SECURE CLOUD STORAGE WITH ATTRIBUTE BASED ENCRYPTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CEYDA TUĞBA BAĞLAÇER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

JULY 2019

Approval of the thesis:

SECURE CLOUD STORAGE WITH ATTRIBUTE BASED ENCRYPTION

submitted by **CEYDA TUĞBA BAĞLAÇER** in partial fulfillment of the requirements for the degree of **Master of Science in Cryptography Department, Middle East Technical University** by,

Prof. Dr. Ömür Uğur _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh Özbudak _____
Head of Department, **Cryptography**

Prof. Dr. Murat Cenk _____
Supervisor, **Cryptography, METU**

Examining Committee Members:

Prof. Dr. Şeref Sağıroğlu
Computer Engineering, Gazi University _____

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU _____

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: CEYDA TUĞBA BAĞLAÇER

Signature :

ABSTRACT

SECURE CLOUD STORAGE WITH ATTRIBUTE BASED ENCRYPTION

Bağlaçer, Ceyda Tuğba
M.Sc., Department of Cryptography
Supervisor : Prof. Dr. Murat Cenk

July 2019, 67 pages

As data storage needs increase, importance of and need for public cloud storage systems increase as well. However, given the amount, variety and importance of the data security can become a significant concern. Attribute based encryption schemes are important tools that can be used for access control in secure cloud storage systems. There are many attribute based encryption schemes proposed overs the years; key policy, ciphertext policy and multi-authority schemes. It is important to choose the most suitable attribute based encryption scheme to use in secure cloud storage systems. In this work, we investigate which Attribute Based Encryption Scheme would be suitable to use in a global scale secure cloud storage system. We analyze 5 different Attribute Based Encryption Schemes about; how they work, usage of pairings and suitability global scale secure cloud storage systems. After choosing the most suitable scheme we then discuss how we can improve performance of this scheme.

Keywords: Attribute Based Encryption, Key Policy, Ciphertext Policy, Multiauthority, Cloud Storage

ÖZ

ÖZELLİK TABANLI ŞİFRELEME İLE GÜVENLİ BULUT DEPOLAMA

Bağlaçer, Ceyda Tuğba
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Murat Cenk

Temmuz 2019, 67 sayfa

Veri depolama ihtiyaçları arttıkça, bulut depolama sistemlerinin önemi ve onlara duyulan ihtiyaç da artmaktadır. Bununla birlikte, verinin miktarı, çeşitliliği ve önemi göz önünde bulundurulduğunda veri güvenliği önemli bir sorun haline gelebilir. Özellik tabanlı şifreleme şemaları, güvenli bulut depolama sistemlerinde erişim kontrolü için kullanılabilecek önemli araçlardır. Yıllar içerisinde önerilen birçok özellik tabanlı şifreleme şeması vardır; anahtar politika, şifreli metin politikası ve çok otoriteli şemalar. Güvenli bulut depolama sistemlerinde kullanmak için en uygun özellik tabanlı şifreleme şemasını seçmek önemlidir. Bu çalışmada, hangi Özellik Bazlı Şifreleme Programının global ölçekte güvenli bir bulut depolama sisteminde kullanım için uygun olacağını araştırılmıştır. 5 farklı Özellik Tabanlı Şifreleme şemasının; çalışma şekilleri, eşleşmelerin kullanımı ve global ölçekte güvenli bulut depolama sistemlerine uygunluğu analiz edilmiştir. En uygun şema seçildikten sonra, bu şemanın performansının nasıl artırabileceği tartışılmıştır.

Anahtar Kelimeler: özellik tabanlı şifreleme, anahtar politikası, şifreli metin politikası, çok otoriteli, bulut depolama

To My Family...

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisor Assoc. Prof. Dr. Murat Cenk for his patient guidance, enthusiastic encouragement and valuable advices during the development and preparation of this thesis.

I am also grateful to my family for their encouragement during this study and all my life. I especially thank my mother for sacrifices she had made in this process. Lastly, I would like to thank my father for his support, I only wish he could see my finish this thesis.

TABLE OF CONTENTS

ABSTRACT vii

ÖZ ix

ACKNOWLEDGMENTS xiii

TABLE OF CONTENTS xv

LIST OF ABBREVIATIONS xix

CHAPTERS

1 INTRODUCTION 1

1.1 Secure Cloud Storage 1

1.2 Attribute Based Encryption 3

1.3 About The Thesis 5

2 BACKGROUND 7

2.1 Access Structure 7

2.2 Lagrange Coefficient 7

2.3 Prime Order Bilinear Groups 8

2.4 Composite Order Bilinear Groups 10

2.5 Threshold Scheme for Secret Sharing [22] 12

2.6 Linear Secret Sharing Scheme (LSSS) 12
2.7 Monotone Span Program (MSP)[15] 14
2.8 Dual Pairing Vector Spaces [19, 17] 14
2.9 The Subspace Assumption [17] 15
3 ATTRIBUTE BASED ENCRYPTION SCHEMES 17
3.1 Fuzzy Identity Based Encryption Scheme (FIBE) 17
3.1.1 Construction of FIBE Scheme[21] 18
3.1.2 Security 20
3.1.3 Efficiency 21
3.1.4 Secure Cloud Storage System Suitability Analysis . 22
3.2 Key Policy Attribute Based Encryption Scheme (KP-ABE) . 22
3.2.1 Construction of the KP-ABE Scheme [15] 24
3.2.2 Security 27
3.2.3 Efficiency 28
3.2.4 Secure Cloud Storage System Suitability Analysis . 28
3.2.5 KP-ABE with Non-monotonic Access Structure . . 29
3.3 Ciphertext Policy Attribute Based Encryption 29
3.3.1 Construction of the CP-ABE scheme [3] 30
3.3.2 Security 33
3.3.3 Efficiency 35
3.3.4 Secure Cloud Storage System Suitability Analysis . 35

3.4 Multi-Authority Attribute Based Encryption 36

3.5 Multi-Authority Ciphertext Policy Attribute Based Encryption Scheme 38

3.5.1 Security 41

3.5.2 Efficiency 57

3.5.3 Secure Cloud Storage System Suitability Analysis . 57

4 MULTI-AUTHORITY CP-ABE SCHEME WITH PRIME ORDER BILINEAR GROUP SETTING 59

5 CONCLUSION 63

REFERENCES 65

LIST OF ABBREVIATIONS

$\mathbb{R}$	Real Numbers
$\mathbb{Z}$	Integers
AA	Attribute Authority
ABE	Attribute Based Encryption
CA	Central Authority
CP-ABE	Ciphertext Policy Attribute Based Encryption
FIBE	Fuzzy Identity Based Encryption
IBE	Identity Based Encryption
KP-ABE	Key Policy Attribute Based Encryption
LSSS	Linear Secret Sharing Sheme
MABE	Multi-authority Attribute Based Encryption
MSP	Monotone Span Program

CHAPTER 1

INTRODUCTION

1.1 Secure Cloud Storage

Increase of volume of data to be stored prompted many individual users and organizations to outsource their storage needs. This model of data storage is called the cloud storage. By using storage services based on public clouds customer can evade the costs of having and maintaining private storage infrastructure while having the benefits such as availability, reliability, data sharing and efficient retrieval. These benefits can be described as below:

- **availability:** data should be accessible with any machine and any time.
- **reliability:** data should be backed up in a way that loss is prevented
- **data sharing:** customers of public cloud storage services are able to share their data with parties they choose.
- **efficient retrieval:** while availability is very important, data retrieval times are also important for efficiency of the public cloud storage system.

However, for enterprises and government organizations confidentiality and integrity of the data in public cloud storage services becomes a significant concern given the amount, variety and importance of the data to be stored. We can define these properties as:

- **confidentiality:** cloud storage provider is not able to learn anything about customers' data

- **integrity:** if a customer data is altered by the cloud storage provider, then customer should be able to detect this alteration.

Therefore, there is a need for a secure cloud service that provides confidentiality and integrity while providing the benefits availability, reliability, data sharing and efficient retrieval.

To address this need, Kamara and Lauter [16] proposed a cryptographic cloud storage system using the concepts of searchable encryption, attribute based encryption and symmetric encryption. Kamara-Lauter cryptographic cloud storage system works as follows;

- First, data is prepared by indexing it with data processor and then data is encrypted using a symmetric encryption scheme with a unique key.
- Second, the index, that is created in the first step, is encrypted with a searchable encryption scheme.
- Third, the unique key, used in the first step, is encrypted using an attribute based encryption scheme with an appropriate policy.
- Finally, encrypted data that is created in the first step and index encrypted with searchable encryption scheme encoded together.

Then, we can say that in Kamara-Lauter cryptographic cloud storage system, attribute based encryption scheme is used for access control.

Assume user A wants to encrypt a data M and store it on the cryptographic cloud storage. A also wants to share the data with parties that have certain attributes. Attribute based encryption is used to encrypt unique key that can decrypt the encrypted data, so that only parties that have attributes user A specified has access to the unique key. Therefore, only those parties can decrypt the ciphertext and access the data.

There are many attribute based encryption shemes proposed over the years. While choosing an ABE scheme to use in Kamara-Lauter cryptographic cloud storage access control, we need to make sure it satisfies the following conditions:

- Being compatible with global scale construction and scalability.
- Having fine grained access control.
- Efficient decryption because cloud storage system should have efficient retrieval.
- Confidentiality property of the cloud storage system must be maintained, i.e. storage provider shouldn't have access to unique keys encrypted with the ABE scheme.

1.2 Attribute Based Encryption

We can view Attribute Based Encryption (ABE) systems as a generalized version of Identity Based Encryption (IBE) system. Considering identity as an attribute, an IBE system is an ABE system where ciphertexts are associated with only one attribute.

IBE idea, introduced by Shamir [23], allows users to use any string they choose as their public key. This means that, only requirement for sending a message to a recipient is knowing recipients' identity. Thus, in this system the need for a public key distribution infrastructure is eliminated. The first complete IBE systems proposed by Boneh and Franklin [9] and Cocks [13] in 2001. After [9, 13] were published, IBE received a lot of attention and [4, 5, 24, 10] were published soon after.

First ABE scheme proposed in 2005 by Sahai and Waters [21] and it was called Fuzzy Identity Based Encryption (FIBE). In FIBE scheme users are identified with a set of explanatory attributes and both ciphertext and user's key are associated with attribute sets. A specific key is able to decrypt a specific ciphertext only if they have d common attributes between attributes of key and attributes of ciphertext. Where d is a threshold value. Thus, FIBE scheme allows sender to specify who should be able to decrypt the message, in terms of attributes alone. Note that, with this scheme more than one user can have attributes that satisfy ciphertext. Most important of the Sahai-Waters scheme is that their initial construction was limited in terms of expressibility of who can decrypt the ciphertext because formula over attributes consisted of one threshold gate.

In following work, Goyal, Pandey, Sahai and Waters [15] introduced two forms of ABE.

- In Key Policy Attribute Based Encryption (KP-ABE), a sender labels ciphertext with a set of descriptive attributes and a trusted authority issues each user a private key that is labeled with an access structure over attributes. A user can successfully decrypt the ciphertext only if ciphertext's attributes satisfies the access structure that is in the user's private key.

- We can consider Ciphertext Policy Attribute Based Encryption (CP-ABE) scheme as the reverse of the KP-ABE scheme. In CP-ABE schemes, a sender labels ciphertexts with access structures and a trusted authority issues each user a private key that is labeled with an attribute set. A user is able to successfully decrypt the message only if private key of the user satisfies the access structure of the ciphertext.

While introducing the idea of KP-ABE and CP-ABE systems, Goyal et al. [15] increased the expressibility of ABE systems by using monotone access structures which consists of **AND**, **OR** and threshold gates. In 2007, Ostrovsky, Sahai and Waters proposed a KP-ABE scheme with non-monotonic access structure. Non-monotonic access structures use **NOT** gates in addition to **AND**, **OR** and threshold gates.

In 2007, Chase [11] claimed that one major drawback of ABE schemes is the need to go through a trusted party. As a result Chase [11] proposed a multi-authority attribute based encryption (MABE) scheme based on the FIBE [21] scheme. To realize a multi-authority scheme Chase defined two techniques: using a global identifier (GID) and central authority (CA). Other MABE schemes such as [12, 18] were proposed after [11].

An important security challenge of ABE schemes is to prevent collusion attacks. In particular, group of users, where normally none of them can decrypt the ciphertext alone, should not be able to combine their keys and create a joint key that can decrypt the ciphertext. This property is called ***collusion resistance***.

1.3 About The Thesis

We consider the Kamara-Lauter secure cloud storage model and assume that customer does not completely trust the public cloud storage provider. We try to solve access control problem of these secure cloud storage services.

In Chapter 2, we give necessary background to understand how different ABE schemes work and security of said schemes.

In Chapter 3, different ABE schemes, i.e. FIBE [21], KP-ABE [15], CP-ABE [3], ABE with non-monotonic access structure [20] and MABE [11, 12, 18], are examined. We also evaluate which ABE scheme is more suitable to use in Kamara-Lauter cryptographic cloud storage access control, in this chapter.

In Chapter 4, we discuss how to improve the ABE scheme we choose in the previous chapter.

Finally in Chapter 5 we give the conclusion.

CHAPTER 2

BACKGROUND

In this chapter, we give formal definitions on access structures, and relevant background on prime and composite order bilinear groups, Linear Secret Sharing Schemes (LSSS) and Monotone Span Program. Additionally we give security assumptions of different attribute based encryption (ABE) schemes.

2.1 Access Structure

Definition 2.1 (**Access Structure** [**1**])**.** Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$ for $\forall B, C$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e. $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets that are in $\mathbb{A}$ are called the authorized sets, and the sets that are not in $\mathbb{A}$ are called the unauthorized sets.

Schemes given in [15, 3, 18], use monotone access structures where attributes play the role of the parties. Thereby, in these schemes access structure denoted by $\mathbb{A}$ contains authorized sets of attributes.

2.2 Lagrange Coefficient

Definition 2.2 (**Lagrange Coefficient**)**.** Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set of elements S in $\mathbb{Z}_p$ defined as: $\Delta_{i,S}(x) = \prod_{\substack{j \in S \\ j \neq i}} \frac{x-j}{i-j}$

2.3 Prime Order Bilinear Groups

Definition 2.3 ([8]). Let G_1, G_2 and G_T be multiplicative cyclic groups and $|G_1| = |G_2| = p$ where p is a prime. Let $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and $e : G_1 \times G_2 \rightarrow G_T$ be a bilinear map. e has the following features:

1. **Bilinearity:** $\forall a \in G_1$, $\forall b \in G_2$ and $u, t \in \mathbb{Z}_p$, we have $e(a^u, b^t) = e(a, b)^{ut}$.
2. **Non-degeneracy:** $e(g_1, g_2) \neq 1$.

Several ABE schemes consider groups where $G_1 = G_2$.

Definition 2.4 ([6, 17]). Let G, G_T be multiplicative cyclic groups and $|G| = p$ where p is a prime. Let $G = \langle g \rangle$ and $e : G \times G \rightarrow G_T$ denote to a bilinear map. e has the following features:

1. **Bilinearity:** $\forall a, b \in G$ and $u, t \in \mathbb{Z}_p$, we have$e(a^u, b^t) = e(a, b)^{ut}$.
2. **Non-degeneracy:** $e(g, g) \neq 1$.

Continuing with Definition 2.4, we will also consider vectors of group elements. For $v = (v_1, v_2, \ldots, v_n) \in \mathbb{Z}_p^n$ and $a \in G$, n-tuple of elements of G can be denoted as $a^v = (a^{v_1}, a^{v_2}, \ldots, a^{v_n})$.

$\forall t \in \mathbb{Z}_p$ and $v, w \in \mathbb{Z}_p^n$, we can perform following operations:

$$a^{tv} = (a^{tv_1}, a^{tv_2}, \ldots, a^{tv_n})$$

$$a^{v+w} = (a^{v_1+w_1}, a^{v_2+w_2}, \ldots, a^{v_n+w_n})$$

Let e_n denote to the product in terms of component pairings:

$$e_n(g^v, g^w) = \prod_{i=1}^{n} e(a^{v_i}, a^{w_i}) = e(g, g)^{v \cdot w}$$

Definition 2.5 (Decisional Bilinear Diffie-Hellman (d-BDH) Assumption [6, 21, 15]). Let G and G_T be multiplicative cyclic groups of prime order p. Let $G = \langle g \rangle$ and $e : G \times G \rightarrow G_T$ denote to a bilinear map and $s, t, u, z \xleftarrow{R} \mathbb{Z}_p$ are chosen and $S = g^s, T = g^t, U = g^u$.

The d-BDH assumption states that no probabilistic polynomial time algorithm $\mathcal{B}$ can distinguish the $(S, T, U, e(g,g)^{stu})$ from the $(S, T, U, e(g,g)^{z})$ with no more than negligible advantage.

Algorithm $\mathcal{B}$ outputs a bit in $\{0,1\}$ and has the advantage $Adv_{\mathcal{B}}^{d-BDH} = \epsilon$ in solving the d-BDH problem.

$$|Pr[\mathcal{B}(S,T,U,e(g,g)^{stu}) = 0] - Pr[\mathcal{B}(S,T,U,e(g,g)^{z}) = 0]| \geq \epsilon$$

Definition 2.6 (Decisional Modified Bilinear Diffie-Hellman Assumption [21]). Let G and G_T be multiplicative cyclic groups of prime order p. Let $G = \langle g \rangle$ and $e : G \times G \rightarrow G_T$ denote to a bilinear map and $s, t, u, z \overset{R}{\leftarrow} \mathbb{Z}_p$ be chosen and $S = g^s, T = g^t, U = g^u$.

The Decisional Modified Bilinear Diffie-Hellman Assumption states that no probabilistic polynomial time algorithm $\mathcal{B}$ can distinguish the $(S, T, U, e(g,g)^{\frac{st}{u}})$ from te tuple $(S, T, U, e(g,g)^{z})$ with no more than negligible advantage.

Algorithm $\mathcal{B}$ outputs a bit in $\{0,1\}$ and has the advantage ϵ in solving the problem.

$$|Pr[\mathcal{B}(S,T,U,e(g,g)^{\frac{st}{u}}) = 0] - Pr[\mathcal{B}(S,T,U,e(g,g)^{z}) = 0]| \geq \epsilon$$

Definition 2.7 (Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption [25]). Let G be a group with $|G| = p$ where p is a prime and $G = \langle g \rangle$. Let $s, t, b_1, \ldots, b_q \overset{R}{\leftarrow} Z_p$ be chosen. If an adversary is given $y =$

$$g, g^t, g^s, \ldots, g^{s^q}, \ \ , g^{s^{q+2}}, \ldots, g^{s^{2q}}$$

$$\forall_{1 \leq j \leq q} \ \ g^{t \cdot b_j}, g^{s/b_j}, \ldots, g^{s^q/b_j}, \ \ , g^{s^{q+2}/b_j}, \ldots, g^{s^{2q}/b_j}$$

$$\forall_{1 \leq j,\, k \leq q,\, k \neq j} \ \ g^{s \cdot t \cdot b_k / b_j}, \ldots, g^{s^q \cdot t \cdot b_k / b_j}$$

it must remain hard to distinguish $g^{s^{q+1}t} \in G_T$ from a random element Z in G_T. An algorithm $\mathcal{B}$ that outputs a bit in $\{0,1\}$ has advantage ϵ in Decisional q-Parallel Diffie-Hellman Exponent Problem in G with

$$|Pr[\mathcal{B}(y, e(g,g)^{s^{q+1}t}) = 0] - Pr[\mathcal{B}(y, Z) = 0]| \geq \epsilon$$

Definition 2.8 (Decisional q-Bilinear Diffie-Hellman Exponent Assumption [25]). Let G be a group with $|G| = p$ where p is a prime and $G = \langle g \rangle$. Let $s, t \overset{R}{\leftarrow} Z_p$. If an

adversary is given $y = g, g^t, g^s, \ldots, g^{s^q}, g^{s^{q+2}}, \ldots, g^{s^{2q}}$ it must remain hard to distinguish $g^{s^{q+1}t} \in G_T$ from a random element Z in G_T. An algorithm $\mathcal{B}$ that outputs a bit in $\{0, 1\}$ has advantage ϵ in Decisional q-Parallel Diffie-Hellman Exponent Problem in G with

$$|Pr[\mathcal{B}(y, e(g, g)^{s^{q+1}t}) = 0] - Pr[\mathcal{B}(y, Z) = 0]| \geq \epsilon$$

Definition 2.9 (**Decisional Linear Assumption [17]**). Given a group generator $\mathcal{G}$, following distribution is defined:

$$\mathbb{G} = (p, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$g, f, v, w \xleftarrow{R} G, \qquad c_1, c_2, w \xleftarrow{R} \mathbb{Z}_p$$

$$D = (g, f, v, f^{c_1}, v^{c_2})$$

For probabilistic polynomial time algorithm $\mathcal{B}$ with output in $\{0, 1\}$ we assume that,

$$Adv_{\mathcal{G},\mathcal{B}} = |Pr[\mathcal{B}(D, g^{c_1+c_2})] = 1 - Pr[\mathcal{B}(D, g^{c_1+c_2+w})] = 1|$$

is negligible in security parameter κ.

Note that, $w \xleftarrow{R} \mathbb{Z}_p$ denotes that w is uniformly random element of $\mathbb{Z}_p$.

2.4 Composite Order Bilinear Groups

Definition 2.10 (**Composite Order Bilinear Groups [17]**). Let G be a bilinear group of composite order $N = p_1 p_2 \ldots p_m$ where $p_1, p_2, \ldots, p_m$ are m distinct primes and $e : G \times G \to G_T$ denote its bilinear map. Both G,G_T are cyclic groups and $|G| = |G_T| = N$. e has the bilinearity and non-degeneracy features. For each p_i, G has a subgroup G_{p_i} of order p_i. We let $g_1, g_2, \ldots, g_m$ denote generators of these subgroups as $G_{p_1} = \langle g_1 \rangle, G_{p_2} = \langle g_2 \rangle, \ldots, G_{p_m} = \langle g_m \rangle$.

Each element $h \in G$ can be stated as $h = {g_1}^{a_1} {g_2}^{a_2} \ldots {g_m}^{a_m}$ for some $a_1, a_2, \ldots, a_m \in \mathbb{Z}_N$ where each a_i is unique modulo p_i. ${g_i}^{a_i}$ is referred as the G_{p_i} component of h. When $a_i \equiv 0 \mod p_i$, it is said that h has no G_{p_i} component. The subgroups $G_{p_1}, G_{p_2}, \ldots, G_{p_m}$ are orthogonal under the bilinear map e, i.e. if $t \in G_{p_i}$ and $w \in G_{p_j}$ for $i \neq j$, then $e(t, w) = 1$, where 1 is the identity element of the group G_T.

For each non-empty subset $S \subseteq [m]$, there is a associated subgroup of order $\prod_{i \in S} p_i$ in G, which is denoted by G_S. Let $\mathcal{G}$ be a group generation algorithm which takes in m and security parameter κ then outputs a bilinear group G with $|G| = N$. Also let $(N, G, G_T, e) \xleftarrow{R} \mathcal{G}$ denote the generation of the group G by running the algorithm $\mathcal{G}$. Finally let, $Z_i \xleftarrow{R} G_{S_i}$ mean that Z_i is chosen to be a random generator of G_{S_i}.

Definition 2.11 (General Subgroup Decision Assumption [17]). Let $S_0, S_1, \ldots, S_k$ be non-empty subsets of $[m]$ such that for each $2 \leq j \leq k$, $S_j \cap S_0 = \emptyset = S_j \cap S_1$ or $S_j \cap S_0 \neq \emptyset \neq S_j \cap S_1$. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$Z_0 \xleftarrow{R} G_{S_0},\ Z_1 \xleftarrow{R} G_{S_1},\ \ldots\ , Z_k \xleftarrow{R} G_{S_k}$$

$$D = (\mathbb{G}, Z_2, \ldots, Z_k)$$

We assume that for probabilistic polynomial time algorithm $\mathcal{B}$ with output in $\gamma \in \{0, 1\}$,

$$Adv_{\mathcal{G},\mathcal{B}} = |Pr[\mathcal{B}(D, Z_0)] = 1 - Pr[\mathcal{B}(D, Z_1)] = 1|$$

is negligible in security parameter κ.

In multi-authority attribute based encryption scheme described by Lewko and Waters in [18], composite order bilinear group G with $|G| = p_1 p_2 p_3$ with p_1, p_2, p_3 are three distinct primes.

Definition 2.12 (Composite Order Bilinear Groups with order $N = p_1 p_2 p_3$ [18]). Let $\mathcal{G}$ be a group generator algorithm such that $\mathcal{G}(\lambda) \rightarrow (p_1, p_2, p_3, G, G_T, e)$ where κ is a security parameter, G is a bilinear group, p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups with $|G| = |G_T| = N = p_1 p_2 p_3$ and $e : G \times G \rightarrow G_T$ is map which has the features bilinearity and non-degeneracy.

We assume that group operations are computable in polynomial time with respect to λ in groups G, G_T and bilinear map e. The group descriptions of G and G_T include generators of the respective cyclic groups. Let $G_{p_1}, G_{p_2}, G_{p_3}$ denote the subgroups in G with $|G_{p_1}| = p_1, |G_{p_2}| = p_2, |G_{p_3}| = p_3$.

Note that, when $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, $e(h_i, h_j) = 1$ where 1 is the identity element in G_T. Suppose $h_1 \in G_{p_1}$ and $h_2 \in G_{p_2}$ and $G = \langle g \rangle$. Then, $g^{p_1 p_2}$

generates G_{p_3}, $g^{p_1p_3}$ generates G_{p_2} and $g^{p_2p_3}$ generates G_{p_1}. Then for some α_1, α_2, $h_1 = (g^{p_2p_3})^{\alpha_1}$ and $h_2 = (g^{p_1p_3})^{\alpha_2}$. Then:

$$e(h_1, h_2) = e(g^{p_2p_3\alpha_1}, g^{p_1p_3\alpha_2}) = e(g^{\alpha_1}, g^{p_3\alpha_2})^{p_1p_2p_3} = 1$$

This orthogonality property of $G_{p_1}, G_{p_2}, G_{p_3}$ is used to implement semi-fuctionality in multi-authority ABE scheme given in [18].

2.5 Threshold Scheme for Secret Sharing [22]

Consider a k-dimensional plane, assume k-points $(x_1, y_1), \ldots, (x_k, y_k)$ with distinct x_i's are given in the said plane. There there is a sole polynomial $q(x)$ with $deg(q(x)) = k - 1$ such that $q(x_i) = y_i$ for $\forall i$.

To divide a number L into i pieces L_i, we pick a random polynomial $q(x)$ with $deg(q(x)) = k - 1$ such that $q(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1}$ where $a_0 = L$ and evaluate: $L_1 = q(1), \ldots, L_i = q(i)$. Given a subset of k of these L_i values and their identifying indexes, we can find the coefficients of $q(x)$ by interpolation. Finally we can calculate $L = q(0)$. Note that, knowing $k - 1$ of these values, does not suffice in order to calculate the number L.

To be more precise, we can use modular arithmetic. Consider the field $\mathbb{Z}_p$, where p is a prime. Given integer L, where $p > L$ and $p > n$. The coefficients $a_1, \ldots, a_{k-1}$ in $q(x)$ are chosen uniformly at random over the integers in the interval $[0, p)$ and $L_1, \ldots, L_n$ are computed $\mod p$. Assume $k - 1$ of these L_i values are revealed to the adversary. Using p possible values L in the interval $[0, p)$ adversary can construct only one polynomial $q'(x)$ of degree $k - 1$ such that $q'(0) = L'$ and $q'(i) = L_i$ for the $k - 1$ given arguments. Since these p possible polynomials are equally likely, the adversary can't deduce anything about the real value L.

2.6 Linear Secret Sharing Scheme (LSSS)

Our definition of Linear Secret Sharing Scheme is adapted from definitions given in [1].

Definition 2.13 (**Linear Secret Sharing Scheme**)**.** Consider a secret sharing scheme Π over a set of parties $\mathcal{P}$. Π is considered to be linear over $\mathbb{Z}_p$ if;

1. The shares of each party of $\mathcal{P}$ form a vector over $\mathbb{Z}_p$.

2. There exists a share generating matrix, for the secret sharing scheme Π, that is denoted by M and M has l rows and n columns. Where $\rho(x)$ is a function such that $\rho(x) : \{1, \ldots, l\} \rightarrow \mathcal{P}$, $\forall x \in \{1, \ldots, l\}$ x^{th} row of the matrix M is labeled with $\rho(x)$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$ where $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen and $s \in \mathbb{Z}_p$ is the secret to be shared, then Mv is the vector of l shares of the secret s according to Π. The share $(Mv)_x$ belongs to party the $\rho(x)$.

In [1] it is shown that every LSSS described as above definition also has the linear reconstruction property.

Definition 2.14 (**Linear Reconstruction Property**[1])**.** Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be an authorized set and let $I \subset \{1, 2, \ldots, l\}$ be defined as $I = \{x : \rho(x) \in S\}$. Note that, by convention the target vector of any LSSS is the vector $(1, 0, 0, \ldots, 0)$. For any satisfying set of rows I in M, we will have that the target vector is in the span of I. Then there exists constants $\{w_x \in \mathbb{Z}_p\}_{x \in I}$ such that, if $\{\lambda_x\}$ are valid shares of any secret s according to Π, then $\sum_{x \in I} w_x \lambda_x = s$. Constants $\{w_x\}$ can be found in polynomial time with respect to the size of the share-generating matrix M.

For multi-authority ABE scheme given in [18] we need to consider composite order group bilinear group construction given in Definition 2.12, i.e. LSSS matrices over $\mathbb{Z}_N$ where $N = p_1 p_2 p_3$ with p_1, p_2, p_3 are distinct primes. Assume S is a set that is authorized and access matrix rows which are labeled by the elements of set S, have $(1, 0, \ldots, 0)$ vector in their span modulo N. In the security proof of [18] it is also assumed that, rows of $\mathbb{A}$ which corresponds to an unauthorized set do not include the vector $(1, 0, \ldots, 0)$ in their span modulo p_1, p_2 or p_3.

2.7 Monotone Span Program (MSP)[15]

Definition 2.15 (**Monotone Span Program**). Let $\mathcal{H}$ be a field and $\{t_1, \ldots, t_n\}$ be a set of variables. Also let M be a matrix over $\mathcal{H}$ and ρ is a labeling of the rows M such that every row of M is labeled with one the variables $t_i \in \{t_1, \ldots, t_n\}$. Then a monotone span program (MSP) over field $\mathcal{H}$ is the labeled matrix $\hat{M}(M, \rho)$.

Assume γ is a set of variables. For every γ, define a submatrix M_γ of M where M_γ consists of the rows of M which are labeled with $t_i \in \gamma$. A MSP accepts or rejects an input according to the following criterion.

Criterion: MSP $\hat{M}$ accepts γ if and only if some linear combination of the rows of $\hat{M}$ gives the all-one vector $\vec{1}$, i.e. $\vec{1} \in span(M_\gamma)$. $\hat{M}$ computes a Boolean function $f_M(\gamma)$ if it accepts exactly those inputs γ where $f_M(\gamma) = 1$. The size of $\hat{M}$ is the number of rows in M.

2.8 Dual Pairing Vector Spaces [19, 17]

Consider the six-tuple (p, G, G_T, g, g_T, e), where G and G_T are multiplicative cyclic groups with $|G| = |G_T| = p$ and $p$is a prime, $G = \langle g \rangle$, e is a bilinear map such that $e : G \times G \to G_T$ and $g_T = e(g, g) \neq 1$.

We choose two random bases $\mathbb{B} = (b_1, b_2, \ldots, b_n)$ and $\mathbb{B}^* = (b_1^*, b_2^*, \ldots, b_n^*) \in \mathbb{Z}_p^n$ up to the constraint that pair $(\mathbb{B}, \mathbb{B}^*)$ is dual orthonormal, where n is a fixed dimension and b_i, b_i^* denotes to vectors forms the bases $\mathbb{B}, \mathbb{B}^*$ respectively.We say that $(\mathbb{B}, \mathbb{B}^*)$ is dual orthonormal if;

- Whenever $i \neq j$, $b_i \cdot b_j^* = 0 \ (mod\ p)$
- For all $i = 1, \ldots, n$, $b_i \cdot b_i^* = \psi$ where ψ is a uniformly random element in $\mathbb{Z}_p$.

Note that, for g of $G = \langle g \rangle$, $e(g^{b_i}, g^{b_i^*}) = 1$ as long as $i \neq j$, where 1 denotes to the identity element of the group G_T. If $\mathbb{B}$ and $\mathbb{B}^*$ are dual orthonormal bases, then they will be denoted by $(\mathbb{B}, \mathbb{B}^*) \in Dual(\mathbb{Z}_p^n)$.

2.9 The Subspace Assumption [17]

In [17] Lewko intorduced a complexity assumption, called the Subspace Assumption, in prime order groups that imitate the effects of subgroup decision assumption in composite order groups given in Definition 2.11. Lewko also showed that the subspace assumption is implied by the subgroup decision assumption. The subspace assumption employs dual pairing vector spaces given in section 2.8.

Instead of subgroups that are used in the composite order groups, basis vectors used in the exponent is used in the prime order groups. If we have dual orthonormal bases $\mathbb{B} = (b_1, \ldots, b_n), \mathbb{B}^* = (b_1^*, \ldots, b_n^*)$. we think "subgroup 1" in $\mathbb{B}$ corresponds to the $b_1, \ldots, b_4$ and this is orthogonal to the $b_5^*, \ldots, b_n^*$ in $\mathbb{B}^*$.

For a definite dimension $n \geq 3$ and prime p, $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$ denotes the choosing random dual orthonormal bases $\mathbb{B}$ and $\mathbb{B}^*$ of $\mathbb{Z}_p^n$ and $Dual(\mathbb{Z}_p^n)$ denotes the set of dual orthonormal bases. $x \xleftarrow{R} \mathbb{Z}_p$ denotes that x is a uniformly random element of $\mathbb{Z}_p$. Also assume k is positive integer such that $k \leq \frac{n}{3}$.

Definition 2.16 (**Subspace Assumption [17]**). Considering the groups and bilinear map definition given in Definition 2.4. Given a group generator $\mathcal{G}$ we define the following distribution:

$$\mathbb{G} = (p, G, G_T, e) \xleftarrow{R} \mathcal{G}, \qquad (\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$$

$$g \xleftarrow{R} G, \qquad \eta, \beta, \tau_1, \tau_2, \tau_3, \mu_1, \mu_2, \mu_3 \xleftarrow{R} \mathbb{Z}_p$$

$$U_1 = g^{\mu_1 b_1 + \mu_2 b_{k+1} + \mu_3 b_{2k+1}}, \;\; U_2 = g^{\mu_1 b_2 + \mu_2 b_{k+2} + \mu_3 b_{2k+2}}, \;\ldots,\; U_k = g^{\mu_1 b_k + \mu_2 b_{2k} + \mu_3 b_{3k}}$$

$$V_1 = g^{\tau_1 \eta b_1^* + \tau_2 \beta b_{k+1}^*}, \;\; V_2 = g^{\tau_1 \eta b_2^* + \tau_2 \beta b_{k+2}^*}, \;\ldots,\; V_k = g^{\tau_1 \eta b_k^* + \tau_2 \beta b_{2k}^*}$$

$$W_1 = g^{\tau_1 \eta b_1^* + \tau_2 \beta b_{k+1}^* + \tau_3 b_{k+1}^*}, \;\; W_2 = g^{\tau_1 \eta b_2^* + \tau_2 \beta b_{k+2}^* + \tau_3 b_{2k+2}^*}, \;\ldots,\; W_k = g^{\tau_1 \eta b_k^* + \tau_2 \beta b_{2k}^* + \tau_3 b_{3k}^*}$$

$$D = (g^{b_1}, g^{b_2}, \ldots, g^{b_{2k}}, g^{b_{3k+1}}, \ldots, g^{b_n}, g^{\eta b_1^*}, \ldots, g^{\eta b_k^*}, g^{\beta b_{k+1}^*}, \ldots, g^{\beta b_{2k}^*},$$
$$g^{b_{2k+1}^*}, \ldots, g^{b_n^*}, U_1, U_2, \ldots, U_k, \mu_3)$$

We assume that for any probabilistic polynomial time algorithm $\mathcal{B}$ with output in $\{0, 1\}$,

$$Adv_{\mathcal{G},\mathcal{B}} = |Pr[\mathcal{B}(D, V_1, \ldots, V_k) = 1] - Pr[\mathcal{B}(D, W_1, \ldots, W_k) = 1]|$$

is negligible in security parameter κ.

CHAPTER 3

ATTRIBUTE BASED ENCRYPTION SCHEMES

In this chapter, we will look at different ABE schemes and examine how these schemes work, their security, efficiency and suitability to used in secure cloud storage system.

3.1 Fuzzy Identity Based Encryption Scheme (FIBE)

In this section, we examine Fuzzy Identity Based Encryption (FIBE) scheme introduced by Sahai and Waters [21]. In FIBE scheme, identity of the user defined by a set of attributes. Let the universe of attributes this system is defined in denoted by $\mathcal{U}$, then identity w is a subset of $\mathcal{U}$, i.e. $w \subseteq \mathcal{U}$.

This scheme uses four algorithms:

1. **Setup**, algorithm takes an error tolerance or threshold value d as an input and it gives the public parameters PP and a master key MK as output.
2. **Key Generation**, is an algorithm that takes the identity w and master key MK as inputs and it gives the private key PK as an output.
3. **Encryption**, is an algorithm that takes the public parameters PP, identity w' and message M as inputs and it gives the ciphertext C as an output.
4. **Decryption** algorithm takes the ciphertext C (encrypted under the attribute set of the identity w'), the private key PK generated for the identity w and threshold value d as inputs. If $|w \cap m'| \geq d$ then decryption algorithm outputs the message M as an output.

In this scheme, private key PK that is generated for the user with identity w, constructed as a set of components where there is a component for each attribute $a \in w$. In key generation algorithm, shares of the secret values distributed across the exponents of the user's private key components using Shamir's method of Threshold Scheme for Secret Sharing [22] given in Subsection 2.5. By using Threshold Scheme for Secret Sharing method, Sahai and Waters ensured that FIBE scheme has the error-tolerance property because only a subset of private key components are needed to decrypt a message.

In FIBE scheme, to provide collusion resistance, each users' private key components created with a different polynomial. When multiple users try to collude, they won't be able to combine their private keys in a useful way because their keys were generated with different polynomials.

The threshold value d represent the error-tolerance i.e. w and w' should have minimum d overlapping attributes. While a users' private key is generated, authority chooses a random polynomial $q(x)$ with $deg(q) = d - 1$ up to the constraint that $q(0) = y$ where y is the valuation point that y is the same value for each user. For each attribute that composes a user's identity, a private key element which is affiliated to the user's random polynomial $q(x)$ is issued by key generation algorithm. If user's private key components matches at least d elements of the ciphertext, then user can perform the decryption successfully.

3.1.1 Construction of FIBE Scheme[21]

In FIBE scheme, ciphertext that is encrypted under the attribute set of identity w, can be decrypted by a private key of the identity w' only if $|w \cap w'| \geq d$, where d is the threshold value.

Consider the prime order bilinear map definition given in Definition 2.4 and Lagrange Coefficient given in Definition 2.2.

Let the size of the groups G and G_T that are defined in Definition 2.4 be determined by a security parameter, κ.

Identities are subsets of some attribute universe $\mathcal{U}$, of size $|\mathcal{U}| = n$.

Setup ($\kappa, d \rightarrow PP, MK$)**:** Following steps are performed in the Setup algorithm.

- Define the universe of attributes $\mathcal{U}$. To provide simplicity, Sahai and Waters choose to take first n elements of $\mathbb{Z}_p^*$ as the universe $\mathcal{U}$, i.e. the integers $1, \ldots, n \ (mod\ p)$.
- Choose $t_i \xleftarrow{R} \mathbb{Z}_p$ for $\forall i = 1, \ldots, n$ and $y \xleftarrow{R} \mathbb{Z}_p$.
- Compute parameters T_i as, $T_i = g^{t_i}$ and Y as $Y = e(g, g)^y$.

Then public parameters PP is;

$$PP = (T_1, T_2, \ldots, T_n, Y)$$

Master key MK is;

$$MK = (t_1, t_2, \ldots, t_n, y)$$

Key Generation ($w,\ MK \rightarrow PK$)**:** To create a private key for identity $w \subseteq \mathcal{U}$ following steps are performed in the Key Generation algorithm.

- A polynomial $q(x)$ such that $deg(q) = (d-1)$ and $q(0) = y$ is chosen.
- Private components (P_i) for every attribute i of w i.e. $\forall i \in w$ as;

$$P_i = g^{\frac{q(i)}{t_i}}$$

Then private key of identity w is;

$$PK = (\{P_i\}_{\forall i \in w})$$

Encryption ($w',\ M \rightarrow C$)**:** To encrypt a message M of the form $M \in G_T$ with the identity w', following steps are performed in the Encryption algorithm.

- Choose a random value $s \in \mathbb{Z}_p$.
- Compute ciphertext components C_i for $\forall i \in w'$ as $C_i = T_i^s$.

- Compute ciphertext component C' as $C' = MY^s$.

Then ciphertext C is published as;

$$C = (w', C', \{C_i\}_{\forall i \in w'})$$

Decryption (C, PK, d): Suppose we have $|w \cap w'| \geq d$, then following steps are performed in the Decryption algorithm.

- Choose d-element subset, S, of $w \cap w'$.
- Then the ciphertext can be computed as:

$$\begin{aligned} C' / \prod_{i \in S} e(P_i, C_i)^{\Delta_{i,S}(0)} &= Me(g,g)^{sy} / \prod_{i \in S} (e(g^{\frac{q(i)}{t_i}}, g^{st_i}))^{\Delta_{i,S}(0)} \\ &= Me(g,g)^{sy} / \prod_{i \in S} (e(g,g)^{sq(i)})^{\Delta_{i,S}(0)} \\ &= M \end{aligned}$$

  Note that, in the last equality $deg(sq(x)) = d-1$ so it can be interpolated using d points.

3.1.2 Security

We now look at the security of the FIBE scheme given above. Sahai and Waters [21] define a Selective-ID model of security for FIBE. In the Fuzzy Selective-ID game the adversary is only permitted to inquire for private key for the identity as long as it has fewer than d overlapping attributes with the target identity.

Fuzzy Selective-ID [21]

Assume $\mathcal{A}$ is the adversary and $\mathcal{B}$ is the challenger.

Init $\mathcal{A}$ announces the identity α, as the identity to be challenged upon.
Setup $\mathcal{B}$ runs the Setup algorithm gives resulting PP to $\mathcal{A}$.
Phase 1 $\mathcal{A}$ is permitted to issue multiple inquiries for PKs for many identities, γ_j,

where $|\gamma \cap \alpha| < d$ for every j.

Challenge $\mathcal{A}$ presents two messages M_0 and M_1 such that length of M_0 is equal to length of M_1. $\mathcal{B}$ randomly chooses a bit $b \in \{0, 1\}$, and encrypts M_b using the identity α. Then ciphertext is presented to $\mathcal{A}$.

Phase 2 Phase 1 is done again.

Guess $\mathcal{A}$ outputs a guess b'.

The advantage of $\mathcal{A}$ in Fuzzy Selective-ID game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 3.1 ([21]). If advantage in the game above for all polynomial time adversaries are less than negligible then a FIBE scheme is secure in the Fuzzy Selective-ID model of security.

Sahai and Waters stated that the security of this scheme in the Fuzzy Selective-ID model reduces to the hardness of the Decisional Modified Bilinear Diffie-Hellman assumption given in Definition 2.6.

Theorem 3.1 ([21]). *If an adversary can break the scheme in the Fuzzy Selective-ID Model, then a simulator can be constructed to play the Decisional Modified Bilinear Diffie-Hellman game with a non-negligible advantage.*

Proof of the Theorem 3.1 can be seen in [21].

3.1.3 Efficiency

- In encryption algorithm, the number of exponentiations will depend of the number of attributes in the identity w', because $E_i = T_i^s$ for $\forall i \in w'$.
- In decryption algorithm, cost is dominated by d bilinear map computations, because we choose d-element subset, S, of $|w \cap w'|$.
- The amount of group elements the public parameters have grows linearly with the size of the attribute universe, i.e. $|\mathcal{U}| = n$, $PP = (T_1, T_2, \ldots, T_n, Y)$ where $T_i = g^{t_i}$, for all $i = 1, \ldots, n$.
- The amount of group elements that belong to a user's private key, grows linearly with the number of attributes related to their identity, because $PK = (\{P_i\}_{\forall i \in w})$ for all $P_i = g^{\frac{q(i)}{t_i}}$.

- The amount of group elements in a ciphertext grows linearly with the size of the identity we are encrypting to, because $C = (w', C' = MY^s, \{C_i\}_{\forall i \in w'})$.

3.1.4 Secure Cloud Storage System Suitability Analysis

Now we look at if FIBE scheme is suitable to use in Kamara-Lauter cryptographic cloud storage scheme according to the conditions given in Section 1.1.

- Since there is a large universe construction given in [21], we can say that it is compatible with global scale construction. Although it is important to note that this scheme is not scalable.
- There is limited expressibility in terms of access control.
- Efficiency of the decryption depends on the threshold value d.
- In this scheme there is one trusted party which is responsible for key generation. If we use this scheme in the cryptographic cloud storage system this trusted party most likely reside in the storage provider, then confidentiality property is compromised because storage provider has access to the unique keys.

3.2 Key Policy Attribute Based Encryption Scheme (KP-ABE)

In this section, we examine KP-ABE scheme introduced by Goyal et al. [15]. In this scheme, we use Access Structure given in Definition 2.1.

This KP-ABE scheme consists of four algorithms:

1. **Setup** algorithm only has a security parameter κ as an input. It gives two outputs; the public parameters PP and master key MK.
2. **Encryption** algorithm takes the message M, a set of attributes w and public parameters PP as inputs. It gives the ciphertext C as an output.
3. **Key Generation** algorithm takes an access structure $\mathbb{A}$ and the master key MK as inputs. It gives the decryption key DK as an output.

4. **Decryption** algorithm takes the ciphertext C which is encrypted with set of attributes w, the decryption key DK for access structure $\mathbb{A}$ and the public parameters PP as inputs. If $w \in \mathbb{A}$ then decryption algorithm outputs the message M, else decryption algorithm fails.

In this KP-ABE scheme Goyal et al. specified access structure as an access tree.

Construction of the Access Trees [15]

In this scheme's access tree construction:

- Private keys are tagged with access tree structures where all non-leaf nodes of the tree are threshold gates and all leaf nodes represent an attribute.
- Ciphertexts are tagged with attributes.

A user can decrypt a ciphertext with their private key if and only if attributes from the ciphertext satisfy the access tree.

Definition 3.2 (**Access Tree $\mathcal{T}$**)**.** Suppose tree $\mathcal{T}$ is an access structure. All non-leaf nodes of $\mathcal{T}$ are threshold gates and they are described by their children and a threshold value. Assume x is node of tree $\mathcal{T}$, then:

- The number of children node x has represented by num_x.
- Threshold value of node x represented by k_x and $0 < k_x \leq num_x$.
- If x is a leaf node than it is described by an attribute and $k_x = 1$.
- If x is a non-leaf node and $k_x = 1$ then x is an **OR** gate.
- If x is a non-leaf node and $k_x = num_x$ then x is an **AND** gate.

To be able to work with access trees we need to define the functions given below:

- Parent of node x denoted by $parent(x)$.
- If x is a leaf node, $att(x)$ denotes to the attribute associated with leaf node x.

- There is a relationship between every non-leaf node x and their children defined as $index(x)$, the children of node x are numbered arbitrarily from 1 to num_x.

Let r be the root node of $\mathcal{T}$ and $\mathcal{T}_x$ is the subtree of $\mathcal{T}$ which has x as a root node. Then we can say that $\mathcal{T} = \mathcal{T}_r$. Let w be a set of attributes, if w satisfies the $\mathcal{T}_x$, then $\mathcal{T}_x(w) = 1$. $\mathcal{T}_x(w)$ can be recursively computed as follows:

1. If x ia a on-leaf node then for every children z of x, compute $\mathcal{T}_z(w)$. If k_x of the children return 1, then $\mathcal{T}_x(w) = 1$.
2. If x is a leaf node, then $\mathcal{T}_x(w) = 1$ if and only if $att(x) \in w$

3.2.1 Construction of the KP-ABE Scheme [15]

Same as the FIBE scheme discussed in Section 3.1, we consider the prime order bilinear map definition given in Definition 2.4 and Lagrange Coefficient given in Definition 2.2.

Let the size of the groups G and G_T that are defined in Definition 2.4 be determined by a security parameter, κ.

Each attribute associated with a unique element in $\mathbb{Z}_p^*$

Construction of KP-ABE scheme given by Goyal et al. [15] follows:

Setup ($\kappa \rightarrow PP, MK$): Following steps are performed in the Setup algorithm.

- Define the universe of attributes as $\mathcal{U} = \{1, 2, \ldots, n\}$.
- Choose $t_i \xleftarrow{R} \mathbb{Z}_p$ for $\forall i \in \mathcal{U}$ and $y \xleftarrow{R} \mathbb{Z}_p$.
- Compute parameters T_i as $T_i = g^{t_i}$ for $\forall i \in \mathcal{U}$ and Y as $Y = e(g, g)^y$.

Then public parameters PP is published as;

$$PP = (T_1, T_2, \ldots, T_n, Y)$$

Master key MK is;

$$MK = (t_1, t_2, \ldots, t_n, y)$$

Encryption $(PP, M, w \rightarrow C)$**:** To encrypt a message $M \in G_T$ with a set of attributes w following steps are performed in the Encryption algorithm.

- Choose $s \in \mathbb{Z}_p$ as a random value.
- Compute ciphertext components C_i for $\forall i \in w$ as $C_i = T_i^s$.
- Compute ciphertext component C' as $C' = MY^s$.

Then ciphertext C is published as;

$$C = (w, C', \{C_i\}_{\forall i \in w})$$

Key Generation $(MK, \mathcal{T} \rightarrow DK)$**:** This algorithm creates a decryption key DK such that user with key DK is can decrypt the ciphertext that is encrypted under attribute set w, if and only if $\mathcal{T}(w) = 1$. To create such key following steps are performed in the Key Generation algorithm.

- Choose a polynomial q_x for every node x of the tree $\mathcal{T}$. Starting from the root node r, these polynomials are chosen in a top to bottom as follows:
  - For each node x, d_x is the degree of the polynomial q_x.
  - For root node r; set $q_r(0) = y$ and $d_r = k_r - 1$, then define polynomial q_r randomly.
  - For a node x that is not a root node; set $q_x(0) = q_{parent(x)}(index(x))$ and $d_x = k_x - 1$, then define polynomial q_x randomly.
- Compute secret values D_x for every leaf node x as;

$$D_x = g^{\frac{q_x(0)}{t_i}}$$

The set of above secret values is the decryption key DK.

$$DK = (\mathcal{T}, \{D_x\}_{\forall i = att(x)})$$

Decryption (C, DK, PP)**:** To decrypt the ciphertext C with the decryption key DK Decryption algorithm works in a recursive manner.

- Define a recursive algorithm $DecryptNode(C, DK, x)$ which outputs a group element of G_T or $\perp$ where x is a node in the tree $\mathcal{T}$.
  - If x is a leaf node and $i = att(x)$ then:

$$DecryptNode(C, DK, x) = \begin{cases} e(D_x, C_i) = e(g^{\frac{q_x(0)}{t_i}}, g^{s \cdot t_i}) = e(g,g)^{s \cdot q_x(0)} ... \\ ...if\ i \in w \\ \perp\ otherwise \end{cases}$$

  - If x is a non-leaf node; for all nodes z that are children of x, then call the algorithm $DecryptNode$ again as $F_z = DecryptNode(C, DK, z)$. Let S_x be an arbitrary set of child nodes z such that $F_z \neq \perp$ and $|S_x| = k_x$.
    - If there exists no such set, then the node x is not satisfied, i.e. $DecryptNode(C, DK, x) = \perp$
    - Otherwise,

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)},\ i = index(z),\ S'_x = \{index(z) : z \in S_x\} \\ &= \prod_{z \in S_x} (e(g,g)^{s \cdot q_z(0)})^{\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} (e(g,g)^{s \cdot q_{parent(z)}(index(z))})^{\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} e(g,g)^{s \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)} \\ &= e(g,g)^{s \cdot q_x(0)} \end{aligned}$$

- Starting from the root node r, call $DecryptNode(C, DK, r)$ and work recursively.

If attributes of the ciphertext satisfies the tree $\mathcal{T}$, then $DecryptNode(C, DK, r) = e(g,g)^{ys} = Y^s$. Since $C' = MY^s$, message M can be recovered simply dividing, $C'/Y^s = M$.

Remark 3.1. It is possible to improve efficiency of the decryption algorithm. Because as it is described till this point, the number of pairings to decrypt will always be as large as the number of the nodes in $\mathcal{T}$. To make this process more optimal, we can run a pre-process before making any cryptographic computations. In this pre-process, algorithm would discover which nodes are not satisfied. Then, when performing cryptographic computations, it does not include these nodes.

3.2.2 Security

We know discuss the security of the KP-ABE scheme given above. Goyal et al. [15] defines a selective-set model to prove security of the said scheme under chosen plaintext attack.

Selective-Set Model

Assume $\mathcal{A}$ is the adversary and $\mathcal{B}$ is the challenger.

Init $\mathcal{A}$ chooses the set of attributes, w, to be challenged upon.
Setup $\mathcal{B}$ runs the Setup algorithm and resulting public parameters, PP, are given to $\mathcal{A}$.
Phase 1 $\mathcal{A}$ is permitted to issue multiple inquiries for private keys for many access structures $\mathbb{A}_j$, as long as $w \notin \mathbb{A}_j$ for all j.
Challenge $\mathcal{A}$ submits messages M_0 and M_1 such that length of M_0 is equal to length of M_1. $\mathcal{B}$ randomly selects a bit $b \in \{0, 1\}$, and encrypts M_b with w. Resulting ciphertext C is given to $\mathcal{A}$.
Phase 2 Phase 1 is done again.
Guess The adversary outputs a guess b'.

The advantage of $\mathcal{A}$ in this game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 3.3 ([15]). If advantage in the Selective-Set game given above for all polynomial time adversaries are at most negligible then a KP-ABE scheme is secure in the Selective-Set model of security.

Goyal et al. [15] stated that the security of this scheme in the Selective-Set model reduces to the hardness of the d-BDH assumption given in Definition 2.5.

Theorem 3.2 ([15]). *If an adversary can break the scheme in the Selective-Set Model, then a simulator can be constructed to play the d-BDH game with a non-negligible advantage.*

Proof of the Theorem 3.2 can be seen in [15].

3.2.3 Efficiency

- In encryption algorithm, the number of exponentiations will be linear with number of attributes in the w, because of the $\{C_i\}_{i \in w}$ where $C_i = T_i^s$.
- Efficiency of decryption algorithm already been discussed in Subsection 3.2.1.
- The number of elements in the user's private key grows linearly with the number of leaf nodes in $\mathcal{T}$, because $i = att(x)$ means that i is a leaf node where $DK = (\mathcal{T}, \{D_x\}_{i=att(x)})$.
- The number of elements in PP grows linearly with the size of the attribute universe, i.e. $|\mathcal{U}|$, because $PP = (T_1, T_2, \ldots, T_n, Y)$ where $T_i = g^{t_i}$ for $\forall i \in \mathcal{U}$.
- The number of elements in the ciphertext, grows linearly with the number of attributes in the set w, because $C = (w, C' = MY^s, \{C_i\}_{i \in w})$.

3.2.4 Secure Cloud Storage System Suitability Analysis

Now we look at if [15] KP-ABE scheme is suitable to use in Kamara-Lauter cryptographic cloud storage scheme according to the conditions given in Section 1.1.

- Since there is a large universe construction given in [15], we can say that it is compatible with global scale construction. Although it is important to note that this scheme is not scalable.
- There is fine grained access control, provided by access tree.
- Decryption is inefficient, although ways to improve efficiency of decryption given in Subsection 3.2.1
- In this scheme there is one trusted party which is responsible for key generation. If we use this scheme in the cryptographic cloud storage system this trusted party most likely reside in the storage provider, then confidentiality property is compromised because storage provider has access to the unique keys.

3.2.5 KP-ABE with Non-monotonic Access Structure

Unlike monotonic access structures, non-monotonic access structures use **NOT** gates in addition to **AND**, **OR** and threshold gates. A scheme for KP-ABE with Non-monotonic access structures was proposed by Ostrovsky, Sahai and Waters in [20]. We will not examine this scheme further because of the inefficiencies that come with using negative attributes.

Assume there are several positive and negative attributes in the ciphertext, but negative attributes are useless in describing the ciphertext while raising ciphertext overhead. In large universe constructions, such as global scale secure cloud storage systems, storing the such ciphertexts will be problematic.

3.3 Ciphertext Policy Attribute Based Encryption

In this section, we examine a CP-ABE scheme Bethencourt, Sahai and Waters introduced in [3]. In this scheme, a users have private keys that are affiliated with a set of attributes. These attributes are expressed as strings. While encrypting a message M, user A needs to establish an access structure over attributes. Then user B is able to decrypt the ciphertext if and only if B's attributes satisfy the ciphertext's access structure.

This CP-ABE scheme comprise of four main algorithms and an optional algorithm (Delegate):

1. **Setup** is an algorithm that takes an implicit security parameter κ as an input and it gives the public parameters denoted by PP and a master key denoted by MK as outputs.

2. **Encryption** is an algorithm that takes public parameters PP, a message M and access structure $\mathbb{A}$, that is defined over attribute universe $\mathcal{U}$, as inputs. It gives the ciphertext CT, which also contains access structure $\mathbb{A}$, as output.

3. **Key Generation** is an algorithm that takes the master key MK and a set of attributes S inputs. It gives a private decryption key DK as outputs.

4. **Decryption** algorithm takes public parameters PP, ciphertext CT and a private decryption key DK for a attribute set S as inputs. If attribute set S satisfies $\mathbb{A}$ then decryption will be successful, then the algorithm gives message M as output. Otherwise, decryption will fail.

5. **Delegate** algorithm takes a private decryption key DK for a attribute set S and a set $\tilde{S} \subseteq S$ as inputs. It gives a secret key $\tilde{DK}$ for the set of attributes $\tilde{S}$ as outputs.

3.3.1 Construction of the CP-ABE scheme [3]

In this scheme, same as the FIBE scheme discussed in Section 3.1 and KP-ABE scheme disccussed in Section 3.1, Bethencourt et al. consider the prime order bilinear map definition given in Definition 2.4 and Lagrange Coefficient given in Definition 2.2.

Let the size of the groups G and G_T that are defined in Definition 2.4 be determined by a security parameter, κ.

In this scheme, Bethencourt et al. employ access structures defined in Definition 2.1, where attributes represent the parties and access structure $\mathbb{A}$ consists of attributes, and access tree $\mathcal{T}$ defined in KP-ABE scheme Definition 3.2.

Finally, in this scheme Bethencourt et al. also use a hash function $H : \{0,1\}^* \rightarrow G$ which is a random oracle that maps each attribute to a random element of G.

Setup ($\kappa \rightarrow PP, MK$): Following steps are performed in this algorithm.

- Choose a bilinear group G with $G = |p|$ where p is a prime order and $G = \langle g \rangle$.
- Choose $\alpha, \beta \in \mathbb{Z}_p$ as two random values.
- Compute $h = g^{\beta}$, $f = g^{1/\beta}$ and g^{α}.

Then the public parameters is published as:

$$PP = (G, g, h, f, e(g,g)^{\alpha})$$

And the master key is:

$$MK = (\beta, g^{\alpha})$$

Encrypt ($PP, M, \mathcal{T} \rightarrow CT$): To encrypt a message $M \in G_T$ under access tree $\mathcal{T}$ following steps are performed in the Encryption algorithm.

- Choose a polynomial denoted by q_x for every node x of the tree $\mathcal{T}$. Starting from the root node r, these polynomials are chosen as follows:
  - For each node x, $deg(q_x) = d_x$.
  - For root node r; set $q_r(0) = y$ and $d_r = k_r - 1$, then define polynomial q_r randomly.
  - For a node x that is not a root node; set $q_x(0) = q_{parent(x)}(index(x))$ and $d_x = k_x - 1$, then define polynomial q_x randomly.
- Compute $\tilde{C} = Me(g, g)^{\alpha y}$, $C = h^y$
- Let, W be set of leaf nodes in $\mathcal{T}$. Then compute the values $\forall w \in W\ C_w : g^{q_w(0)}$, $C'_w = H(att(w))^{q_w(0)}$.

Then ciphertext CT is:

$$CT = (\mathcal{T}, \tilde{C}, C, \{C_w\}_{\forall w \in W}, \{C'_w\}_{\forall w \in W})$$

Key Generation ($MK, S \rightarrow DK$): To create a private decryption key for the attribute set S following steps are performed in the Key Generation algorithm.

- Choose $r \in \mathbb{Z}_p$ as a random value.
- For every attribute s in the attribute set S, choose $r_s \in \mathbb{Z}_p$.
- Compute $D = g^{(\alpha + r)/\beta}$.
- For $\forall s \in S$, compute $D_s = g^r \cdot H(s)^{r_s}$ and $D'_s = g^{r_s}$.

Then private decryption key is:

$$DK = (D, \{D_s\}_{\forall s \in S}, \{D'_s\}_{\forall s \in S})$$

.

Delegate ($DK, \tilde{S} \rightarrow \tilde{DK}$): $\tilde{S}$ is an attribute set such that $\tilde{S} \subseteq S$ then following steps are performed in the delegate algorithm.

- Choose random values $\tilde{r}$ and $\tilde{r}_k \in \tilde{S}$.
- Compute $\tilde{D} = Df^{\tilde{r}}$.
- For $\forall k \in \tilde{S}$ compute $\tilde{D}_k = D_k g^{\tilde{r}} \cdot H(k)^{\tilde{r}_k}$ and $\tilde{D}'_k = D'_k g^{\tilde{r}_k}$.

Resulting private decryption key for the attribute set $\tilde{S}$ is:

$$\tilde{DK} = (\tilde{D}, \{\tilde{D}_k\}_{\forall k \in \tilde{S}}, \{\tilde{D}'_k\}_{\forall k \in \tilde{S}})$$

Decryption (CT, DK): The ciphertext CT is decrypted with the key DK in a recursive manner.

- Define a recursive algorithm $DecryptNode(CT, DK, x)$.
  - If x is a leaf node and $i = att(x)$ where $i \in S$ then:

$$\begin{aligned} DecryptNode(CT, DK, x) &= \frac{e(D_i, C_x)}{e(D'_i, C'_x)} \\ &= \frac{e(g^r \cot H(i)^{r_i}, h^{q_x(0)})}{e(g^{r_i}, H(i)^{q_x(0)})} \\ &= e(g, g)^{r q_x(0)} \end{aligned}$$

    If $i \notin S$ then $DecryptNode(CT, DK, x) = \perp$

  - If x is a non-leaf node, then $DecryptNode(CT, DK, x)$ then:
    - For each node z where $parent(z) = x$ call $DecryptNode(CT, DK, z)$ and store the outputs in F_z.
    - Let S_x be a set of child nodes such that $F_z \neq \perp$ and $|S_x| = k_x$.
      - If no such and $k_x = 0$ then the node is not satisfied and $DecryptNode(CT, DK, x) = \perp$.

· Otherwise,

$$
\begin{aligned}
F_x &= \prod_{z \in S_x} {F_z}^{\Delta_{i,S'_x}(0)},\ i = index(z),\ S'_x = \{index(z) : z \in S_x\} \\
&= \prod_{z \in S_x} (e(g,g)^{r \cdot q_z(0)})^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} (e(g,g)^{r \cdot q_{parent(z)}(index(z))})^{\Delta_{i,S'_x}(0)} \\
&= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)} \\
&= e(g,g)^{r \cdot q_x(0)}
\end{aligned}
$$

Then return $F_x = e(g,g)^{r \cdot q_x(0)}$

- If the tree $\mathcal{T}$ is satisfied by S, set $A = DecryptNode(CT, DK, r) = e(g,g)^{rq_r(0)} = e(g,g)^{ry}$.
- Finally, decrypt CT by:

$$\tilde{C}/(e(C,D)/A) = \tilde{C}/(e(h^y, g^{(\alpha+r)/\beta})/e(g,g)^{ry}) = M$$

As the Decryption algorithm of the KP-ABE scheme, it is possible to improve efficiency of the decryption algorithm of this CP-ABE scheme.

3.3.2 Security

We know discuss the security of the CP-ABE scheme given above. Bethencourt et al. [3] defines a security model to prove security of the said scheme under.

Security Model for CP-ABE

Assume $\mathcal{A}$ is the adversary and $\mathcal{B}$ is the challenger.

Init $\mathcal{A}$ chooses an access structure, $\mathbb{A}^*$ to be challenged upon.
Setup $\mathcal{B}$ runs the Setup algorithm and gives the resulting public parameters, PP, to the adversary $\mathcal{A}$.
Phase 1 $\mathcal{A}$ can issue multiple inquiries for private keys corresponding to the sets of

attributes $S_1, \ldots, S_{q_1}$, as long as none of these sets satisfies $\mathbb{A}^*$.

Challenge $\mathcal{A}$ submits messages M_0 and M_1 such that length of M_0 is equal to length of M_1. $\mathcal{B}$ randomly selects a bit $b \in \{0, 1\}$, and encrypts M_b with $\mathbb{A}^*$. Resulting ciphertext CT is given to $\mathcal{A}$.

Phase 2 Repeat Phase 1 with sets of attributes $S_{q_1+1}, \ldots, S_q$ while the same restrictions still in place.

Guess $\mathcal{A}$ outputs a guess b' about which message is encrypted.

The advantage of $\mathcal{A}$ in this game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 3.4. A CP-ABE scheme is secure in the Security Model for CP-ABE if all polynomial time adversaries have at most a negligible advantage.

While proving the security of this CP-ABE scheme, Bethencourt et al. use the generic bilinear group model of [7] and the random oracle model of [2]. They claim that no efficient adversary that acts generically on the groups G, G_T can break this scheme with any reasonable probability.

Definition 3.5 (The generic bilinear group model [7]). Consider two random encodings ψ, ψ_T of the additive group $\mathbb{F}_p$, i.e. injective maps $\psi, \psi_T : \mathbb{F}_p \to \{0, 1\}^m$ where $m > 3log(p)$. $G = \{\psi(x) : x \in \mathbb{F}_p\}$ and $G_T = \{\psi_T(x) : x \in \mathbb{F}_p\}$. We are given oracles to compute the induced group action on G, G_T and an oracle to compute a non-degenerate bilinear map $e : G \times G \to G_T$. A random oracle that serves as a hash function H is given to us. We refer G as a generic bilinear group.

Theorem 3.3 ([3]). *Let ψ, ψ_T, G, G_T be defined as above. For any adversary $\mathcal{A}$, let q be a bound on the total number of group elements it receives from inquiries it makes to the oracles for the hash function, groups G, G_T, bilinear map e and from its interaction with the CP-ABE security game. Then the advantage of adversary $\mathcal{A}$ in CP-ABE security game is $\mathcal{O}(q^2/p)$.*

Proof of the Theorem 3.3 can be seen in [3].

Note that, this is a relatively weak security model. Later, Waters presented three more CP-ABE schemes in [25] with non-interactive assumptions: Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption gicen in Definition 2.7, Decisional

Bilinear Diffie-Hellman (d-BDH) Assumption given in Definition 2.5 and q-Bilinear Diffie-Hellman Exponent Assumption gicen in Definition 2.8.

3.3.3 Efficiency

- In encryption algorithm, for each leaf node of the access tree $\mathcal{T}$, two exponentiations are needed. (Let, W be set of leaf nodes in $\mathcal{T}$. $\forall w \in W\ C_w = g^{q_w(0)}$, $C'_w = H(att(w))^{q_w(0)}$)
- In key generation algorithm, two exponentiations are required for each attribute of the user. ($s \in S$, $D_s = g^r \cdot H(s)^{r_s}$ and $D'_s = g^{r_s}$)
- The ciphertext CT has two group elements for each leaf of the tree $\mathcal{T}$. (Let, W be set of leaf nodes in $\mathcal{T}$. $\{C_w\}_{\forall w \in W}$, $\{C'_w\}_{\forall w \in W}$)
- Private key DK has two group elements for every attribute in S. ($\{D_s\}_{\forall s \in S}$, $\{D'_s\}_{\forall s \in S}$)
- Efficiency of the decryption algorithm largely depends on the construction of the access tree $\mathcal{T}$ as it is. It can require two pairings for every leaf of $\mathcal{T}$ that is matched with a private key attribute and one exponentiation algorithm visits during its path to such nodes.

3.3.4 Secure Cloud Storage System Suitability Analysis

Now we look at if [3] CP-ABE scheme is suitable to use in Kamara-Lauter cryptographic cloud storage scheme according to the conditions given in Section 1.1.

- Large universe construction is possible. Although it is important to note that this scheme is not scalable.
- There is fine grained access control.
- Decryption is inefficient, because decryption algorithm visits every node of access tree $\mathcal{T}$, whether it is necessary or not.

- In this scheme there is one trusted party which is responsible for key generation. If we use this scheme in the cryptographic cloud storage system this trusted party most likely reside in the storage provider, then confidentiality property is compromised because storge provider has access to the unique keys.

3.4 Multi-Authority Attribute Based Encryption

In this section, we will first look at Multi-Authority Attribute Based Encryption (MABE) scheme proposed by Chase in [11], then will examine the improved MABE scheme proposed by Chase and Chow [12].

MABE scheme, proposed by Chase in [11], is based pn the FIBE scheme proposed by Sahai and Waters [21]. In this scheme, Chase introduced two techniques to realize a multi-authority scheme. These techniques are:

- **Global Identifier (GID):** This scheme uses GID's so that, no user can claim another user's identifier and all attribute authorities can verify a user's identifier.
- **Central Authority (CA):** CA is used to provide synchronization between attribute authorities. It should be noted that, CA holds master key of the whole system and it is a trusted authority. Essentially, CA can decrypt any message encrypted with this scheme.

We will not examine this scheme any further because existence of a central authority with capabilities given above creates a discrepancy with the one of the main purposes of a multi-authority scheme, which are providing scalability and eliminating central trusted authority. In regards to suitability with cloud storage system; existence of CA also compromises confidentiality property of a cryptographic cloud storage system.

Chase and Chow [12] later improved Chase's earlier work [11] and proposed a MABE scheme without the central authority. In addition to removing central authority Chase and Chow also aimed to design a scheme with user privacy.

In this scheme there are multiple attribute authorities (AA), multiple users and a set of public parameters available for anyone.

- If a user wants to decrypt a message, they go to attribute authorities, prove that they are authorized to subset of attributes controlled by each authority and ask for the decryption keys that corresponds to these attributes. The authorities separately use the key generation algorithm and give back the decryption keys they get to the user.

- If a user wants to encrypt a message, they use the public parameters and a attribute set they choose in the encryption algorithm.

It is assumed that all attribute sets can be divided into N disjoint sets, controlled by the N different AA.

This MABE scheme made up of four algorithms:

1. **Setup,** is an algorithm that takes security parameter κ and number of authorities N as inputs. It gives system parameters $params$ which includes the threshold values $\{d_k\}_{k\in\{1,\ldots,N\}}$, public key pk and secret key sk pairs (pk_k, sk_k) for each attribute authority $k \in \{1, \ldots, N\}$ as outputs.

2. **Key Generation,** is an algorithm that takes attribute authority k's secret key sk_k, users GID, attribute set handled by the authority k, $\mathbb{A}_k$, as inputs. It gives a decryption key for the user with identity GID that corresponds to the attribute set $\mathbb{A}_k$.

3. **Encryption** algorithm takes set of attributes $\mathbb{A}_k$ and a message M as inputs. It gives ciphertext C as output.

4. **Decryption** algorithm takes the decryption keys and ciphertext C as inputs. ıf user has a sufficient set of decryption keys for each authority k, then decrypts C and gives M as an output. Otherwise, decryption fails.

We will not examine this scheme any further either. Because, even though need for central authority is eliminated, setting the number of attribute authorities to N from the very start and not being able to increase it later on shows that this scheme is not scalable which is one of main purposes of designing a multi-authority scheme.

3.5 Multi-Authority Ciphertext Policy Attribute Based Encryption Scheme

In this section, we examine Multi-Authority CP-ABE scheme introduced by Lewko and Waters [18]. In this scheme we employ Access Structure given in Definition 2.1, Linear Secret Sharing Scheme (LSSS) given in Definition 2.13 and Composite Order Bilinear Groups with order $N = p_1p_2p_3$ given in Definition 2.12.

This Multi-Authority CP-ABE scheme comprises of five algorithms:

1. **Universal Setup**, algorithm takes in the security parameter κ as an input and it gives the universal public parameters UP as output.

2. **Authority Setup**, is an algorithm that each authority runs separately. It takes UP as an input and gives authority's secret key that is denoted by SK and public key that is denoted by PK as outputs.

3. **Encryption**, is an algorithm that takes in message M, an access matrix $\mathbb{A}$ that has n rows and l columns with mapping ρ denoted by $(\mathbb{A}, \rho)$, universal public parameters UP and the set of public keys of other authorities $\{PK\}$ as inputs. It gives the ciphertext C as output.

4. **Key Generation**, algorithm takes an identifier ID, the universal public parameters UP, an attribute i that is controlled by an authority and the secret key SK for the same authority as inputs. It gives a key for this attribute-identifier pair $\{i, ID\}$, denoted by $K_{i,ID}$, as output.

5. **Decryption**, is an algorithm that takes in universal public parameters UP, the ciphertext C and group of keys that correspond to the i, ID pairs all with the same identifier ID. If group of attributes i satisfies $\mathbb{A}$ that corresponds to the ciphertext, the algorithm gives message M as an output.

Definition 3.6 (**[18]**). A multi-authority CP-ABE is said to be correct if whenever UP, C (obtained from encryption of M), $\{K_{i,ID}\}$ for a set of attributes satisfying the access structure of the ciphertext, $Decrypt(C, UP, \{K_{i,ID}\}) = M$

Different from schemes we have examined so far, this scheme uses a composite order bilinear group. Specifically, it uses bilinear group G with $|G| = N = p_1p_2p_3$ where

p_1, p_2, p_3 three distinct primes. G has subgroups $G_{p_1}, G_{p_2}, G_{p_3}$ with $|G_{p_1}| = p_1$, $|G_{p_2}| = p_2$ and $|G_{p_3}| = p_3$. In this scheme, all system is enclosed in the subgroup G_{p_1} except for the random oracle *H*. *H* maps identifiers to random group elements of G. Subgroups are only used for the proof of security of this scheme which employs the dual system encryption technique.

In the dual system there are two types of keys and two types of ciphertexts: normal and semi functional. While normal keys and ciphertexts has elements from the subgroup of order p_1, semi-functional keys and ciphertexts has elements from the subgroups of order p_2 and p_3. Then we can say that, semi-functional space is formed by the subgroups G_{p_2} and G_{p_3} and they are orthogonal to the subgroup G_{p_1}.

To provide collusion resistance, this scheme uses the universal identifier (ID). Collusion resistance makes use of the linear reconstruction property (given in Definition 2.14) and works as follows:

- To blind the message M, encryption algorithm uses $e(g_1, g_1)^s$ where s is a random value in $\mathbb{Z}_N$ and g_1 is a generator of G_{p_1}.

- Then s is split into shares λ_x according to the LSSS matrix and value 0 is split into shares w_x.

- In order to find the blinding factor $e(g_1, g_1)^s$, decryptor must acquire the shares of s. To do that decrpytor must pair key elements $K_{i,ID}$ with the ciphertext elements. To do this, decryptor will use the terms of the form $e(g_1, \boldsymbol{H}(ID))^{w_x}$.

- If the set of keys decryptor has satisfies the access structure, these extra terms $e(g_1, \boldsymbol{H}(ID))^{w_x}$ will cancel out, because w_x's are the shares of 0.

Note that, if two different users with identifiers ID_1 and ID_2 try to collude; terms $e(g_1, \boldsymbol{H}(ID_1))^{w_{x_1}}$ and $e(g_1, \boldsymbol{H}(ID_2))^{w_{x_2}}$ will not cancel out. Therefore, blinding factor $e(g_1, g_1)^s$ can not be recovered and decryptor can not successfully decrypt the ciphertext.

Construction of Multi-Authority CP-ABE Scheme [18]

Let G be a bilinear group of composite order $|G| = N = p_1p_2p_3$ where p_1, p_2, p_3 are distinct primes. System is enclosed in the subgroup G_{p_1} in G. Subgroups G_{p_2} and G_{p_3} of G are used in the security proof, which is done using dual system encryption technique.

Universal Setup $(\lambda \rightarrow UP)$**:** A bilinear group G with $|G| = N = p_1p_2p_3$ is chosen. Universal public parameters $UP = (N, g_1)$ is published. Description of the hash function $H : \{0,1\}^* \rightarrow G$, where H is a random oracle that maps every ID to an element of G is also published.

Authority Setup $(UP \rightarrow PK, SK)$**:** Assume Authority Setup algorithm is ran by authority O. For each attribute i controlled by the authority O, O chooses random elements $\alpha_i, y_i \in \mathbb{Z}_N$, then;

- Publishes public key as, $PK = \{e(g_1, g_1)^{\alpha_i}, g_1^{y_i} \ : \ \forall i\}$.
- Keeps the secret key, $SK = \{\alpha_i, y_i \ : \ \forall i\}$

Every authority runs Authority Setup algorithm.

Encryption $(M, (\mathbb{A}, \rho), UP, \{PK\} \rightarrow C)$**:** Encryption algorithm first chooses a random value $s \in \mathbb{Z}_N$. Then a random vector $v \in \mathbb{Z}_N^l$ is chosen up to the constraint that v has s as its first entry. Where $\mathbb{A}_x$ is the x^{th} row of $\mathbb{A}$, $\mathbb{A}_x \cdot v$ is denoted by λ_x. Then a random vector $w \in \mathbb{Z}_N^l$ is chosen up to the constraint that it has 0 as its first entry. Let w_x denote $\mathbb{A}_x \cdot w$. Finally it chooses $r_x \in \mathbb{Z}_N$ for each row $\mathbb{A}_x$ of $\mathbb{A}$. After all elements are chosen, following computations will be made:

$$C_0 = Me(g_1, g_1)^s$$

$$C_{1,x} = e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha_{\rho(x)} r_x}, \ : \ \forall x$$

$$C_{2,x} = g_1^{r_x}, \ \ \forall x$$

$$C_{3,x} = g_1^{y_{\rho(x)} r_x} g_1^{w_x}, \ : \ \forall x$$

Then ciphertext C is, $C = \{C_0, \{C_{1,x}\}_{\forall x}, \{C_{2,x}\}_{\forall x}, \{C_{3,x}\}_{\forall x}\}$

Key Generation $(ID, i, SK, UP \rightarrow K_{i,ID})$**:** Assume user with a identity ID wants to create a key for their ID and attribute i that belongs to an authority O. Then authority O computes:

$$K_{i,ID} = g_1^{\alpha_i} H(ID)^{y_i}$$

Then sends $K_{i,ID}$ to the user who has the universal identifier ID.

Decryption $(C, \{K_{i,ID}\}, UP \rightarrow M)$**:** Before performing decryption steps on the ciphertext C, decryptor needs to compute the hash function $H(ID)$ first. If decryptor has $\{K_{\rho(x),ID}\}$ for a subset of rows $\mathbb{A}_x$ of $\mathbb{A}$ such that $(1, 0, \ldots, 0)$ is in the span of these rows, then for each such x decryptor does the following computations:

$$C_{1,x} \cdot e(H(ID), C_{3,x}) / e(K_{\rho(x),ID}, C_{2,x}) = e(g_1, g_1)^{\lambda_x} e(H(ID), g_1)^{w_x}$$

Then the constant $c_x \in \mathbb{Z}_N$ is chosen up to the constraint that $\sum_x c_x \mathbb{A}_x = (1, 0, \ldots, 0)$ by the decryptor. Then following computations are made:

$$\prod_x (e(g_1, g_1)^{\lambda_x} e(H(ID), g_1)^{w_x})^{c_x} = e(g_1, g_1)^s$$

Note that, this computations work because $\lambda_x = \mathbb{A}_x \cdot v$ where $v \cdot (1, 0, \ldots, 0) = s$ and $w_x = \mathbb{A}_x \cdot w$ where $w \cdot (1, 0, \ldots, 0) = 0$. Finally M can be obtained by performing one final division as: $C_0 / e(g_1, g_1)^s = M \cdot e(g_1, g_1)^s / e(g_1, g_1)^s = M$

3.5.1 Security

We now discuss the security of the Multi-Authority CP-ABE scheme given above. Lewko and Waters gave the following game played by a challenger and an adversary. In this game they assume adversary can corrupt authorities only statically and make queries adaptively.

Multi-Authority CP-ABE Security Game: $Game_{Real}$

Assume $\mathcal{A}$ is the adversary. Let the set of authorities be denoted by S and the universe of attributes be denoted by $\mathcal{U}$. Each attribute is controlled by only one authority and every authority can be responsible for multiple attributes.

Setup: First universal setup is completed and universal public parameters UP are published. Then attacker $\mathcal{A}$ specifies a set of corrupt authorities S' such that $S' \subseteq S$. The challenger employs the authority setup to acquire public key-secret key pairs, i.e. (PK, SK), for every non-corrupt authority in $S - S'$. Then gives these public keys he collected to $\mathcal{A}$.

Key Query Phase 1: $\mathcal{A}$ submits attribute-identity pairs (i, ID), where attribute i is controlled by a non-corrupt authority, to the challenger. The challengers responds by giving the corresponding key $K_{i,ID}$ to the $\mathcal{A}$.

Challenge: $\mathcal{A}$ specifies an access matrix $(\mathbb{A}, \rho)$ and messages M_0, M_1 and. Let subset of rows of $\mathbb{A}$, which are labeled with attributes belonging to corrupt authorities, is denoted by V. Let subset of rows of $\mathbb{A}$ which are labeled with by attributes i $\mathcal{A}$ has queried pairs (i, ID) in the Key Query Phase 1, denoted by V_{ID} for each ID. For every ID, it is required that $V \cup V_{ID}$ not include $(1, 0, \ldots, 0)$. The challenger must be given public keys of corrupt authorities who controls at least one attribute appear in the labeling ρ by attacker $\mathcal{A}$.

Key Query Phase 2: $\mathcal{A}$ submits additional key queries (i, ID) that does not breach the constraints on the access matrix $(\mathbb{A}, \rho)$.

Guess: $\mathcal{A}$ announces a guess b'.

$\mathcal{A}$'s advantage in this game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 3.7 ([18]). A multi-authority CP-ABE scheme is secure against static corruption of authorities if all polynomial time adversaries have at most a negligible advantage in the security game given above.

Complexity Assumptions [18]

Lewko and Waters defined complexity assumptions to prove security of their multi-authority CP-ABE scheme. These assumptions designed for a bilinear group G with $|G| = N = p_1 p_2 p_3$ where p_1, p_2, p_3 are distinct primes defined in Definition 2.12.

While defining these assumptions, we will use the notations given below:

- $G_{p_1p_2}$ denotes the subgroup of G with $G_{p_1p_2} = p_1p_2$.
- $g_1 \xleftarrow{R} G_{p_1}$ means that g_1 is chosen as a random generator of G_{p_1}.
- $T_1 \xleftarrow{R} G$ means that T_1 is chosen as a random generator of G.

Assumption 3.4 (Subgroup Decision Problem for 3 Primes)**.** *Given a group generator $\mathcal{G}$, define following distribution:*

$$\mathbb{G} = (N, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$g_1 \xleftarrow{R} G_{p_1}$$

$$D = (\mathbb{G}, g_1)$$

$$T_1 \xleftarrow{R} G, \;\; T_2 \xleftarrow{R} G_{p_1}$$

The advantage of an algorithm $\mathcal{B}$ in breaking Assumption 3.1 defined as;

$$Adv1_{\mathcal{G},\mathcal{B}}(\lambda) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|$$

T_1 can be written uniquely as a product of; an element of G_1 called G_{p_1} part of T_1, an element of G_2 called G_{p_2} part of T_1 and an element of G_3 called G_{p_3} part of T_1.

Definition 3.8. $\mathcal{G}$ satisfies Assumption 3.1 if for any polynomial algorithm $\mathcal{B}$ $Adv1_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ.

Assumption 3.5. *Given a group generator $\mathcal{G}$, define following distribution:*

$$\mathbb{G} = (N, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$g_1, X_1 \xleftarrow{R} G_{p_1}, \; X_2 \xleftarrow{R} G_{p_2}, \; g_3 \xleftarrow{R} G_{p_3}$$

$$D = (\mathbb{G}, g_1, g_3, X_1X_2)$$

$$T_1 \xleftarrow{R} G_{p_1}, \;\; T_2 \xleftarrow{R} G_{p_1p_2}$$

The advantage of an algorithm $\mathcal{B}$ in breaking Assumption 3.2 defined as;

$$Adv2_{\mathcal{G},\mathcal{B}}(\lambda) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|$$

Definition 3.9. $\mathcal{G}$ satisfies Assumption 3.2 if for any polynomial algorithm $\mathcal{B}$ $Adv2_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ.

Assumption 3.6. *Given a group generator $\mathcal{G}$, define following distribution:*

$$\mathbb{G} = (N, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$g_1, X_1 \xleftarrow{R} G_{p_1},\ Y_2 \xleftarrow{R} G_{p_2},\ X_3, Y_3 \xleftarrow{R} G_{p_3}$$

$$D = (\mathbb{G}, g_1, X_1X_3, Y_2Y_3)$$

$$T_1 \xleftarrow{R} G_{p_1p_2},\ \ T_2 \xleftarrow{R} G_{p_1p_3}$$

The advantage of an algorithm $\mathcal{B}$ in breaking Assumption 3.3 defined as;

$$Adv3_{\mathcal{G},\mathcal{B}}(\lambda) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|$$

Definition 3.10. $\mathcal{G}$ satisfies Assumption 3.3 if $Adv3_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ for any polynomial algorithm $\mathcal{B}$.

Assumption 3.7. *Given a group generator $\mathcal{G}$, define following distribution:*

$$\mathbb{G} = (N, G, G_T, e) \xleftarrow{R} \mathcal{G}$$

$$g_1 \xleftarrow{R} G_{p_1},\ g_2 \xleftarrow{R} G_{p_2},\ g_3 \xleftarrow{R} G_{p_3},\ a, b, c, d \xleftarrow{R} \mathbb{Z}_N$$

$$D = (\mathbb{G}, g_1, g_2, g_3, g_1^a, g_1^b g_3^b, g_1^c, g_1^{ac} g_3^d)$$

$$T_1 \xleftarrow{R} G_{p_1p_2},\ \ T_2 \xleftarrow{R} G_{p_1p_3}$$

The advantage of an algorithm $\mathcal{B}$ in breaking Assumption 3.4 defined as;

$$Adv4_{\mathcal{G},\mathcal{B}}(\lambda) = |Pr[\mathcal{B}(D, T_1) = 1] - Pr[\mathcal{B}(D, T_2) = 1]|$$

Definition 3.11. $\mathcal{G}$ satisfies Assumption 3.4 if $Adv4_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ for any polynomial algorithm $\mathcal{B}$.

Security Definition [18]

Lewko and Waters prove security of their scheme using a form of dual system encryption technique. The proof of this scheme uses a hybrid argument over a sequence of games where challenge normal ciphertexts and normal keys are changed to be semi-functional.

Now we give definition of sequence of games defined in [18] by Lewko and Waters. In these definitions, we will assume an attribute i can belong to only one authority, i.e. the row labeling ρ of the $(\mathbb{A}, \rho)$ must be injective.

- Real security game is denoted by $Game_{Real}$.
- $Game_{Real'}$ differs from $Game_{Real}$ only as the random oracle H is defined as $H : \{0,1\}^* \rightarrow G_{p_1}$ instead of $H : \{0,1\}^* \rightarrow G$.

In addition to terms from subgroup G_{p_1}, semi-functional ciphertexts also contain terms from subgroups G_{p_2} and G_{p_3}. In addition terms from subgroup G_{p_1}, semi-functional keys of Type 1 contain terms from G_{p_2} and semi-functional keys of Type 2 contains terms from G_{p_3}. So if we try to decrypt a semi-functional ciphertext with semi-functional key of Type 1, G_{p_2} terms of ciphertext and key will be paired with each other which will prevent successful decryption. Similarly if we try to decrypt a semi-functional ciphertext with semi-functional key of Type 2, G_{p_3} terms of ciphertext and key will be paired with each other which will prevent successful decryption.

While describing the semi-functional ciphertexts and keys, for each attribute i first random values $z_i, t_i \in \mathbb{Z}_N$ are fixed and these values do not differ for different users. B denotes the subset of rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities. $\overline{B}$ denotes the subset of rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities.

Semi-functional ciphertext created by performing following steps:

- Obtain normal ciphertext, $C_0, C_{1,x}, C_{2,x}, C_{3,x}$ for every x by running the encryption algorithm.
- Random vectors $u_2, u_3 \in \mathbb{Z}_N^l$ are chosen then δ_x and σ_x defined as $\delta_x = \mathbb{A}_x \cdot u_2$, $\sigma_x = \mathbb{A}_x \cdot u_3$ for every row $\mathbb{A}_x$ of $\mathbb{A}$.
- For every $\mathbb{A}_x \in \overline{B}$, random exponents $\gamma_x, \psi_x \in \mathbb{Z}_N$ are chosen.

Then semi-functional ciphertexts formed as:

$$\left.\begin{aligned} C'_{1,x} &= C_{1,x} \\ C'_{2,x} &= C_{2,x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} \\ C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} \end{aligned}\right\} \forall x \; s.t. \mathbb{A}_x \in \overline{B}$$

$$\left.\begin{aligned} C'_{1,x} &= C_{1,x} \\ C'_{2,x} &= C_{2,x} \\ C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x} \end{aligned}\right\} \forall x \; s.t. \mathbb{A}_x \in B$$

Key that belongs to the user with identity ID, is actually a collection of values $H(ID), \{K_{i,ID}\}_{\forall i \in J}$ where J is the set of attributes that belong to non-corrupt authority requested by the attacker through Key Query Phase 1 and Key Query Phase 2 of the game.

For an ID, semi-funtional keys can be two types: Type 1 and Type 2. Before semi-functional key for an ID is created, assume $H(ID) \in G_{p_1}$ and choose $c \xleftarrow{R} \mathbb{Z}_N$.

Semi-functional key of Type 1 can be obtained by following steps:

- Before Semi-functional key of Type 1 is created we first need to define: $H(ID)' = H(ID)g_2^c$.
- Secondly we need to create a normal key $K_{i,ID}$.
- Lastly Semi-functional key of Type 1 is set as, $K'_{i,ID} = K_{i,ID} g_2^{cz_i}$.

Semi-functional key of Type 2 can be obtained by following steps:

- Before Semi-functional key of Type 2 is created we first need to define: $H(ID)' = H(ID)g_3^c$.
- Secondly we need to create a normal key $K_{i,ID}$.
- Lastly Semi-functional key of Type 1 is set as, $K'_{i,ID} = K_{i,ID} g_3^{ct_i}$

Let q be the number of identities ID for which the attacker has queried for the pair $K_{i,ID}$. We define $Game_0$, $Game_{j,1}$ and $Game_{j,2}$ for each j from 1 to q and $Game_{Final}$ as follows:

- $Game_0$ is similar to $Game_{Real'}$, except that a semi-functional ciphertext is given to the attacker.
- $Game_{j,1}$ is similar to $Game_0$, except that,
  - For the first $j-1$ identities that are queried, semi-functional keys of Type 2 are received.
  - For the j^{th} queried identity is semi-functional key of Type 1 is received.
  - After the first j queried identities, remaining keys that are received are normal.
- $Game_{j,2}$ is similar to $Game_0$, except that,
  - For the first j queried identities, semi-functional keys of Type 2 are received.
  - After the first j queried identities, remaining keys that are received are normal.

  Notice that, in $Game_{q,2}$, all received keys are semi-functional keys of Type 2.
- In $Game_{Final}$, where all keys are semi-functional keys of Type 2, and the ciphertext is a semi-functional ciphertext of a random message. Note that the attacker has advantage 0 in this game.

With following Lemmas, Lewko and Waters show these games are indistinguishable. The proof of security relies on the limit that ρ is injective, i.e. each attribute is used at only once in the $\mathbb{A}$.

Lemma 3.8 ([18]). *Assume there is a polynomial time algorithm $\mathcal{B}$ thus $Game_{Real}Adv_{\mathcal{B}}$ $-Game_{Real'}Adv_{\mathcal{B}} = \epsilon$. A polynomial time algorithm $\mathcal{A}$ with advantage ϵ in breaking Assumption 3.4 can be constructed.*

Lemma 3.9 ([18]). *Assume there is a polynomial time algorithm $\mathcal{B}$ thus $Game_{Real'}Adv_{\mathcal{B}}$ $-Game_0Adv_{\mathcal{B}} = \epsilon$. A polynomial time algorithm $\mathcal{A}$ with advantage ϵ in breaking Assumption 3.4 can be constructed.*

Lemma 3.10 ([18]). *Assume there is a polynomial time algorithm $\mathcal{B}$ thus $Game_{j-1,2}Adv_{\mathcal{B}}$ $-Game_{j,1}Adv_{\mathcal{B}} = \epsilon$. A polynomial time algorithm $\mathcal{A}$ with advantage ϵ in breaking Assumption 3.5 can be constructed.*

Lemma 3.11 ([18]). *Assume there is a polynomial time algorithm $\mathcal{B}$ thus $Game_{j,1}Adv_{\mathcal{B}} - Game_{j,2}Adv_{\mathcal{B}} = \epsilon$. A polynomial time algorithm $\mathcal{A}$ with advantage ϵ in breaking Assumption 3.6 can be constructed.*

Lemma 3.12 ([18]). *Assume there is a polynomial time algorithm $\mathcal{B}$ thus $Game_{q,2}Adv_{\mathcal{B}} - Game_{Final}Adv_{\mathcal{B}} = \epsilon$. A polynomial time algorithm $\mathcal{A}$ with advantage ϵ in breaking Assumption 3.7 can be constructed.*

Proves of these lemmas can be seen in [18].

While executing dual system encryption technique to explain the security of their scheme, Lewko and Waters made some claims about which results will be achieved if decryption is performed with following key and ciphertext pairs;

- Normal keys and semi-functional ciphertexts
- Semi-functional keys and normal ciphertexts
- Semi-functional keys and semi functional ciphertexts

Claim 3.13. *Normal keys can decrypt semi-functional ciphertexts.*

Proof of Claim 3.13. Normal key for identity ID and attribute i:

$$K_{i,ID} = g_1^{\alpha_i} H(ID)^{y_i}$$

Semi-functional ciphertexts have two types.
(i) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities, i.e. $\rho(x)$ maps to the attribute i which belongs to a corrupt authority:

$$\left.\begin{aligned} C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \\ C'_{2,x} &= C_{2,x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} = g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} \\ C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} \end{aligned}\right\} \forall x$$

$\forall x$ such that $\mathbb{A}_x \in \overline{B}$.

Then if we perform the computations according to the first part of the decryption algorithm, we will get the following results:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H(ID), C'_{3,x})}{e(K_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID), g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}})}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}}, g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot \\
&\quad e(H(ID), g_2)^{\delta_x + \gamma_x (z_{\rho(x)} - y_{\rho(x)})} \cdot e(H(ID), g_3)^{\sigma_x + \psi_x (t_{\rho(x)} - y_{\rho(x)})} \cdot \\
&\quad e(g_1, g_2)^{-\gamma_x \alpha_{\rho(x)}} e(g_1, g_3)^{-\psi_x \alpha_{\rho(x)}}
\end{aligned}
$$

Remember that, in the Universal Setup of the construction hash function is defined as $H : \{0,1\}* \rightarrow G$. But while discussing security of their scheme hash function is defined as $H : \{0,1\}* \rightarrow G_{p_1}$ instead of $H : \{0,1\}* \rightarrow G$. So we know that $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$. By composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_2) = e(g_1, g_3) = e(H(ID), g_2) = e(H(ID), g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x}$$

(ii) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities:

$$
\left.
\begin{aligned}
C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \\
C'_{2,x} &= C_{2,x} = g_1^{r_x} \\
C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x}
\end{aligned}
\right\} \forall x \; s.t. \mathbb{A}_x \in B
$$

Then if we perform decryption with this normal key and semi-functional ciphertext, we will get:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H(ID), C'_{3,x})}{e(K_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID), g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x})}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}}, g_1^{r_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(H(ID), g_2)^{\delta_x} \cdot e(H(ID), g_3)^{\sigma_x}
\end{aligned}
$$

Since $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(H(ID), g_2) = e(H(ID), g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x}$$

Finally we need to perform the computations of the second part of the decryption algorithm, i.e. $\prod_x (C_x)^{c_x}$. Since there are no extra elements, for both cases, decryption will be successful. So claim 1 is true. □

Claim 3.14. *Semi-functional keys can decrypt normal ciphertexts.*

Proof of Claim 3.14. Components of the normal ciphertext are:

$$\left.\begin{aligned} C_0 &= Me(g_1, g_1)^s \\ C_{1,x} &= e(g_1, g_1)^{\lambda_x} e(g_1, g_1)^{\alpha_{\rho(x)} r_x} \\ C_{2,x} &= g_1^{r_x} \\ C_{3,x} &= g_1^{y_{\rho(x)} r_x} g_1^{w_x} \end{aligned}\right\} \forall x$$

There are two types of semi-functional keys; Type 1 and Type 2.
(i) Consider Type 1 Semi-functional Key.

$$K'_{i,ID} = K_{i,ID} g_2^{cz_i} = g_1^{\alpha_i} H(ID)^{y_i} g_2^{cz_i}$$

where $H'(ID) = H(ID) g_2^c$.

Then if we perform first part of the decryption algorithm with normal ciphertext and Type 1 Semi-functional Key, we will get:

$$\begin{aligned} C_x &= \frac{C_{1,x} \cdot e(H'(ID), C_{3,x})}{e(K'_{\rho(x),ID}, C_{2,x})} \\ &= \frac{e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} e(g_1, g_2)^{cy_{\rho(x)} r_x + cw_x}}{e(g_1, g_2)^{cz_{\rho(x)} r_x}} \\ &= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_1, g_2)^{cw_x} \cdot e(g_1, g_2)^{cr_x (y_{\rho(x)} - z_{\rho(x)})} \end{aligned}$$

Since $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_2) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x}$$

Finally we need to complete decryption by calculating $\prod_x (C_x)$ since there are no extra elements decryption will be successful as $\prod_x (C_x)^{c_x} = e(g_1, g_1)^s$ and $C_0 / e(g_1, g_1)^s = M$.

(ii) Consider Type 2 Semi-functional Key.

$$K'_{i,ID} = K_{i,ID} g_3^{ct_i} = g_1^{\alpha_i} H(ID)^{y_i} g_3^{ct_i}$$

where $H'(ID) = H(ID) g_3^c$.

Then if we perform first part of the decryption with normal ciphertext and Type 2 Semi-functional Key, we will get:

$$\begin{aligned} C_x &= \frac{C_{1,x} \cdot e(H'(ID), C_{3,x})}{e(K'_{\rho(x),ID}, C_{2,x})} \\ &= \frac{e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} e(g_1, g_3)^{c y_{\rho(x)} r_x + c w_x}}{e(g_1, g_3)^{ct_{\rho(x)} r_x}} \\ &= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_1, g_3)^{c w_x} \cdot e(g_1, g_3)^{c r_x (y_{\rho(x)} - t_{\rho(x)})} \end{aligned}$$

Since $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x}$$

Like we did before, finally we need to complete decryption by calculating $\prod_x (C_x)^{c_x}$ since there are no extra elements decryption will be successful as $\prod_x (C_x)^{c_x} = e(g_1, g_1)^s$ and $C_0 / e(g_1, g_1)^s = M$.

Decryption is successful for both cases, then we can conclude that Claim 3.14 is true. □

Previously we explained that Lewko and Waters used hybrid argument over array of games, where ciphertext is challenged to be semi-functional then keys are changed to be semi-functional gradually. To prove that these games are not distinguishable, it must be certain that simulator cannot detect the form of the key being turned normal to semi-functional by trying to decrypt a semi-functional ciphertext.

Lewko and Waters claimed to prevent this by only allowing simulator to make a challenge ciphertext and key pairs that are nominally semi-functional, i.e. the key and ciphertext both have semi-functional elements that cancel each other out during the decryption.

Lewko and Waters made two following claims while discussing security of their scheme.

Claim 3.15. *When Type 1 semi-functional key is used to decrypt a semi-functional ciphertext, successful decryption is prevented by extra terms* $e(g_2, g_2)^{c\delta_x}$. *Apart from if the values* δ_x *are shares of* 0*, decryption will be successful because the semi-functional ciphertext is nominally semi-functional.*

Proof of Claim 3.15. Type 1 Semi-functional Key is:

$$K'_{i,ID} = K_{i,ID} g_2^{cz_i} = g_1^{\alpha_i} H(ID)^{y_i} g_2^{cz_i}$$

where $H'(ID) = H(ID) g_2^c$.
(i) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities:

$$\left.\begin{aligned} C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \\ C'_{2,x} &= C_{2,x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} = g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} \\ C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} \end{aligned}\right\} \forall x$$

$\forall x$ such that $\mathbb{A}_x \in \overline{B}$.

Then if we perform decryption with Type 1 semi-functional key of and semi-functional

ciphertext, we will get:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H'(ID), C'_{3,x})}{e(K'_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID) g_2^c, g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}})}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}} g_2^{c z_{\rho(x)}}, g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \\
&\cdot e(H(ID), g_2)^{\delta_x + \gamma_x \cdot (z_{\rho(x)} - y_{\rho(x)})} \cdot e(H(ID), g_3)^{\sigma_x + \psi_x \cdot (t_{\rho(x)} - y_{\rho(x)})} \\
&\cdot e(g_1, g_2)^{c w_x - \gamma_x \alpha_{\rho(x)} + c r_x (y_{\rho(x)} - z_{\rho(x)})} \cdot e(g_1, g_3)^{-\psi_x \alpha_{\rho(x)}} \\
&\cdot e(g_2, g_2)^{c \delta_x} \cdot e(g_2, g_3)^{c \sigma_x + c \psi_x (t_{\rho(x)} - z_{\rho(x)})}
\end{aligned}
$$

We know that $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_2) = e(g_1, g_3) = e(g_2, g_3) = e(H(ID), g_2) = e(H(ID), g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_2, g_2)^{c \delta_x}$$

(ii) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities:

$$
\left.
\begin{aligned}
C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x) r_x} + \lambda_x} \\
C'_{2,x} &= C_{2,x} = g_1^{r_x} \\
C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x}
\end{aligned}
\right\} \forall x \; s.t. \mathbb{A}_x \in B
$$

Then if we perform decryption with Type 1 semi-functional key and semi-functional ciphertext, we will get:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H'(ID), C'_{3,x})}{e(K'_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID) g_2^c, g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x}}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}} g_2^{c z_{\rho(x)}}, g_1^{r_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(H(ID), g_2)^{\delta_x} \cdot e(H(ID), g_3)^{\sigma_x} \\
&\cdot e(g_2, g_1)^{c w_x + c r_x (y_{\rho(x)} - z_{\rho(x)})} \cdot e(g_2, g_2)^{c \delta_x} \cdot e(g_2, g_3)^{c \sigma x}
\end{aligned}
$$

We know that $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_2) = e(g_2, g_3) = e(H(ID), g_2) = e(H(ID), g_3) = 1$where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_2, g_2)^{c\delta_x}$$

While, performing decryption with Type 1 semi-functional key and semi-functional ciphertext for both corrupt and non-corrupt authorities, even before performing the last step of the decryption, it is obvious that $e(g_2, g_2)^{c\delta_x}$ will prevent these decryptions from completing successfully. But if the values δ_x are shares of 0, then when we perform $\prod_x (C_x)^{c_x}$ terms $e(g_2, g_2)^{c\delta_x}$ will be eliminated, then it can be seen that these decryptions will be successful. So we can say that Claim 3.15 is true. □

Claim 3.16. *When Type 2 semi-functional key is used to decrypt a semi-functional ciphertext, successful decryption is prevented by extra terms $e(g_3, g_3)^{c\sigma x}$. Apart from if the values σ_x are shares of 0, decryption will be successful because the semi-functional ciphertext is nominally semi-functional.*

Proof of Claim 3.16. Semi-functional Key of Type 2 is:

$$K'_{i,ID} = K_{i,ID} g_3^{ct_i} = g_1^{\alpha_i} H(ID)^{y_i} g_3^{ct_i}$$

where $H'(ID) = H(ID) g_3^c$.

(i) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities:

$$\left.\begin{aligned} C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \\ C'_{2,x} &= C_{2,x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} = g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x} \\ C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}} \end{aligned}\right\} \forall x$$

$\forall x$ such that $\mathbb{A}_x \in \overline{B}$.

Then if we perform decryption with Type 2 semi-functional key and semi-functional

ciphertext, we will get:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H'(ID), C'_{3,x})}{e(K'_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID) g_3^c, g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x + \gamma_x \cdot z_{\rho(x)}} \cdot g_3^{\sigma_x + \psi_x \cdot t_{\rho(x)}})}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}} g_3^{ct_{\rho(x)}}, g_1^{r_x} \cdot g_2^{\gamma_x} \cdot g_3^{\psi_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \\
&\cdot e(H(ID), g_2)^{\delta_x + \gamma_x \cdot (z_{\rho(x)} - y_{\rho(x)})} \cdot e(H(ID), g_3)^{\sigma_x + \psi_x \cdot (t_{\rho(x)} - y_{\rho(x)})} \\
&\cdot e(g_1, g_2)^{-\gamma_x \alpha_{\rho(x)}} \cdot e(g_1, g_3)^{c w_x - \gamma_x \alpha_{\rho(x)} + c r_x (y_{\rho(x)} - t_{\rho(x)}) - \psi_x \alpha_{\rho(x)}} \\
&\cdot e(g_2, g_3)^{c\delta_x + c\gamma_x (z_{\rho(x)} - t_{\rho(x)})} \cdot e(g_3, g_3)^{c\sigma_x}
\end{aligned}
$$

We know that $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_2) = e(g_1, g_3) = e(g_2, g_3) = e(H(ID), g_2) = e(H(ID), g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_3, g_3)^{c\sigma_x}$$

(ii) Semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities:

$$
\left.
\begin{aligned}
C'_{1,x} &= C_{1,x} = e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \\
C'_{2,x} &= C_{2,x} = g_1^{r_x} \\
C'_{3,x} &= C_{3,x} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x} = g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x}
\end{aligned}
\right\} \forall x \; s.t. \mathbb{A}_x \in B
$$

Then if we perform decryption with Type 2 semi-functional key and semi-functional ciphertext, we will get:

$$
\begin{aligned}
C_x &= \frac{C'_{1,x} \cdot e(H'(ID), C'_{3,x})}{e(K'_{\rho(x),ID}, C'_{2,x})} \\
&= \frac{e(g_1, g_1)^{\alpha_{\rho(x)} r_x + \lambda_x} \cdot e(H(ID) g_3^c, g_1^{w_x + r_x \cdot y_{\rho(x)}} \cdot g_2^{\delta_x} \cdot g_3^{\sigma_x})}{e(g_1^{\alpha_{\rho(x)}} H(ID)^{y_{\rho(x)}} g_3^{ct_{\rho(x)}}, g_1^{r_x})} \\
&= e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \\
&\cdot e(H(ID), g_2)^{\delta_x} \cdot e(H(ID), g_3)^{\sigma_x} \\
&\cdot e(g_1, g_3)^{c w_x + c r_x (y_{\rho(x)} - t_{\rho(x)})} \cdot e(g_2, g_3)^{c\delta_x} \cdot e(g_3, g_3)^{c\sigma_x}
\end{aligned}
$$

We know that $H(ID) \in G_{p_1}$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$ and $g_3 \in G_{p_3}$, by composite order bilinear group definition given in Definition 2.10 we know that $e(g_1, g_3) = e(g_2, g_3) =$

$e(H(ID), g_2) = e(H(ID), g_3) = 1$ where 1 denotes to the identity element of G_T. Then,

$$C_x = e(g_1, g_1)^{\lambda_x} \cdot e(H(ID), g_1)^{w_x} \cdot e(g_3, g_3)^{c\sigma_x}$$

While, performing decryption with semi-functional key of Type 2 for semi-functional ciphertext for both corrupt and non-corrupt authorities, even before performing the last step of the decryption, it is obvious that $e(g_3, g_3)^{c\sigma_x}$ will prevent these decryptions from completing successfully. But if the values σ_x are shares of 0, then when we perform $\prod_x (C_x)^{c_x}$ terms $e(g_3, g_3)^{c\sigma_x}$ will be eliminated, then it can be seen that these decryptions will be successful. So we can say that Claim 3.16 is true. □

Claim 3.17. *Semi-functional keys can not decrypt semi-functional ciphertexts.*

Proof of Claim 3.17. While we tried to prove Claim 3.15 and Claim 3.16, we have seen that,

- If we try to decrypt semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities with Semi-functional key of Type 1 decryption will fail (unless δ is shares of 0). (See Proof of Claim 3.15)
- If we try to decrypt semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities with Semi-functional key of Type 1 decryption will fail (unless δ is shares of 0). (See Proof of Claim 3.15)
- If we try to decrypt semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the corrupt authorities with Semi-functional key of Type 2 decryption will fail (unless σ is shares of 0). (See Proof of Claim 3.16)
- If we try to decrypt semi-functional ciphertext for rows of $\mathbb{A}$ that corresponds to the attributes belong to the non-corrupt authorities with Semi-functional key of Type 2 decryption will fail (unless σ is shares of 0). (See Proof of Claim 3.16)

Therefore, we can say that Claim 3.17 is true and semi-functional keys can not decrypt semi-functional ciphertexts. □

3.5.2 Efficiency

- In encryption algorithm, for every row x of $\mathbb{A}$ two exponentiations are needed.
- While generating a key for the a ID, Key generation algorithm performs one exponentiations for each attribute i that belongs to an authority.
- The number of group elements in authorities public key grows linearly with the number of attributes associated with the said authority.

3.5.3 Secure Cloud Storage System Suitability Analysis

Now we look at if [18] Multi Authority CP-ABE scheme is suitable to use in Kamara-Lauter cryptographic cloud storage scheme according to the conditions given in Section 1.1.

- Large universe construction is possible. This algorithm is scalable, because at any time new attribute authorities can added to the system.
- There is fine grained access control.
- Decryption is efficient.
- This scheme is decentralized, that is storage provider does not have access to the keys, so confidentiality property of secure cloud storage system is not compromised.

CHAPTER 4

MULTI-AUTHORITY CP-ABE SCHEME WITH PRIME ORDER BILINEAR GROUP SETTING

When we look at the Attribute Based Encryption schemes given in the previous chapter it is clear that most suitable scheme to use in secure cloud storage scheme access control is multi-authority CP-ABE scheme proposed by Lewko and Waters [18]. However, creating prime order variant of this system can lead to a more efficient system via faster group operations. In this Chapter, we will examine how such a system can be created.

In [14] Freeman identified two properties of composite order bilinear groups that are used in pairing based cryptosystems as "projecting property" and "canceling property". Out of these two properties Multi-Authority CP-ABE scheme proposed by Lewko and Waters and discussed in Section 3.5 relies on canceling property of the composite order bilinear maps.

Considering the composite order bilinear group definition given in Definition 2.10, since subgroups $G_{p_1}, G_{p_2}, \ldots, G_{p_m}$ of G are orthogonal for the bilinear map $e : G \times G \rightarrow G_T$ then for elements $t \in G_{p_x}$ and $s \in G_{p_y}$ $e(t, s) = 1$ where 1 denotes to the identity element of G_T as long as $x \neq y$. If we look specifically at scheme given in Section 3.5 and Definition 2.12 then we have $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$, $g_3 \in G_{p_3}$ and $e(g_1, g_2) = e(g_1, g_3) = e(g_2, g_3) = 1$ where 1 denotes to the identity element of G_T.

In [17] Lewko identified another property of composite order bilinear groups that is used in pairing based cryptosystems as "parameter hiding" and noted that this property is often used together with canceling property.

In multi-authority CP-ABE given in [18] Lewko and Waters used subgroup decision assumption defined over composite order bilinear groups. In [17] Lewko shows that effects of subgroup decision assumption can be imitated by subspace assumption over prime order bilinear groups given in Definition 2.16. Then showed that subspace assumption is implied by the Decisional Linear Assumption given in Definition 2.9. Then we can assume that, we can develop a prime order bilinear group version of Multi-Authority CP-ABE Scheme given in Section 3.5 using subspace assumption. For such scheme, we will need a further generalized version of the subspace assumption.

Definition 4.1 (**Generalized Subspace Assumption**)**.** Considering the groups and bilinear map definition given in Definition 2.4. Given a group generator $\mathcal{G}$ we define the following distribution:

$$\mathbb{G} = (p, G, G_T, e) \xleftarrow{R} \mathcal{G}, \quad (\mathbb{B}_1, \mathbb{B}_1^*), (\mathbb{B}_2, \mathbb{B}_2^*), \ldots, (\mathbb{B}_m, \mathbb{B}_m^*) \xleftarrow{R} Dual(\mathbb{Z}_p^n)$$

$$g \xleftarrow{R} G, \qquad \eta, \beta, \tau_1, \tau_2, \tau_3, \mu_1, \mu_2, \mu_3 \xleftarrow{R} \mathbb{Z}_p$$

$$\left.\begin{aligned} U_{1,i} &= g^{\mu_1 b_{1,i} + \mu_2 b_{(k+1),i} + \mu_3 b_{(2k+1),i}} \\ U_{2,i} &= g^{\mu_1 b_{2,i} + \mu_2 b_{(k+2),i} + \mu_3 b_{(2k+2),i}} \\ &\vdots \\ U_{k,i} &= g^{\mu_1 b_{k,i} + \mu_2 b_{2k,i} + \mu_3 b_{3k,i}} \end{aligned}\right\} \forall i \in [m]$$

$$\left.\begin{aligned} V_{1,i} &= g^{\tau_1 \eta b^*_{1,i} + \tau_2 \beta b^*_{(k+1),i}} \\ V_{2,i} &= g^{\tau_1 \eta b^*_{2,i} + \tau_2 \beta b^*_{(k+2),i}} \\ &\vdots \\ V_{k,i} &= g^{\tau_1 \eta b^*_{k,i} + \tau_2 \beta b^*_{2k,i}} \end{aligned}\right\} \forall i \in [m]$$

$$\left.\begin{aligned} W_{1,i} &= g^{\tau_1 \eta b^*_{1,i} + \tau_2 \beta b^*_{(k+1),i} + \tau_3 b^*_{(2k+1),i}} \\ W_{2,i} &= g^{\tau_1 \eta b^*_{2,i} + \tau_2 \beta b^*_{(k+2),i} + \tau_3 b^*_{(2k+2),i}} \\ &\vdots \\ W_{k,i} &= g^{\tau_1 \eta b^*_{k,i} + \tau_2 \beta b^*_{2k,i} + \tau_3 b^*_{3k,i}} \end{aligned}\right\} \forall i \in [m]$$

$$D = (\{g^{b_{1,i}}, g^{b_{2,i}}, \ldots, g^{b_{2k,i}}, g^{b_{(3k+1),i}}, \ldots, g^{b_{n,i}}, g^{\eta b^*_{1,i}}, \ldots, g^{\eta b^*_{k,i}}, g^{\beta b^*_{(k+1),i}}, \ldots, g^{\beta b^*_{2k,i}}\}_{\forall i \in [m]}$$

$$\{g^{b^*_{(2k+1),i}}, \ldots, g^{b^*_{n,i}}\}_{\forall i \in [m]}, \{U_{1,i}, U_{2,i}, \ldots, U_{k,i}\}_{\forall i \in [m]}, \mu_3)$$

We assume that for any probabilistic polynomial time algorithm $\mathcal{B}$ with output in $\{0, 1\}$,

$$Adv_{\mathcal{G},\mathcal{B}} = |Pr[\mathcal{B}(D, \{V_{1,i}, \ldots, V_{k,i}\}_{\forall i \in \mathcal{U}})] = 1 - Pr[\mathcal{B}(D, \{W_{1,i}, \ldots, W_{k,i}\}_{\forall i \in \mathcal{U}}) = 1]|$$

is negligible in security parameter κ.

Assume $\mathcal{U}$ denotes to the attribute universe and $|\mathcal{U}| = m$. For every attribute i in the universe $\mathcal{U}$, we should have dual bases involved in the subspace assumption and these bases can be denoted by;

$$(\mathbb{B}_1, \mathbb{B}^*_1), (\mathbb{B}_2, \mathbb{B}^*_2), \ldots, (\mathbb{B}_m, \mathbb{B}^*_m) \in Dual(\mathbb{Z}^{12}_p)$$

where $\mathbb{B}_i = (b_{1,i}, b_{2,i}, \ldots, b_{12,i})$ and $\mathbb{B}^*_i = (b^*_{1,i}, b^*_{2,i}, \ldots, b^*_{12,i})$. The subspace assumption with $n = 12$ and $k = 2$ and $|\mathcal{U}| = m$ dual orthogonal bases (one for each attribute i).

We can use $b^*_{1,i}, \ldots, b^*_{4,i}$ to create normal keys and normal cihertexts (Like we used G_{p_1} elements in scheme given in Section 3.5) and use them in the encryption and decryption processes. Then we can use $b^*_{5,i}, \ldots, b^*_{12,i}$ in creating semi-functional keys and semi-functional ciphertexts (Like we used G_{p_2}, G_{p_3} elements in scheme given in Section 3.5) and use them to show security of this scheme.

CHAPTER 5

CONCLUSION

Using cloud storage services can be beneficial for enterprises and government organizations. Because they can evade costs of having and maintaining private storage infrastructure while providing data sharing and availability, reliability and efficient retrieval of the data. However, because of the need for confidentiality and integrity of the data, these enterprises and government organizations need secure cloud storage services.

We can use attribute based encryption schemes to provide access control in such secure cloud storage systems. With ABE schemes, only users with necessary attributes can access the unique key that can decrypt the data.

There are many ABE schemes that were proposed over the years. While choosing an ABE scheme to use in a secure cloud storage system we need to make sure that, it is compatible with global scale construction and scalable, has fine grained access control, has efficient decryption and it maintains the confidentiality property of the cloud storage system.

After examining several ABE schemes, we came to the conclusion that multi-authority CP-ABE scheme presented by Lewko and Waters [18] provides these properties. But because these scheme uses composite order bilinear groups, we can improve of its performance by creating a version of it that uses prime order bilinear groups. We presented some thoughts on how such scheme can be designed in Chapter 4.

REFERENCES

[1] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, Ph.D. thesis, 1996.

[2] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in *Proceedings of the 1st ACM conference on Computer and communications security*, pp. 62–73, ACM, 1993.

[3] J. Bethencourt, A. Sahai, and B. Waters, Ciphertext-policy attribute-based encryption, in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 321–334, IEEE, 2007.

[4] D. Boneh and X. Boyen, Efficient selective-id secure identity-based encryption without random oracles, in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, 2004.

[5] D. Boneh and X. Boyen, Secure identity based encryption without random oracles, in *Annual International Cryptology Conference*, pp. 443–459, Springer, 2004.

[6] D. Boneh and X. Boyen, Efficient selective identity-based encryption without random oracles, Journal of Cryptology, 24(4), pp. 659–693, 2011.

[7] D. Boneh, X. Boyen, and E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 440–456, Springer, 2005.

[8] D. Boneh, X. Boyen, and H. Shacham, Short group signatures, in *Annual International Cryptology Conference*, pp. 41–55, Springer, 2004.

[9] D. Boneh and M. Franklin, Identity-based encryption from the weil pairing, in *Annual international cryptology conference*, pp. 213–229, Springer, 2001.

[10] R. Canetti, S. Halevi, and J. Katz, Chosen-ciphertext security from identity-based encryption, in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 207–222, Springer, 2004.

[11] M. Chase, Multi-authority attribute based encryption, in *Theory of Cryptography Conference*, pp. 515–534, Springer, 2007.

[12] M. Chase and S. S. Chow, Improving privacy and security in multi-authority attribute-based encryption, in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 121–130, ACM, 2009.

[13] C. Cocks, An identity based encryption scheme based on quadratic residues, in *IMA International Conference on Cryptography and Coding*, pp. 360–363, Springer, 2001.

[14] D. M. Freeman, Converting pairing-based cryptosystems from composite-order groups to prime-order groups, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 44–61, Springer, 2010.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Acm, 2006.

[16] S. Kamara and K. Lauter, Cryptographic cloud storage, in R. Sion, R. Curtmola, S. Dietrich, A. Kiayias, J. M. Miret, K. Sako, and F. Sebé, editors, *Financial Cryptography and Data Security*, pp. 136–149, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[17] A. Lewko, Tools for simulating features of composite order bilinear groups in the prime order setting, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 318–335, Springer, 2012.

[18] A. Lewko and B. Waters, Decentralizing attribute-based encryption, in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 568–588, Springer, 2011.

[19] T. Okamoto and K. Takashima, Hierarchical predicate encryption for inner-products, in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 214–231, Springer, 2009.

[20] R. Ostrovsky, A. Sahai, and B. Waters, Attribute-based encryption with non-monotonic access structures, in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 195–203, ACM, 2007.

[21] A. Sahai and B. Waters, Fuzzy identity-based encryption, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, 2005.

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

[23] A. Shamir, Identity-based cryptosystems and signature schemes, in *Workshop on the theory and application of cryptographic techniques*, pp. 47–53, Springer, 1984.

[24] B. Waters, Efficient identity-based encryption without random oracles, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 114–127, Springer, 2005.

[25] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in *International Workshop on Public Key Cryptography*, pp. 53–70, Springer, 2011.