, f(j))$ an unused share if it has not been assigned to any receiver.

Encryption. The sender randomly selects unused shares $(j_1, f(j_1)), (j_2, f(j_2)), \ldots,$ $(j_z, f(j_z))$ and a random number $r \in Z_q$, and a session key s. The sender computes the enabling block

$$T = \left\langle sg^{ra_0}, g^r, (j_1, g^{rf(j_1)}), (j_1, g^{rf(j_2)}), \ldots, (j_1, g^{rf(j_z)}) \right\rangle$$

And broadcast $\langle T, E'(s,M)\rangle$ where E' is a secret-key cipher.

Decryption. When receiving $\langle T, E'(s,M)\rangle$**,** the receiver compute s by

$$sg^{ra_0} /[(g^r)^{f(i)\lambda_z} \cdot \prod_{t=0}^{z-1} (g^{rf(x_t)})^{\lambda_t}] = sg^{ra_0} / g^{r(\sum_{t=0}^{z-1} f(x_t)\lambda_t + f(i)\lambda_z)} = sg^{ra_0} / g^{ra_0} = s$$

Where $x_0 = j_1, x_1 = j_2, ..., x_{z-1} = j_z$ **and** $x_z = i$

$$\lambda_t = \prod_{0 \le j \ne t \le z} \frac{x_j}{x_j - x_t} \text{ are lagrange coefficients.}$$

He then uses s to decrypt $E'(s,M)$ to obtaion M

Traitor Tracing. There are two black box traitor tracing algorithms for following situations.

I. If pirate decoder's key is not linear combination of shares.

1. For every possible m-receiver set $\{c_1, c_2, ..., c_m\}$, $m \le k$,

   (a) Randomly select z-m unused shares $\{j_1, ..., j_{z-m}\}$ and construct a test $\langle T, E'(s,M)\rangle$ **where**

   $$T = \left\langle \begin{matrix} sg^{ra_0}, g^r, (c_1, g^{rf(c_1)}), (c_2, g^{rf(c_2)}), ..., (c_m, g^{rf(c_m)}), \\ (j_1, g^{rf(j_1)}), (j_2, g^{rf(j_2)}), ..., (j_{z-m}, g^{rf(j_{z-m})}) \end{matrix} \right\rangle$$

   (b) Feed $\langle T, E'(s,M)\rangle$ to the decoder.

   (c) If the pirate decoder does not output correct M , $\{c_1, c_2, ..., c_m\}$ is a possible traitor set.

2. Output the smallest of all possible traitor sets found in Step 1c.

II. If pirate decoder's key is linear combination of shares.

1. For every possible m-receiver set $\{c_1, c_2, ..., c_m\}$, $m \le k$,

   (a) Randomly select a degree-z polynomial $h(x) = \sum_{i=0}^{z} a_i z^i$ that passes $(c_1, f(c_1)), (c_2, f(c_2)), ..., (c_m, f(c_m))$ points.

   (b) Randomly select z unused shares $j_1, ..., j_z$ and construct a test

$\langle T, E'(s,M)\rangle$ where $T = \langle sg^{ra_0}, g^r, (j_1, g^{rh(j_1)}), (j_2, g^{rh(j_2)}), \ldots, (j_z, g^{rh(j_z)})\rangle$

(c) Feed $\langle T, E'(s,M)\rangle$ to the decoder.

(d) If the pirate decoder outputs correct M, $\{c_1, c_2, \ldots, c_m\}$ is a possible traitor set.

2. Output the smallest of all possible traitor sets found in Step 1c.

Chapter3

Background

3.1Bilinear Groups

We use bilinear maps and bilinear map groups.

1. G and G_1 are two (multiplicative) cyclic groups of prime order *q*;

2. g is a generator of G

3. $\hat{e}$ is a bilinear map $\hat{e}: G \times G \rightarrow G_1$

$\hat{e}$ has the following properties:

1.For all $u, v \in G_q$ and $x, y \in Z_q$, $\hat{e}(u^x, v^y) = \hat{e}(u, v)^{xy}$

2.Let g be a generator of G, we have $\hat{e}(g, g) = g_1 \neq 1$ is a generator of G_1

3.2 CBDH assumption

The CBDH problem is to compute $\hat{e}(g, g)^{abc}$ from give (g, g^a, g^b, g^c), where g is random generators of G_q and *a,b,c* are random over Z_q. We say that CBDH is $(t, \varepsilon) - hard$ if for any probabilistic algorithm A with time bound *t*, there is some k_0 such that for any $k \geq k_0$,

$$\Pr[A(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc} : g \xleftarrow{u} G_q \setminus \{1\}, a, b, c \xleftarrow{u} Z_q] \leq \varepsilon$$

3.3 CDH assumption

The CDH problem is to compute g^{ab} from give (g, g^a, g^b), where g is random generators of G_q and *a,b* are random over Z_q. We say that CDH is $(t, \varepsilon) - hard$ if for any probabilistic algorithm A with time bound *t*, there is some k_0 such that for any $k \geq k_0$,

$$\Pr[A(g, g^a, g^b) = g^{ab} : g \xleftarrow{u} G_q \setminus \{1\}, a, b \xleftarrow{u} Z_q] \leq \varepsilon$$

3.4Broadcast encryption

A public-key BE scheme consist of three probability polynomial-time algorithms:

-$Setup(1^z, ID, U)$. let $U = \{U_1, U_2, ..., U_N\}$. It takes as input the security parameter z, a system identity ID and a set U of users and output a public key PK and N private key sets $SK_1, SK_2, ..., SK_N$, one for each user in U.

-$Enc(PK, S, M)$. It takes as input the public key PK, a set $S \subseteq U$ of authorized users and a message M. It outputs a pair $\langle Hdr(S, m), C \rangle$ of the ciphertext header and body, where m is a randomly generated session key and C is the ciphertext of M encrypted by m via some standard symmetric encryption scheme, e.g AES.

-$Dec(PK, SK_k, Hdr(S, m), C))$.It takes as input the public key PK, the private key SK_k of user U_k , the header $Hdr(S, m)$ and the body C. If $U_k \not\in S$, it cannot decrypt C to obtain the message M. If $U_k \in S$, it can decrypt the header $Hdr(S, m)$ to obtain the session key m and then uses m to decrypt the ciphertext body C for message M.

The system is correct if all users in S can get the broadcasted message M.

Security. We describe the indistinguishability security against the adaptive chosen ciphertext attack(**IND-CCA** security) for broadcast encryption[4] as follows. Here, we focus on the security of the session key, which in turn guarantees the security of the ciphertext body C. Let $Enc*$ and $Dec*$ be like Enc and Dec except that message M and the ciphertext body C are ignored. The security is defined by an adversary A and a challenger C via the following game.

Init. The adversary A choose a system identity ID and a subset $S* \subseteq U$ of users that it wants to attack.

Setup. The challenger C runs $Setup(1^z, ID, U)$ to generate a public key PK and private key sets $SK_1, SK_2, ..., SK_N$. The challenger C gives $SK_i, U_i \not\in S*$ to A.

Query phase 1. The adversary A issues decryption queries $Q_i, 1 \le i \le n$ of form (U_k, S, Hdr), $S \subseteq S^*$, $U_k \in S$ and the challenger C responds with $Dec^*(PK, SK_k, Hdr)$, which is the session key encrypted in Hdr .

Challenge. The challenger C runs $Enc^*(PK, S^*)$ and output $Hdr^*(S^*, m)$, where m is randomly chosen. Then, C choose a random bit b and a random session key m^* and sets $m_b = m$ and $m_{1-b} = m^*$. C gives $Hdr^*((S^*, m), m_0, m_1)$ to A .

Query phase 2. The adversary A issues decryption queries $Q_i, n+1 \le i \le q_D$ of form (U_k, S, Hdr), $S \subseteq S^*$, $U_k \in S$ and the challenger C responds with $Dec^*(PK, SK_k, Hdr)$.

Guess. A outputs a guess b' for b.

In the above the adversary A is static since it choose the target user set S^* before the system setup. Let:

$$Adv_{A,\pi}^{ind-cca}(z) = 2 \cdot \Pr[A^O(PK, SK_{U \backslash S^*}, Hdr^*, m_0, m_1) = b :$$
$$S^* \subseteq U, (PK, SK_U) \leftarrow Setup(1^z, ID, U), Hdr^* \leftarrow Enc^*(PK, S^*), b \xleftarrow{u} \{0,1\}] - 1,$$

Where $SK_U = \{SK_i : 1 \le i \le N\}$ and $SK_{U \backslash S^*} = \{SK_i : U_i \notin S^*\}$.

Definition 1. A public-key BE scheme $\Pi = (Setup, Enc, Dec)$ is $(t, \varepsilon, q_D) - IND - CCA$ secure if for all t-time bounded adversary A that makes at most q_D decryption queries, we have $Adv_{A,\Pi}^{ind-cca}(z) < \varepsilon$.

Chapter4

Our Scheme

We show two public BE schemes with different performance tradeoff. First, we introduce our main idea of construction. In BE-PI scheme, we use $\lceil \log n \rceil + 1$ polynomials $f_i(x)$ on exponents with degree 2^i $(i = 0$ to $\lceil \text{logn} \rceil)$, and the secrets of $f_i(u_t)$ are shared to all users u_t. When sending an encrypted message to $U \setminus R$, we broadcast information about $f_i(R)$. Then all users in $U \setminus R$ can compute $f_i(0)$ and decrypt the message. In PK-SD-PI scheme, deployment is similar to SD scheme, all users are leaves of the tree. We use polynomials with degree 1. In this scheme not all user's shares are over the same polynomial. An polynomial $f_j^v(x)$ only pass point $(u_t, f_j^v(u_t))$ where u_t are *v*'s descendants of level *j*. Users only get secret shares of $f_j^v(w_t)$,where *v* is an ancestor of user and w_j are index of nodes on the path from user to *v* .When broadcasting to a subset $S_{i,t}$ in SD scheme, we broadcast $f_j^i(t)$ in this scheme. For each scheme ,we present two methods for implementation. The first one has smaller storage size and decryption cost. The second one can reduce header size with slight modification.

4.1The BE-PI scheme

4.1.1 BE-PI scheme

1. $Setup(1^z, ID, U)$: *z* is the security parameter, *ID* is the identity name of the system, and $U = \{U_1, U_2, \ldots, U_n\}$ is the set of users in the system. Let G_q and G_1 be the bilinear groups with the pairing function $\hat{e}$, where q is a large prime. This bilinear system as described above is of security parameter *z*. Then, the system dealer does

the following:

- Choose a cryptographically secure hash function $H:\{0,1\}^* \to G_q$.
- Choose a secure symmetric encryption scheme E with key space G_q.
- Choose a generator g of group G_q, and let $\lg = \log_g$ and $g_1 = \hat{e}(g,g)$
- Compute $g^{a_j^{(i)}} = H(ID \| "f" \| i \| j)$ for $0 \le i \le \lfloor \log_2 n \rfloor$ and $0 \le j \le 2^i$, where “f” means polynomial-related parameters.

  <u>Remark.</u> The underlined polynomials, are, $0 \le i \le \lceil \log_2 n \rceil$,

  $$f_i(x) = \sum_{j=0}^{2^i} a_j^{(i)} x^j \quad (mod\ q)$$

  The system dealer does not know the coefficients $a_j^{(i)} = \lg H(ID \| "f" \| i \| j)$

  But, this does not matter.
- Randomly choose a secret $\rho \in Z_q$ and compute g^{ρ}.
- Publish the public key $PK = (ID, H, E, G_q, G_1, \hat{e}, g, g^{\rho})$
- Assign a set $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log N \rceil}\}$ of private keys to user U_k, $1 \le k < N$, where $s_{k,i} = (g^{\rho f_i(k)})$

  ρ is the master key of the system.

2. $Enc(PK, S, M): S \subseteq U, R = U \setminus S = \{U_{i_1}, U_{i_2}, \ldots, U_{i_l}\}$ is the set of revoked users, where $l \ge 1$. M is the sent message. The broadcaster does the following:

- Let $\alpha = \lceil \log_2 l \rceil$ and $L = 2^{\alpha}$.
- Randomly select distinct $i_{l+1}, i_{l+2}, \ldots, i_L > n$. These $U_{i_t}, l+1 \le t \le L$ are dummy users.
- Randomly select a session key $m \in G_q$
- Randomly select $r \in Z_q$ and compute $1 \le t \le L$

  $$g^{rf_\alpha(i_t)} = \left(\prod_{j=0}^{L} H(ID \| "f" \| \alpha \| j)^{i_t^j}\right)^r$$

- The ciphertext header $Hdr(S,m)$ is

  $(\alpha, m\hat{e}(g^{\rho}, g^{f_{\alpha}(0)})^{r}, g^{r}, (i_1, \hat{e}(g^{\rho}, g^{f_{\alpha}(i_1)})^{r}, (i_2, \hat{e}(g^{\rho}, g^{f_{\alpha}(i_2)})^{r}), \ldots, (i_L, \hat{e}(g^{\rho}, g^{f_{\alpha}(i_L)})^{r})$

  Let $b_j = \hat{e}(g^{\rho}, g^{f_{\alpha}(i_j)})^{r} = g_1^{r\rho f(i_j)}, 1 \le j \le L$

- The ciphertext body is $C = E_m(M)$

3. $Dec(PK, SK_k, Hdr(S,m), C) : U_k \in S$. The user U_k does the following

- Compute $b_0 = \hat{e}(g^{r}, g^{\rho f(k)}) = g_1^{r\rho f_{\alpha}(k)}$

- Use the header and Lagrange interpolation method to compute

$$g_1^{r\rho f_{\alpha}(0)} = \prod_{j=0}^{L} b_j^{\lambda_j}$$

where $\lambda_j = \dfrac{(-i_0)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j - i_0)\cdots(i_j - i_{j-1})(i_j - i_{j+1})\cdots(i_j - i_L)} (\bmod q), i_0 = k$

- Compute the session key

$$\frac{m\hat{e}(g^{\rho}, g^{f_{\alpha}(0)})^{r}}{g_1^{r\rho f_{\alpha}(0)}} = m\frac{g_1^{r\rho f_{\alpha}(0)}}{g_1^{r\rho f_{\alpha}(0)}} = m \quad (2)$$

- Use *m* to decrypt the ciphertext body *C* to obtain the message *M*.

Correctness. We can easily see that the scheme is correct by Equation (2)

(This technique with single polynomial can reduce public key size to $O(1)$ of traitor tracing scheme in section 2.3 [21].)

4.1.2 Performance analysis

For each system, the public key is $(ID, H, E, G_q, G_1, \hat{e}, g^{\rho})$, which is of size $O(1)$. Since all systems can use the same $(H, E, G_q, G_1, \hat{e}, g)$, the real public key specific to a system is simply (ID, g^{ρ}). Each system dealer has a secret ρ for assigning private keys to its users. Each user U_k holds private keys $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log n \rceil}\}$, which each corresponds to a share of polynomial f_i in the masked form, $0 \le i \le \lceil \log n \rceil$. The number of private keys is $O(\log n)$. When r

users are revoked, we choose the polynomial f_α of degree 2^α for encrypting the session key, where $2^{\alpha-1} < r \le 2^\alpha$. Thus, the header size is $O(2^\alpha) = O(r)$. It is actually no more than *2r*.

Since evaluation of a hash function is much faster than computation of a pairing and a modular exponentiation, we omit the cost or evaluating hash functions. To prepare a header, the broadcaster needs to do $2^\alpha + 1$ parings. However, broadcaster can precompute $\hat{e}(g^\rho, g^{f_\alpha(0)}), \hat{e}(g^\rho, g^{f_\alpha(i_1)}), \hat{e}(g^\rho, g^{f_\alpha(i_2)}), \ldots, \hat{e}(g^\rho, g^{f_\alpha(i_L)})$ before broadcasting. Then broadcaster only need $2^\alpha + 2$ modular exponentiations, which is O(r) modular exponentiations .Or broadcaster can send

$$(\alpha, m\hat{e}(g^\rho, g^{f_\alpha(0)})^r, g^r, (i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), \ldots, (i_L, g^{rf_\alpha(i_L)})) \text{ instead of}$$

$$(\alpha, m\hat{e}(g^\rho, g^{f_\alpha(0)})^r, g^r, (i_1, \hat{e}(g^\rho, g^{f_\alpha(i_1)})^r, \hat{e}(g^\rho, g^{f_\alpha(i_2)})^r, \ldots, \hat{e}(g^\rho, g^{f_\alpha(i_L)})^r$$

Receivers can compute $\hat{e}(g^\rho, g^{f_\alpha(i)})^r = \hat{e}(g^\rho, g^{rf_\alpha(i)})$ by itself. Then the broadcaster only need to compute 1 paring and O(r) modular exponentiations. For a user in *S* to decrypt a header, the user needs to perform 1 paring functions and O(r) modular exponentiations.

4.2The BE-PI-2 scheme

This scheme is slightly different to BE-PI scheme. Each user is not over the same polynomial $F(x) = \rho f(x)$. In this scheme, every user u owns unique shares over $F_{r_u}(x) = r_u f(x)$ where r_u is a random value assigned by KDC.

4.2.1 BE-PI-2 scheme

1. $Setup(1^z, ID, U)$: z is the security parameter, *ID* is the identity name of the system, and $U = \{U_1, U_2, \ldots, U_n\}$ is the set of users in the system. Let G_q and G_1 be the bilinear groups with the pairing function $\hat{e}$, where q is a large prime. This bilinear system as described above is of security parameter *z*. Then, the system dealer does the following:

- Choose a cryptographically secure hash function $H : \{0,1\}^* \rightarrow G_q$.
- Choose a secure symmetric encryption scheme E with key space G_q.
- Choose a generator g of group G_q, and let $\lg = \log_g$ and $g_1 = \hat{e}(g,g)$
- Compute $h_i = H(ID \| "h" \| i)$ for $1 \le i \le \lceil \log_2 n \rceil$, where "h" indicates the h-related hash values.
- Compute $g^{a_j^{(i)}} = H(ID \| "f" \| i \| j)$ for $0 \le i \le \lfloor \log_2 n \rfloor$ and $0 \le j \le 2^i$, where "f" means polynomial-related parameters.

  Remark. The underlined polynomials, are, $0 \le i \le \lceil \log_2 n \rceil$,

  $$f_i(x) = \sum_{j=0}^{2^i} a_j^{(i)} x^j \quad (mod\ q)$$

  The system dealer does not know the coefficients $a_j^{(i)} = \lg H(ID \| "f" \| i \| j)$

  But, this does not matter.
- Randomly choose a secret $\rho \in Z_q$ and compute g^ρ.
- Publish the public key $PK = (ID, H, E, G_q, G_1, \hat{e}, g, g^\rho)$
- Assign a set $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log n \rceil}\}$ of private keys to user U_k, $1 \le k < n$, where $s_{k,i} = (g^{r_k}, g^{r_k f_i(k)}, g^{r_k f_i(0)} h_i^\rho)$

  h_i^ρ is the master key of the system.

  and $r_{k,i}$ is randomly chosen from $Z_q, 1 \le i \le \lceil \log n \rceil$.

2. $Enc(PK, S, M) : S \subseteq U, R = U \setminus S = \{U_{i_1}, U_{i_2}, \ldots, U_{i_l}\}$ is the set of revoked users, where $l \ge 1$. M is the sent message. The broadcaster does the following:

- Let $\alpha = \lceil \log_2 l \rceil$ and $L = 2^\alpha$.
- Compute $h_\alpha = H(ID \| "h" \| \alpha)$.
- Randomly select distinct $i_{l+1}, i_{l+2}, \ldots, i_L > n$. These $U_{i_t}, l+1 \le t \le L$ are dummy users.
- Randomly select a session key $m \in G_q$

- Randomly select $r \in Z_q$ and compute $1 \leq t \leq L$

$$g^{rf_\alpha(i_t)} = (\prod_{j=0}^{L} H(ID \| "f" \| \alpha \| j)^{i_t^j})^r$$

- The ciphertext header $Hdr(S,m)$ is

$$(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), ...(i_L, g^{rf_\alpha(i_L)}))$$

- The ciphertext body is $C = E_m(M)$

3. $Dec(PK, SK_k, Hdr(S,m), C) : U_k \in S$. The user U_k does the following

- Compute $b_0 = \hat{e}(g^r, g^{r_k f_\alpha(k)}) = g_1^{rr_k f_\alpha(k)}$
- Compute $b_j = \hat{e}(g^{r_k}, g^{rf_\alpha(i_j)}) = g_1^{rr_k f_\alpha(i_j)}$
- Use the header and Lagrange interpolation method to compute

$$g^{r\rho f_\alpha(0)} = \prod_{j=0}^{L} b_j^{\lambda_j}$$

where $\lambda_j = \dfrac{(-i_0)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j - i_0)\cdots(i_j - i_{j-1})(i_j - i_{j+1})\cdots(i_j - i_L)} (\bmod q), i_0 = k$

- Compute the session key

$$\frac{m\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{rr_k f_\alpha(0)}}{\hat{e}(g^r, g^{r_k f_\alpha(0)} h_\alpha^\rho)} = m \frac{\hat{e}(g^\rho, h_\alpha)^r \cdot g_1^{rr_k f_\alpha(0)}}{\hat{e}(g^r, h_\alpha^\rho) \cdot g_1^{rr_k f_\alpha(0)}} = m \qquad (2')$$

- Use *m* to decrypt the ciphertext body *C* to obtain the message *M*.

Correctness. We can easily see that the scheme is correct by Equation (2’)

4.2.2 Performance analysis

In this scheme, we need two more pairing computation than BE-PI scheme, and the storage size is twice as BE-PI scheme. However, we can reduce header size with following change.

4.2.3 Reduce BE-PI-2 header size

We can reduce about half header size. We assign private keys SK_k to user

$U_k, 1 \le k \le n,$

$$\begin{aligned} SK_k &= \{z_{k,0}, z_{k,1,} s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log n \rceil}\} \\ &= (g^{r_k}, g^{r_k f_1(0)} g^{r_k f_2(0)} \cdots g^{r_L f(0)} h^{\rho}, g^{r_k f_1(k)}, g^{r_k f_2(k)}, \ldots, g^{r_L f(k)}) \end{aligned}$$

When broadcasting message to $S \setminus R$, where $R = \{U_{i_1}, U_{i_2}, \ldots, U_{i_l}\}$ is the set of l revoked users, we select index of functions $c_1, c_2, \ldots c_w$ where $2^{c_1} + 2^{c_2} + \ldots + 2^{c_w} = l$ We can present l in binary string, so all c can be found easily, and $v_1, \ldots, v_t$ are remainder index of functions. Then the header $Hdr(S, m)$ is

$$(l, m\hat{e}(g^{\rho}, h)^r, g^r, g^{r(f_{v_1}(0)+f_{v_2}(0)+\ldots+f_{v_t}(0))}, (i_1, g^{rf_{c_1}(i_1)}), (i_2, g^{rf_{c_1}(i_2)}), (i_3, g^{rf_{c_2}(i_2)}), \ldots (i_L, g^{rf_{c_w}(i_L)}))$$

Decryption: Users not in R can Compute $\hat{e}(g^{\rho}, h)^r$ by

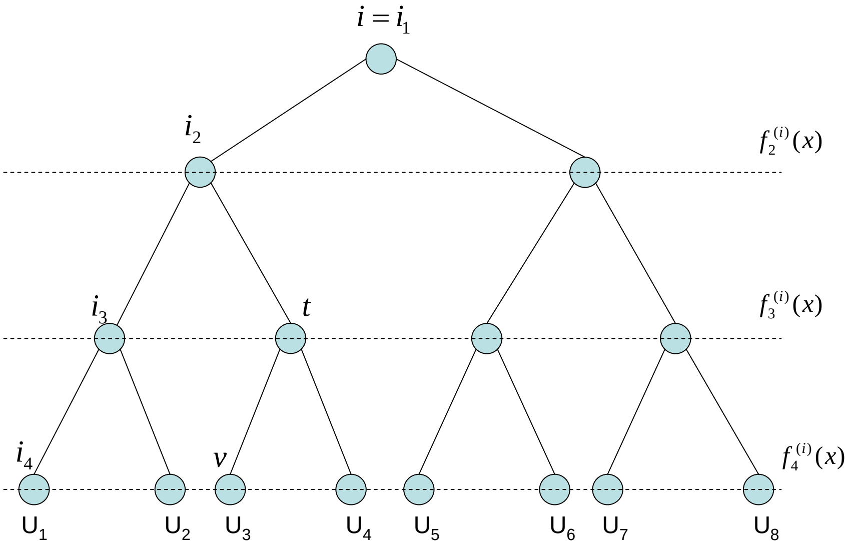
$$\begin{aligned} &\frac{\hat{e}(g^r, z_{k,1})}{\hat{e}(z_{k,0}, g^{r(f_{v_1}(0)+f_{v_2}(0)+\ldots+f_{v_t}(0)})\hat{e}(z_{k,0}, g^{rf_{c_1}(0)})\hat{e}(z_{k,0}, g^{rf_{c_2}(0)})\cdots\hat{e}(z_{k,0}, g^{rf_{c_{kw}}(0)})} \\ &= \frac{\hat{e}(g^r, g^{r_k(f_1(0)+f_2(0)+\ldots+f_L(0))} h^{\rho})}{\hat{e}(g^{r_k}, g^{r(f_{v_1}(0)+f_{v_2}(0)+\ldots+f_{v_t}(0)})\hat{e}(g^{r_k}, g^{rf_{c_1}(0)})\hat{e}(g^{r_k}, g^{rf_{c_2}(0)})\cdots\hat{e}(g^{r_k}, g^{rf_{c_{kw}}(0)})} \\ &= \frac{\hat{e}(g^r, g^{r_k(f_{v_1}(0)+f_{v_2}(0)+\ldots+f_{v_t}(0)+f_{c_1}(0)+f_{r_c}(0)+\ldots+f_{r_{cw}}(0))})\hat{e}(g^r, h^{\rho})}{\hat{e}(g^r, g^{r_k(f_{v_1}(0)+f_{v_2}(0)+\ldots+f_{v_t}(0)+f_{c_1}(0)+f_{r_c}(0)+\ldots+f_{r_{cw}}(0))})} = \hat{e}(g^{\rho}, h)^r \end{aligned}$$

If $k \notin R$, all $\hat{e}(z_{k,0}, g^{rf_{c_d}(0)})$ $, 1 \le d \le t$ can be computed from shares $(i_1, g^{rf_{c_d}(i_1)}), (i_2, g^{rf_{c_d}(i_2)}), (i_3, g^{rf_{c_d}(i_2)}), \ldots$ and private keys $g^{r_k}, g^{r_k f_c(k)}$.)

Then user computes m: $m\hat{e}(g^{\rho}, h)^r / \hat{e}(g^{\rho}, h)^r = m$

4.3The PK-SD-PI scheme

4.3.1 The PK-SD-PI scheme



-U_1 holds $f_2^{(i)}(i_2), f_3^{(i)}(i_3), f_4^{(i)}(i_4)$

-U_3 holds $f_2^{(i)}(i_2), f_3^{(i)}(t), f_4^{(i)}(v)$

-For subset $S_{i,t}$, $f_3^{(i)}(t)$ is broadcasted so that U_3 and U_4 cannot decrypt, but others can.

Figure 13: Level polynomials, private keys and broadcasted shares for subtree T_i

We now present our PK-SD-PI scheme, which is constructed by using the polynomial interpolation technique on the collection of subsets in SD scheme.The system setup is similar to that of the BE-PI scheme. Consider a complete binary tree *T* of $\lceil \log n \rceil + 1$ levels. The nodes in *T* are numbered differently. Each user in U is associated with a different leaf node in *T*. We call a complete subtree rooted at node *i* as "subtree T_i". For each subtree T_i of η levels (Level 1 to level η from top to bottom), we define the degree-1 polynomials $f_j^{(i)}(x) = a_{j,1}^{(i)} x + a_{j,0}^{(i)} (\bmod q)$

Where $a_{j,0}^{(i)} = \lg H(ID \| "sd" \| i \| j \| 0)$ and $a_{j,1}^{(i)} = \lg H(ID \| "sd" \| i \| j \| 1), 2 \le j \le \eta$

For a user U_k in subtree T_i of η levels, he is given the private keys

$$s_{k,i,j} = g^{\rho f_j^{(i)}(i_j)}$$

For $2 \le j \le \eta$, where nodes $i_1, i_2, ..., i_\eta$ are the nodes in the path from node *i* to the leaf node for U_k (including both ends). We can read $s_{k,i,j}$ as the private key of U_k

(including both ends). We can read $s_{k,i,j}$ as the private key of U_k for the *j*th level of subtree T_i. In Figure 1, the private keys (in the unmasked form) of U_1 and U_3 for subtree T_i with $\eta = 4$ are given.

Recall that in the SD scheme, the collection *C* of subset is

$\{S_{i,t} : \text{node } i \text{ is a parent of node } t, i \neq t\}$,

Where $S_{i,t}$ denotes the set of users in subtree T_i, but not in subtree T_t. By our design, if the header contains a masked share for $f_j^{(i)}(t)$, where node *t* is in the *j*-th level of subtree T_i, only user U_k in $S_{i,t}$ can decrypt the header by using his private key $s_{k,i,j}$, that is, the masked form of $f_j^{(i)}(s)$, for some $s \neq t$, In Figure 13, the share $f_3^{(i)}(t)$ is broadcasted so that only the user in $S_{i,t}$ can decrypt the header.

For a set *R* of revoked users, let $S = S_{i_{1,},t_1}, S_{i_{12},t_2}, \ldots, S_{i_{1z},t_z}$ be a subset cover for *U\R,* the header is like

$$(S, g^r, (me(g^{\rho}, g^{f_{j1}^{(i_1)}(0)})^r, e(g^{\rho}, g^{f_{j1}^{(i_1)}(t_1)})^r), (me(g^{\rho}, g^{f_{j2}^{(i_2)}(0)})^r, e(g^{\rho}, g^{f_{j2}^{(i_2)}(t_2)})^r), \ldots$$
$$, (me(g^{\rho}, g^{f_{j_z}^{(i_z)}(0)})^r, e(g^{\rho}, g^{f_{j_z}^{(i_z)}(t_z)})^r)),$$

where node t_k is in the $j_k - th$ level of subtree T_{i_k}, $1 \leq k \leq z$.

For decryption, a non-revoked user fins an appropriate subset S_{i_j,t_j} in the header and applies the Lagrange interpolation to compute the session key *m*.

4.3.2 Performance analysis

The public key is *O (1),* which is the same as that of the BE-PI scheme. Each user belongs to at most $\lceil \log n \rceil + 1$ subtrees and each subtree has at most $\lceil \log n \rceil + 1$ levels. For the subtree of η levels, the user in the subtree hold η -1 private keys. Thus, the total number of shares (private keys) held by each user is $\sum_{i=1}^{\lceil \log n \rceil} i = (\lceil \log n \rceil^2 + \lceil \log n \rceil)/2$, which is $O(\log^2 n)$. According to [12], the number z of subsets in a subset cover is at most $2|R|-1$, which is O(r).

When the header streams in, a non-revoked user U_k needs to find his containing subset S_{i_j,t_j}, $U_k \in S_{i_j,t_j}$. With a proper numbering of the nodes in *T*, this can be done in $O(\log\log n)$ time. Without considering the time of scanning the header to find his containing subset, each user needs to perform 2 modular exponentiations. Thus, the decryption cost is *O(* 1 *)*.

4.3.3 PK-SD-PI-2 Scheme

We can also construct system based on BE-PI-2 scheme.

For a user U_k in subtree T_i of η levels, he is given the private keys

$$s_{k,i,j} = (g^{r_k}, g^{r_k f_j^{(i)}(i_j)}, g^{r_k f_j^{(i)}(0)} h_{(i,j)}^{\rho})$$

For a set *R* of revoked users, let $S = S_{i_1,t_1}, S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a subset cover for $U \setminus R$.

where node t_k is in the $j_k - th$ level of subtree T_{i_k}, $1 \le k \le z$

The header is like:

$$(S, g^r, (e(g^\rho, h_{(i_1,j_1)})^r, g^{rf_{j_1}^{(i_1)}(t_1)}), \ldots (e(g^\rho, h_{(i_z,j_z)})^r, g^{rf_{j_{z1}}^{(i_z)}(t_z)})$$

For decryption, a non-revoked user fins an appropriate subset S_{i_j,t_j} in the header and applies the Lagrange interpolation to compute the session key *m*.

4.3.4 Decrease header size and encryption cost.

In PK-SD-PI scheme, the sender needs to compute $O(r)$ parings and the header include two components $(e(g^\rho, h_{(i,j)})^r, g^{rf_j^{(i)}(t)})$ for each subset $S_{i,t}$. Now, we reduce pairing computation to $O(1)$ and decrease half header size. We use same master key h^ρ in all polynomials. $h = H(ID \| "h")$

For a user U_k in subtree T_i of η levels, he is given the private keys

$$s_{k,i,j} = (g^{r_k}, g^{r_k f_j^{(i)}(i_j)}, g^{r_k f_j^{(i)}(0)} h^{\rho})$$

All shares have same master key h^ρ here.

For a set *R* of revoked users, let $S = S_{i_1,t_1}, S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a subset cover for $U \setminus R$.

where node t_k is in the $j_k - th$ level of subtree T_{i_k}, $1 \le k \le z$

The header is like:

$$(S, g^{r}, me(g^{\rho}, h^{r}), g^{rf_{j_1}^{(i_1)}(t_1)}, ..., g^{rf_{j_z}^{(i_z)}(t_z)})$$

For decryption, a non-revoked user finds an appropriate subset S_{i_j, t_j} in the header and applies the Lagrange interpolation to compute the session key *m*.

We note that the private key size is still $O(\log n)$ and paring computing is reduced to 1 time.

4.4The PK-LSD-PI scheme

We can construct the PK-LSD-PI scheme in the simialar way. The numbers of public and private keys are $O(1)$ and $O(\log^{1+\varepsilon} n)$, respectively , for any constant $0 < \varepsilon < 1$. The header size $O(r/\varepsilon)$, which is $O(r)$ for constant ε. The decryption cost is again $O(1)$

Chapter5

Security Analysis

5.1The BE-PI scheme

We show that BE-PI scheme is fully collusion-resistant. No matter how many revoked users collude, they cannot compute the session key *m*. We show that it is one way secure (without decryption queries). The definition of one-wayness security is similar to the indistinguishability security except that the adversary, who controls the set $U \setminus S^*$ of revoked users, is required to compute the session key m from the challenge $Hdr^*(S^*, m)$, where S^* is chosen by the adversary in advance. Later, we shall how to achieve the IND-CCA security. Let q_H be the number of queries to hash function *H* by the collusion of the revoked users.

Theorem 1. *Assume that the CBDH problem is* $(t_1, \varepsilon_1) - hard$. *For any* $0 \le \alpha \le \lceil \log_2 n \rceil$, *if the number of revoked user is no more than* $L = 2^{\alpha}$, *any collusion of them cannot decrypt the header to obtain the session key with probability* $\varepsilon = \varepsilon'$, *time bound* $t = t_1 - t'$ *and* q_H *hash oracles under the random oracle model, where* t' *is polynomially bounded and* $q_H \le t$.

Proof. We reduce the CBDH problem to the problem of computing the session key from the header by the revoked users. Since the polynomials $f_i = \sum_{j=0}^{L} a_{j=0}^{(i)} x^j$ and secret shares of users for the polynomials are independent for different *i*'s. We simply discuss security for a particular α . For notation simplicity in the proof, we drop the super index (α) from $a_i^{(\alpha)}$. Without loss of generality , let $R = \{U_{1,} U_2, ..., U_L\}$ be the set of revoked users and $S^* = U \setminus R$. Note that S^* was chosen by the adversary in advance. Let the input of the CBDH problem be (g, g^a, g^b, g^c), where the paring function is implicitly known. We set the parameters of decrypting header as follows:

1. Randomly select $\eta_1, \eta_2, \eta_3, w_1, w_2, ..., w_L \in Z_q$.
2. Set the public key of the system:
   - i. Let the input g be the generator g in the system.
   - ii. Set $f_\alpha(i) = w_i, 1 \le i \le L$.
   - iii. Let $g^{a_0} = g^{f_\alpha(0)} = g^{a+\eta_1}$.
   - iv. Compute $g^{a_i}, 1 \le i \le L$, from g^{a_0} and $g^{f(j)} = g^{w_j}, 1 \le j \le L$. This can be done by the Lagrange interpolation method over exponents.
   - v. Set $g^\rho = g^{b+\eta_2}$
3. Set the secret key $g^{\rho f_\alpha(i)}$ of the revoked user $U_i, 1 \le i \le L$,as follows:
   - i. Compute $g^{\rho f_\alpha(i)} = (g^b)^{w_i}$
4. Set the header $(\alpha, m\hat{e}(g^\rho, g^{f_\alpha(0)})^r, g^r, (i_1, \hat{e}(g^\rho, g^{f_\alpha(1)})^r, \hat{e}(g^\rho, g^{f_\alpha(2)})^r, ..., \hat{e}(g^\rho, g^{f_\alpha(L)})^r$ as follows:
   - i. Let $g^r = g^{c+\eta_3}$
   - ii. Compute $\hat{e}(g^\rho, g^{f_\alpha(i)})^r = \hat{e}(g^\rho, g^r)^{f_\alpha(i)} = \hat{e}(g^{b+\eta_2}, g^{c+\eta_3})^{w_i}, 1 \le i \le L$
   - iii. Randomly select $y \in G_1$ and set $me(g^\rho, g^{f_\alpha(0)})^r = y$. We do not know what m is. But, this does not matter.

Assume that the revoked users together can compute the session key *m*. During computation the users can query hash oracles H(.). If the query is of the right form $H(ID \| "f" \| \alpha \| j)$, we set them to be g^{a_j}. If the query has ever been asked, we return the stored hash value for the query. For other non-queried inputs, we return random values in G_q.

We should check whether the distributions of the parameters in our reduction and those in the system are equal. Since $\eta_1, w_1, w_2, ..., w_L$ are randomly chosen, $g^{a_i}, 0 \le i \le L$ are distributed uniformly over G_q^{L+1}, which is again the same as that of corresponding system parameters. The distributions of g^r in the header and g^ρ in the public key are both uniform over G_q. They are the same as the distributions of the system setting. Even thought we don't known about *m*. We can check that they are all computed correctly. So, the reduction preserve the right distribution.

If the revoked users compute *m* from the header with probability ε, we can solve the CBDH problem with probability $\varepsilon_1 = \varepsilon$ by computing the following:

$$y \cdot m^{-1} \cdot e(g,g)^{-\upsilon} = e(g^{a+\eta_1}, g^{b+\eta_2})^{c+\eta_3} \cdot e(g,g)^{-\upsilon} = e(g,g)^{(a+\eta_1)(b+\eta_2)(c+\eta_3)} \cdot e(g,g)^{-\upsilon}$$
$$= e(g,g)^{abc+\upsilon} \cdot e(g,g)^{-\upsilon} = e(g,g)^{abc} \qquad (3)$$

Where $\upsilon = ab\eta_3 + (a\eta_2 + b\eta_1 + \eta_1\eta_2)(c+\eta_3)$

Since η_1, η_2, η_3 are known and $e(g,g)^{ab} = e(g^a, g^b)$, we can compute $e(g,g)^{\upsilon}$ and get $e(g,g)^{abc}$ easily.

Let t' be the time for this reduction and the solution computation in Equation (3). We can see that t' is polynomial bounded. Thus, if the collusion attack of the revoked users takes $t_1 - t'$ time, we can solve CBDH problem within time t_1.

Since each query takes a constant time, q_H cannot exceed runtime *t*. This complete the proof.

5.2The PK-SD-PI scheme

The proof of PK-SD-PI scheme is similart to BE-PI scheme. In PK-SD-PI scheme all polynomial $f_j^{(i)}(x)$ are degree one. Let the CBDH problem input values (g, g^a, g^b, g^c) .Let $g^\rho = g^a$ For $R = \{U_{1,} U_2, ..., U_t\}$, consider secret shares over polynomial $f_j^{(i)}(x)$ assigned to more than one revoked users $U_u \in R$.

1. We choose random number $w_{i,j,0}, w_{i,j,1}$ and let $f_j^{(i)}(x) = w_{i,j,1}x + w_{i,j,0}$.

2. All polynomials are degree 1. We can compute all values of $f_j^{(i)}(x)$.

3. Set $g^{\rho} = g^{b+\eta_2}$. It is equal to above setting.

4. Assign secret shares $g^{\rho f_j^{(i)}(U_{u_1})} = (g^a)^{f_j^{(i)}(U_{u_1})}, g^{\rho f_j^{(i)}(U_{u_2})}, ...$ to $U_u \in R$.

5. Let $H(ID \| "f" \| i \| j \| 0) = g^{w_{i,j,0}} = g^{a_{i,j,0}}$

$$H(ID \| "f" \| i \| j \| 1) = g^{w_{i,j,1}} = g^{a_{i,j,1}}.$$

For polynomial $f_{j'}^{(i')}(x)$ which secret shares are assigned to less than two revoked user. The setting is similar to 5.1 . We choose random numbers $\eta_{i',j',1}, w_{i',j',u}$ and set

$f_{j'}^{(i')}(0) = a + \eta_{i,j,0}$ (a is unknown value)

$f_{j'}^{(i')}(u) = w_{i',j',u}$, $H(ID \| "f" \| i' \| j' \| 0) = g^{a+\eta_{i',j',1}}$, $H(ID \| "f" \| i \| j \| 1) = g^{a_{i,j,1}}$.

Where $g^{a_{i,j,1}}$ is computed from $g^{f_{j'}^{(i')}(0)}, g^{f_{j'}^{(i')}(u)}$.When we send the challenge message to set S\R . For each subset $S_{i,t}$, S_i only appear one time and revoked users under subtree S_i were all contained in subtree S_t. Such that all users in R has only one share $f_j^{(i)}(t)$ over function $f_j^{(i)}(x)$. By the proof of 5.1, if adversary can compute m from any header of subset $S_{i,t}$. We can solve the CBDH problem.

5.3The BE-PI-2 scheme

The proof of BE-PI-2 scheme is also based on CBDH problem. Let the input of the CBDH problem be (g, g^a, g^b, g^c), where the paring function is implicitly known. We set the parameters of decrypting header as follows:

1. Randomly select $\tau, \kappa, \mu_1, \mu_2, ..., \mu_L, w_1, w_2, ..., w_L \in Z_q$.

2. Set the public key of the system:

i. Let the input g be the generator g in the system.

ii. Set $f_\alpha(i) = w_i, 1 \le i \le L$.

iii. Let $g^{a_0} = g^{f_\alpha(0)} = g^{a+\tau}$.

iv. Compute $g^{a_i}, 1 \le i \le L$, from g^{a_0} and $g^{f(j)} = g^{w_j}, 1 \le j \le L$. This can be done by the Lagrange interpolation method over exponents.

v. Set $h_\alpha = g^b \cdot g^\kappa = g^{b+\kappa}$

vi. Set $g^\rho = g^a$

3. Set the secret key $(g^{r_i}, g^{r_i f_\alpha(i)}, g^{r_i f_\alpha(0)} h_\alpha^a)$ of the revoked user $U_i, 1 \le i \le L$,as follows:

i. Let $g^{r_i} = g^{-b} \cdot g^{\mu_i} = g^{-b+\mu_i}$

ii. Compute $g^{r_i f_\alpha(i)} = (g^{r_i})^{w_i}, 1 \le i \le L$

iii. Compute $g^{r_i f_\alpha(0)} h_\alpha^\rho = g^{(-b+\mu_i)(a+\kappa)} (g^{b+\kappa})^a = (g^b)^{-\kappa} \cdot g^{a(\mu+\kappa)} \cdot g^{\mu\kappa}$

4. Set the header $(\alpha, g^r, m\hat{e}(g^\rho, h_\alpha)^r, (1, g^{rf_\alpha(1)}), (2, g^{rf_\alpha(2)}), \ldots, (L, g^{rf_\alpha(L)}))$ as follows:

i. Let $g^r = g^c$

ii. Compute $g^{rf_\alpha(i)} = (g^c)^{w_i}, 1 \le i \le L$

iii. Randomly select $y \in G_1$ and set $m\hat{e}(g^\rho, h_\alpha)^r = y$. We do not know what m is. But, this does not matter.

If the revoked user compute m from the header with probability ε_2, we can solve the CBDH problem with probability $\varepsilon_2' = \varepsilon_2$ by the computing the following:

$y \cdot m^{-1} = \hat{e}(g, g)^{abc}$

5.4The PK-SD-PI-2 scheme

The proof of PK-SD-PI-2 scheme is similar to BE-PI-2 scheme. In PK-SD-PI-2 scheme all polynomial $f_j^{(i)}(x)$ are degree one. Let the CBDH problem input values (g, g^a, g^b, g^c) .we let $g^\rho = g^a$. For $R = \{U_{1,} U_2, \ldots, U_t\}$, consider secret shares over polynomial $f_j^{(i)}(x)$ assigned to more than one revoked users $U_u \in R$. We assign secret keys with following steps:

1. Let $g^\rho = g^a$
2. We choose random number $w_{i,j,0}, w_{i,j,1}$ and let $f_j^{(i)}(x) = w_{i,j,1} x + w_{i,j,0}$.
3. All polynomials are degree 1. We can compute all values of $f_j^{(i)}(x)$.

4. For each user, set $g^{r_i} = g^{-b} \cdot g^{\mu_i} = g^{-b+\mu_i}$. It is equal to above setting.
5. Choose random value υ and set $h_{(i,j)} = g^{\nu}$,then we can compute master key $h^{\rho}_{(i,j)} = (g^{\nu})^{\rho} = (g^{\rho})^{\upsilon}$
6. Assign private keys $g^{r_k}, g^{r_k f_j^{(i)}(0)} h_{(i,j)}, g^{r_k f_j^{(i)}(k)}$ to $U_u \in R$
7. Let $H(ID \| "f" \| i \| j \| 0) = g^{w_{i,j,0}} = g^{a_{i,j,0}}$

   $H(ID \| "f" \| i \| j \| 1) = g^{w_{i,j,1}} = g^{a_{i,j,1}}$.

   $H(ID \| "h" \| i \| j) = g^{\upsilon}$

For polynomial $f_{j'}^{(i')}(x)$ which secret shares are assigned to less than two revoked user. The setting is similar to 5.4 . When we send the challenge message to set S\R . For each subset $S_{i,t}$, S_i only appear one time and revoked users under subtree S_i were all contained in subtree S_t . Such that all users in R has only one share $f_j^{(i)}(t)$ over function $f_j^{(i)}(x)$. By the proof of 5.3, if adversary can compute *m* from any header of subset $S_{i,t}$. We can solve the CBDH problem.

5.5The BE-PI-2 and PK-SD-PI-2 scheme with reducing header size

Both schemes mask the same master key h^{ρ} in all polynomials. Since all polynomials are independent, and all polynomials are masked with different random value r_k for each user. The only relation between users and polynomials is master key. We claim that even if all users are collusion, they can not compute any $g^{r_t f(0)}$ $1 \le t \le k$ and get master key h^{ρ} from $g^{r_t f(0)} h^{\rho}$. We express this problem as follow:

Given $g^{r_1}, g^{r_2}, \ldots, g^{r_k}, g^{f(x)}, g^{r_1 f(1)}, g^{r_2 f(2)}, \ldots, g^{r_k f(k)}$ where $r_1, r_2, \ldots r_k$ are randomly choosed in Z_q and $f(x)$ is a polynomial with degree z. $1 \le z < k$.

Compute any one of the $g^{r_1 f(0)}, g^{r_2 f(0)}, \ldots, g^{r_k f(0)}$

We proof it base on CDH problem:

Proof On input (g, g^a, g^b)

1. Let $g^{r_t}=g^a$, *t* is randomly choose in [1 , k].
2. Randomly choose $u_0,u_1...,u_{z-1}\in Z_q$.

   Let $g^{f(0)}$ = g^b and set $f(t)=u_0$, $f(t_1)=u_1,...,f(t_{z-1})=u_{z-1}$, where $t\neq t_1\neq...\neq t_{z-1}$.

   Then all $g^{f(x)}$ can be computed from $g^{f(0)}$ and $g^{f(t)},g^{f(t_1)},...,g^{f(t_{z-1})}$.
3. Choose *k*-1 random values $w_1,w_2,...w_{t-1},w_{t+1},...,w_k\in Z_q$ and set

   $r_1=w_1,r_2=w_2,...r_{t-1}=w_{t-1},r_{t+1}=w_{t+1},...,r_k=w_k$
4. Compute all $g^{r_1 f(1)},g^{r_2 f(2)},...,g^{r_k f(k)}$ by

   $(g^{f(1)})^{w_1},(g^{f(2)})^{w_2},...(g^{f(t-1)})^{w_{t-1}},(g^{f(t)})^{r_t}=(g^a)^{u_0},(g^{f(t+1)})^{w_{t+1}},...,(g^{f(k)})^{w_k}$
5. Input all $g^{r_1},g^{r_2},...,g^{r_k},g^{f(x)},g^{r_1 f(1)},g^{r_2 f(2)},...,g^{r_k f(k)}$ to *A* and output what *A* output.
6. *A* output value $g^{r_v f(0)}$. With probability $1/k$, $r_v=r_t$ and we get value

   $g^{ab}=g^{r_t f(0)}$

If adversary *A* can solve this problem with ε_3 probability , we can solve CDH problem with ε_3/k probability.

5.6The BE-PI scheme with IND-CCA security

In above, we show that the header is secure against any collusion of revoked users. There are some standard techniques that transfer one-wayness security to indistinguishabiltiy security against the adaptive chosen ciphertext attack. Here we present such a scheme $\prod'$ based on the technique in [9]. The modification is as follows.

- In the **Setup** algorithm, the system dealer selects another symmetric encryption scheme $\Gamma:K\times G_q\rightarrow G_q$, where K is the key space. The symmetric encryption Γ is Find-Guess(FG) secure, which is the counterpart of the IND-security for asymmetric encryption. The system dealer also chooses two additional hash functions $H_1 : \{0,1\}^* \rightarrow Z_q$ and $H_2 : G_q \rightarrow K$. The system dealer incorporates Γ, H_1 and H_2 into the public key PK.

- In the **Enc** algorithm,

  $Hdr(S, m) =$
  $(\alpha, \sigma\, \hat{e}(g^{\rho}, g^{f_\alpha(0)})^r, g^r, \Gamma_{H_2(\sigma)}(m), (i_1, \hat{e}(g^{\rho}, g^{f_\alpha(i_1)})^r, ..., (i_L, \hat{e}(g^{\rho}, g^{f_\alpha(i_L)})^r)$

  where σ is randomly chosen from G_q and $r = H_1(\sigma \| m)$.

- In the **Dec** algorithm, we first compute $\overline{\sigma}$ as described in the BE-PI scheme. Then we compute the session key $\overline{m}$ from $\Gamma_{H_2(\sigma)}(m)$ by using $\overline{\sigma}$. We check whether $\sigma\, \hat{\mathrm{e}}(\mathrm{g}^{\rho}, g^{f_\alpha(0)})^r = \sigma\, \hat{\mathrm{e}}(\mathrm{g}^{\rho}, g^{f_\alpha(0)})^{H_1(\overline{\sigma}\|\overline{m})})$. If they are equal, $\overline{m}$ is outputted. Otherwise, $\perp$ is outputted.

Before applying the result of Theorem 12 in [9], we need to show that (m, Hdr) of $\prod$ is $\gamma - uniform$. This is easy to check since for any PK and $(m, y) \in G_1^2$, $\Pr[Hdr(S, m) = y] = 1/q \cong 2^{-z}$, where *z* is the security parameter. Thus, the encryption part *Hdr* for the session key *m* is 2^{-z} uniform.

Let q_{H_1}, q_{H_2} and q_D be the numbers of queries to H_1, H_2 and the decryption oracle, respectively Recall that *t'* and q_H are described in Theorem 1.

Theorem2. *Assume that the CBDH problem is* $(t_1, \varepsilon_1) - hard$ *and the symmetric encryption* Γ *is* (t_2, ε_2) *FG-secure. The scheme* $\prod'$ *is* $(t, \varepsilon, q_H, q_{H_1}, q_{H_2}, q_D)$ *IND-CCA secure under the random oracle model, where*

$$t = \min\{t_1 - t', t_2\} - O(2z(q_{H_1} + q_{H_2}))and$$
$$\varepsilon = (1 + 2(q_{H_1} + q_{H_2})\varepsilon_1 + \varepsilon_2)(1 - 2\varepsilon_1 - 2\varepsilon_2 - 2^{-z+1})^{-q_D} - 1$$

All other schemes can be translated in the similar way.

Chapter6

Conclusion

We have presented two very efficient public-key BE schemes. One has low public and private keys. The other has a constant decryption time. BE-PI scheme with single polynomial can construct a public traitor tracing scheme [21] with O(1) public key.

We are interested in BE scheme that reducing the ciphertext size while keeping other complexities low or having traitor tracing ability in the future.

Bibliography

[1] A. Fiat and M.Naor. Broadcast Encryption, In *Proceedings of Advances in Cryptology – Crypto 93*, Lecture Notes in Computer Science 773, pp.480-491, Springer, 1994.

[2] D.Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proceedings of Advances in Cryptology – Eurocrypt 05,* Lecture Notes in Computer Science 3494, pp.440-456, Springer, 2005.

[3] D.Boneh, M. Frankling. An efficient public key traitor tracing scheme. In *Proceedings of Advances in Cryptology – Crypto 99*, Lecture Notes in Computer Science 1666, pp.338-353, Springer, 1999.

[4] D.Boneh, C.Gentry, B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of Advances in Cryptology – Crypto 05,* Lecture Notes in Computer Science 3621, pp.258-275, Springer, 2005.

[5] D.Boneh, B.Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceeding of the ACM Conference on Computer and Communications Security – CCS 06*, pp.211-220, ACM Press, 2006.

[6] Y. Dodis, N. Fazio. Public key broadcast encryption for stateless receivers. In *Proceedings of Digital Right Management 02 – DRM 02*, Lecture Notes in Computer Science 2696, pp.61-80, Springer, 2002.

[7] Y.Dodis, N.Fazio. Public key broadcast encryption secure against adaptive chosen ciphertext attack. In *Proceedings of Public Key Cryptography – PKC 03,* Lecture Notes in Computer Science 2567, pp.100-115, Springer, 2003.

[8] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In

Proceedings of Advances in Cryptology – Crypto 01, Lecture Notes in Computer Science 2139, pp.213-229, Springer, 2001.

[9] E. Fujisaki, T.Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of Advances in Cryptology – Crypto 99,* Lecture Notes in Computer Science 1666, pp.537-554, Springer, 1999.

[10] C.Gentry, A. Silverberg. Hierarchical ID-based cryptography. In *Proceedings of Advances in Cryptology – Asiacrypt 02,* Lecture Notes in Computer Science 2501, pp.548-566, Springer,2002.

[11] M.T.Goodrich, J.Z. Sun, R.Tamassia. Efficient Tree-Based Revocation in Groups of Low-State Device. In *Proceedings of Advances in Cryptology – Crypto 04,* Lecture Notes in Computer Science 3152, pp.511-527, Springer, 2004.

[12] D.Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of Advances in Cryptology – Crypto 01,* Lecture Notes in Computer Science 2139, pp.41-62, Springer, 2001.

[13] D. Halevey, A. Shamir. The LSD broadcast encryption scheme. In *Proceedings of Advances in Cryptology – Crypto 02,* Lecture Notes in Computer Science 2442, pp.47-60, Springer, 2002.

[14] N. Jho, J.Y. Hwang, J.H. Cheon, M.H. Kim, D.H. Lee, E.S. Yoo, One-Way Chain Based Broadcast Encryption Schemes, In *Proceedings of Advances in Cryptology – Eurocrypt 05,* Lecture Notes in Computer Science 3494, pp.559-574, Springer,2005.

[15] N.Attrapadung, H. Imai, Graph-Decomposition-Based Frameworks for Subset-Cover Broadcast Encryption and Efficient Instantions. In *Proceedings of Advances in Cryptology – Asiacrypt 05,* Lecture Notes in Computer Science 3788, pp.100-120, Springer,2005.

[16] T.Asano, A Revocation Scheme with Minimal Storage at Receivers, In *Proceedings of Advances in Cryptology – Asiacrypt 02,* Lecture Notes in Computer Science 2501, pp.433-450, Springer,2002.

[17] K.Kurosawa, Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proceedings of Advances in Cryptology – Eurocrypt 98*, Lecture Notes in Computer Science 1403, pp.145-157, Springer, 1998.

[18] K. Kurosawa, T. Yoshida. Linear code implies public-key traitor tracing. In *Proceedings of Public Key Cryptography*, Lecture Notes in Computer Science 2274, pp.172-187, Springer, 2002.

[19] M. Naor, B. Pinkas. Efficient trace and revoke schemes. In *Proceedings of Financial Cryptography 00*, Lecture Notes in Computer Science 1962, pp.1-20, Springer, 2000.

[20] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), pp.612-613, 1979.

[21] W.-G. Tzeng, Z.-J. Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In *Proceedings of Public Key Cryptography – PKC 01*, Lecture Notes in Computer Science 1992, pp.207-224, Springer, 2001.

[22] E.S Yoo, N.-S. Jho, J.J. Cheon, M.-H. Kim. Efficient broadcast encryption using multiple interpolation methods. In *Proceedings of ICISC 04*, Lecture Notes in Computer Science 3506, pp.87-103, Springer,2005.

[23] M. Yoshida, T. Fugiwara. An efficient traitor tracing scheme for broadcast encryption. In *Proceedings of 2000 IEEE International Symposium on Information Theory*, pp.463, IEEE Press, 2000.