2013

Privacy-preserving access control techniques in distributed systems

Jinguang Han
University of Wollongong

Privacy-Preserving Access Control Techniques in Distributed Systems

A thesis submitted in fulfillment of the
requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Jinguang Han

School of Computer Science and Software Engineering
March 2013

Dedicated to

My family

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Jinguang Han
March 2, 2013

Abstract

An access control scheme is designed to restrict users access to the protected data in distributed systems. To satisfy different access requirements, various access control schemes have been proposed. Nevertheless, the privacy problem in them has not been considered extensively, while it is a primary concern of network users. Hence, constructing access control schemes with a sound privacy protection is an important task.

The main contribution of this thesis is to propose privacy-preserving access control schemes in the following three aspects. First, we design access control schemes where the contents required by users are protected against any proxy servers or other parties. We develop two identity-based data storage schemes, which are secure against collusion attacks. In these schemes, a user can access one of the data outsourced by the owner if he has obtained an access permission from the owner. A proxy server can transfer a ciphertext for the owner to a ciphertext for the requester without observing anything about the plaintext.

Second, we construct three access control schemes where users' personal sensitive information, such as access credentials, identities and attributes, can be protected. We develop two attribute-based access control schemes, each with distinctive features. The first scheme is a decentralized attribute-based encryption scheme where a user can obtain secret keys from multiple authorities without releasing anything about his/her identifier to them and furthermore, it is secure against collusion attacks. Multiple authorities can work independently without any cooperation. Especially, an authority can dynamically leave or add in the system without re-initializing the system and re-issuing secret keys to users. Further, the second scheme captures the feature that only the senders whose attributes satisfy the access structure specified by the receiver can send messages to him/her and only the receiver whose attributes satisfies the access structure published by the sender can obtain the protected data. Furthermore, we give a provable generic construction of dynamic single

sign-on schemes where a user can access multiple services using one credential and only the designated service providers can validate his credential.

Third, we develop several access control schemes where an authorized user can access the protected data without releasing anything about his personal sensitive information and the accessed contents to the database. We construct an attribute-based oblivious access control scheme by introducing an attribute-based encryption scheme with constant computation and communication cost to an oblivious transfer scheme. Furthermore, we design efficient oblivious transfer with access control schemes by introducing oblivious signature-based envelope schemes to an oblivious transfer scheme. In these schemes, an authorized user can access the protected data obliviously, while the database only knows the number of the data accessed by the user.

Notably, all schemes developed in this thesis are derived from cryptographic primitives and formally proven in the proposed security models under complexity assumptions.

Acknowledgement

I would like to express my sincere thanks to my supervisors Professor Willy Susilo, Professor Yi Mu and Dr Jun Yan, for their careful guidance, helpful suggestions and insights in my research. I appreciate the encouragement that Professor Susilo gave me at the beginning of my study, the first paper which Professor Mu carefully modified for me and the first report which Dr Yan discussed with me in detail. The experience of working with them is invaluable.

Acknowledgement should also be given to people who helped me in my research, including Dr Man Ho Au, Mr Fuchun Guo, Dr Xinyi Huang, Mr Nan Li, Mr Shams Qazi, Dr Mohammad Reza Reyhanitabar, Dr Pairat Thorncharoensri, Mr Vasilios Evangelos Tourloupis, Dr Guilin Wang, Mr Lei Wang, Mr Yang Wang, Dr Wei Wu, Dr Tsz Hon Yuen and Ms Miao Zhou.

I appreciate the referees of this thesis: Professor Colin Boyd and Professor Masahiro Mambo for their valuable comments and constructive suggestions to improve the quality of this thesis.

It is my honour to be a PhD student of Center for Computer and Information Security Research (CCISR), School of Computer Science and Software Engineering, University of Wollongong. I would like to thank University of Wollongong and Smart Service, Cooperative Research Center (CRC) for providing me scholarships.

Finally, I am sincerely grateful to my family for their love and encouragement. This thesis would have been impossible without their support.

Jinguang Han
March, 2013
Wollongong

Publications

This thesis is based on the following presented or published papers, which were finished when I was in pursuit of the PhD degree in University of Wollongong.

1. Jinguang Han, Willy Susilo, Yi Mu and Jun Yan. Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems*, 23(11): 2150-2162, 2012.

2. Jinguang Han, Willy Susilo, Yi Mu. Identity-based secure distributed data storage schemes. *IEEE Transactions on Computers*, 2013 (accepted on January 14, 2013).

3. Jinguang Han, Willy Susilo, Yi Mu and Jun Yan. Attribute-based oblivious access control. *The Computer Journal*, 55(10): 1202-1215, 2012.

4. Jinguang Han, Willy Susilo and Yi Mu. Identity-based data storage in cloud computing. *Future Generation Computer Systems*, 29(3): 673-681, 2013.

5. Jinguang Han, Willy Susilo, Yi Mu and Jun Yan. New constructions of OSBE schemes and their applications in oblivious access control. *International Journal of Information Security*, 11(6): 389-401, 2012.

6. Jinguang Han, Willy Susilo, Yi Mu and Jun Yan. Efficient oblivious transfers with access control. *Computers and Mathematics with Applications*, 63(4): 827-837, 2012.

7. Jinguang Han, Yi Mu, Willy Susilo and Jun Yan. A generic construction of dynamic single sign-on with strong security. In Sushil Jajodia and Jianying Zhou, editors, Proceedings: *6th International ICST Conference on Security and Privacy in Communication Networks - SecureComm 2010*, volume 50 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and*

Telecommunications Engineering, pages 181-198, Singapore, September 7-9 2010. Springer.

Papers which are under review:

8. Jinguang Han, Willy Susilo, Yi Mu and Jun Yan. Attribute-based data transfer with filtering scheme in distributed systems. (submitted)

List of Notions

The following notations are used throughout this thesis. Some special notations will be defined when they are first used.

ℓ	A security parameter;
1^ℓ	The string of ℓ ones;
$\forall$	For all;
$\exists$	There exists;
$\mathbb{Z}$	The set of integers;
$\mathbb{Z}^+$	The set of positive integers;
$\mathbb{Z}_p$	The set consists of the integers modulo p;
$\mathbb{Z}_p^*$	The multiple group of integers modulo p;
$\|\mathbf{A}\|$	The cardinality of the set A;
$\epsilon(\ell)$	A negligible function on ℓ;
$a\|\|b$	The concatenation of the string a and the string b;
$\Pr[\mathbf{A}]$	The probability of the event $\mathbf{A}$ occurring;
$\mathbf{A} \bigcup \mathbf{B}$	The union of sets $\mathbf{A}$ and $\mathbf{B}$;
$\mathbf{A} \bigcap \mathbf{B}$	The intersection of sets $\mathbf{A}$ and $\mathbf{B}$;
$\mathbf{A} - \mathbf{B}$	The difference of sets $\mathbf{A}$ and $\mathbf{B}$;
$\mathbf{A} \subseteq \mathbf{B}$ ($\mathbf{A} \subset \mathbf{B}$)	The set $\mathbf{A}$ is a (proper) subset of the set $\mathbf{B}$;
$\mathcal{KG}(1^\ell)$	A key generation algorithm;
$a \xleftarrow{R} \mathbf{A}$	a is selected from $\mathbf{A}$ uniformly at random if $\mathbf{A}$ is a finite set;
$f(x) \xleftarrow{R} \mathbb{Z}_p[x]$	$f(x)$ is selected from $\mathbb{Z}_p[x]$ at random;
$\mathcal{X} \xrightarrow{s} \mathcal{Y}$	The party $\mathcal{X}$ sends an element s to the party $\mathcal{Y}$;
$\mathcal{X} \xleftarrow{s} \mathcal{Y}$	The party $\mathcal{Y}$ sends an element s to the party $\mathcal{X}$;
$\mathbf{A}(x) \rightarrow y$	y is computed by running the algorithm $\mathbf{A}$ on input x;
$a \pmod b$	The remainder of a divided by b;
$a \in \mathbf{A}$ ($a \notin \mathbf{A}$)	a is (not) in the set $\mathbf{A}$.

List of Abbreviations

The following abbreviations are used throughout this thesis. Some special abbreviations will be defined when they are first used.

ABAC	Attribute-based Access Control;
ABE	Attribute-based Encryption;
AC-OT	Oblivious Transfer with Access Control;
CDH	Computational Diffie-Hellman;
DBDH	Decisional Bilinear Diffie-Hellman;
DDH	Decisional Diffie-Hellman;
DoS	Denial-of-service;
EU-CMA	Existentially Unforgeable under Chosen-message Attacks;
FIdM	Federated Identity Management;
IND-CCA2	Indistinguishability against Adaptive Chosen Ciphertext Attacks;
IND-CPA	Indistinguishability against Adaptive Chose Plaintext Attacks;
OT	Oblivious Transfer;
PKE	Public Key Encryption;
PKG	Private Key Generator;
PKC	Public-Key Cryptography;
PoK	Proof of Knowledge;
PPT	Probabilistic Polynomial Time;
PRE	Proxy Re-Encryption;
SDH	Strong Diffie-Hellman;
SSO	Single Sign-on;

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In open communication environments, providing confidentiality to sensitive data is one of the most fundamental problems that have attracted a lot of attention. Access control is an essential component in communication. Considering different application scenarios and requirements, access control schemes with distinctive features have been proposed, such as discretionary access control [DAC87], mandatory access control [Osb97] and role-based access control [SCFY96]. An access control scheme should provide the following properties: authentication, authorization and accountability (AAA) [VCF+00].

- Authentication. Authentication is a procedure where a trusted party can confirm whether a user is the entity which he claims to be. Generally, a user must use his private information to convince the trusted party that he is the real entity.
- Authorization. Authorization is a procedure where a user's access privilege is determined. Whether a user can access the protected data depends on whether the specified access policy can be satisfied.
- Accountability. Accountability is a procedure where what a user has done is recorded. It is used to address the appropriate use of data and identify the users who misuse the data.

1.1 Background

1.1.1 Public-Key Cryptography

Cryptography as a primitive has been used to provide secure communications among multiple parties. In a public-key cryptographic system, each user has two keys: one

is called as secret key and the other as public key. Being different from the secret-key cryptography (symmetric cryptography) where these two keys are identical or it is easy to compute one from the other, in a public-key cryptographic scheme, it is infeasible to compute the secret key from the public key. Therefore, the public key can be known by all the users in the system. The distinguishing property of public-key cryptography (PKC) is that two parties can initialize a private conversation without any prior communication. Hence, since its seminal introduction by Diffie and Hellman [DH76], PKC has attracted much attention. More details about PKC can be found in books [MVO96, Mao03].

The main task of PKC is to provide two properties: privacy and authentication [DH76]. Privacy means that the transmitted message should only be retrieved by the intended receivers. Meanwhile, authentication means that all the participants in a conversation are legal and authorized.

Currently, there are two main research directions on PKC: public-key encryption and digital signature.

- Public-key Encryption. In a public-key encryption (PKE) scheme, a sender can encrypt a message under the receiver's public key directly as the public key is publicly known, and send the ciphertext to the receiver. Consequently, the receiver can use his secret key to decrypt the ciphertext and obtain the plaintext. Some classic PKE schemes, to name a few, are ElGamal encryption scheme [ElG85], RSA encryption scheme [RSA78] and Cramer-Shoup encryption scheme [CS98].

- Digital Signature. A digital signature is the electronic version of a handwritten signature. It is a public-key cryptographic protocol where a user can generate a signature on a message using his signing key and the validity of the signature can be verified by anyone. Notably, any other user cannot forge a signature on behalf of the real signer. Hence, a digital signature scheme can provide non-repudiation property which is especially necessary in digital certificates. Some classic digital signature schemes, to name a few, are RSA signature scheme [RSA78], Schnorr signature scheme [Sch90] and digital signature standards (DSS) [DSS94].

1.1.2 Privacy Protection

Privacy issues are the primary concern to network users as the network allows the collection of vast amount of personal information which users release when they access the network [CP02]. Several schemes have been proposed to solve these issues [CDN09, CGH09, Au09, Koh10]. In these schemes, the following three problems were considered: hiding accessed contents, hiding personal information and hiding both accessed contents and personal information.

- Hiding Accessed Contents. We say that a scheme can hide the accessed contents if it is impossible for the database or proxy servers to know the contents which an authorized user has accessed. Cryptographic primitives which can potentially provide this property are oblivious transfer (OT) [Rab81, NP99a, NP99b, AIR01, CNS07] and proxy re-encryption scheme (PRE) [BBS98, CH07, LV08]. In a t-out-of-n OT (OT_t^n) protocol, by an interaction, a receiver can access t-out-of-n services; while the database only know the number of accessed services without knowing anything about the contents. Whereas, in a PRE scheme, a proxy server can transfer a ciphertext for the owner to ciphertext for the receiver without seeing the original plaintext if he has obtained an access permission (re-encryption key) from the owner.

- Hiding Personal Information. We say that a scheme can hide personal information if it can protect users' sensitive information, such as ID card, PIN and credentials, to be collected, modified and disseminated. Cryptographic schemes which can be potential primitives to provide this property are accumulator schemes [BdM94, BP97, CL02], pseudonym systems [LRSW99, YK11], anonymous credential schemes [Cha85, CL01, CL02, BCC+09], blind signature schemes [Cha83, AO01, Oka06], group signature schemes [CH91, ACJT00, BBS04], ring signature schemes [RST01, BSS02, Nao02] and attribute-based systems [SW05, GPSW06, PTMW06, BSW07, OSW07], *etc.* In these schemes, a user can prove that he has obtained the credentials on his private information or he is a member of the group, instead of showing his credentials or membership certificates.

- Hiding Accessed Contents and Personal Information. We say that a scheme can hide both accessed contents and personal information if an authorized user can access the protected sources without releasing anything about his

private information and the accessed contents to the database or a proxy server. Schemes which can provide this property are oblivious transfer with access control (AC-OT) [CGH09, CDN09, ZAW+10].

To provide privacy protection, hiding personal information is not sufficient [IKOS06] as even if an adversary does not know who the user is, he can trace the user by his actions, such as the Websites which he visited, the medicine which he ordered online and communication societies which he attended. Therefore, to provide a sound solution to privacy protection, both the properties of hiding accessed contents and hiding personal information should be addressed.

1.2 Contributions of This Thesis

Privacy protection can be classified into three different types according to the security requirements: access contents protection, personal information protection and protection of accessed contents and personal information. We note that systems with strong privacy protection are more complicated. In this thesis, we mainly focus our attention on the protocols which are developed from cryptographic primitives and can be formally proven. The main contributions of this thesis are as as follows.

1. Accessed Contents Protection. In some databases, such as stock quotes, users are required to register with real personal information. However, their investment strategies will be revealed if their accessed contents are exposed. Therefore, it is important in these systems to protect the accessed content against being known by other parties.

   In an identity-based data storage scheme, the owner can encrypt his files under his identity and outsource them to a proxy server. If a user wants to access one of the encrypted files, he is required to obtain an access permission from the owner and send it to the proxy server. Then, the proxy server can use the access permission to transfer the encrypted file under the owner's identity to the encrypted file under the user's identity without seeing the file. As a result, the user can decrypt the ciphertext and obtain the file. Similarly, in an identity-base proxy re-encryption (ID-based PRE) scheme, a proxy server can transfer a ciphertext for the original decryptor to a ciphertext for a designated decryptor if he obtains a re-encryption key from the original decryptor.

Although ID-based PRE schemes have been proposed, they are not suitable to an identity-based data storage scheme. First, the receiver can cooperate with the proxy server to access all the owner's files if he has obtained an access permission. Second, these schemes are not secure against collusion attacks, namely the receiver can compute the owner's secret key if he can compromise the proxy server. Third, an access permission is decided by the owner with the help of the central authority, instead of the owner himself. Finally, they cannot provide inter-domain query. In this thesis, we propose two identity-based data storage schemes to solve these problems. In the first scheme, for one query, the receiver can only access one file of the owner, instead of all files. In this scheme, the owner can make an access permission independently without the help of the central authority and collusion attacks are resisted. Then, the second scheme can support not only intra-domain but also inter-domain queries. Notably, supporting inter-domain query is especially important in distributed systems, such as cloud computing, ad-hoc network, *etc.*

2. Personal Information Protection. Personalized services require users to register with distinct characters, such as attributes, roles, rights, *etc.* However, if the registered information is illegally distributed and collected, the user can be impersonated. Hence, controlling the release of personal information is important.

   In an attribute-based system, a user is identified by a set of descriptive attributes. A user can access the protected data if his attributes satisfy the specified access structure, while the database does not know the real identity of the receiver. In this thesis, we propose a privacy-preserving decentralized key-policy attribute-based encryption (ABE) scheme. In this scheme, a user can obtain secret keys from multiple authorities without releasing anything about his identifier to them. Therefore, even multiple authorities collaborate, they cannot trace the user by his identifer. Furthermore, multiple authorities can perform independently without any cooperation. However, in previous ABE schemes, there must be a central authority to issue secret keys to users or multiple authorities must cooperate to initial the system. Subsequently, we propose an attribute-based data transfer with filtering (ABDTF) scheme where a receiver can specify an access structure such that only the qualified senders can send messages to him. Prior to decrypting the encrypt messages,

the receiver can use the filtering scheme to filter out the false messages. Hence, this scheme can resist the denial-of-service (DoS) attacks. We also formalize the definition and security model for ABDTF schemes. Finally, we give a generic construction of dynamic single sign-on (SSO). We first give the formal definitions and security models for SSO and dynamic SSO. This makes an important step toward the formal research on SSO. In our construction, a user can access multiple services using one credential and change his service requirements dynamically without the necessity to re-initial the system and re-issue credentials. Note that only the designated service providers can validate the user's credential, while other service providers cannot know anything about the user's credential.

3. Protection of Accessed contents and Personal Information. In some sensitive database, such as DNA database, if a user's personal information can be linked to the accessed contents (DNA sequences), a lot of his/her information will be disclosed, such as the potential diseases, race, *etc.* Furthermore, the user will meet some problems, such as discrimination. Thereafter, in this systems, protecting both accessed contents and personal information are important.

   To provide privacy protection, schemes with hiding accessed contents and personal information have been proposed. However, there are some shortcomings. First, in some of these schemes, zero-knowledge proof must be used by a user to prove that he has been authorized to access the services. Second, the ciphertext in some of these schemes are linear with the number of the required attributes. In this thesis, we propose two schemes where both the accessed contents and the personal information can be protected. First, we propose an ABE scheme with a constant communication and computation cost. Then, based on this ABE scheme, we propose an attribute-based oblivious access control scheme, where an authorized user can access the protected services without releasing anything about his private information and the accessed contents to the database. While, the database only knows the number of the accessed services by the authorized user. Second, we introduce a primitive called oblivious signature-base envelope (OSBE) into an OT protocol to construct a new oblivious transfer with access control scheme without the need of zero-knowledge proof.

1.3 Thesis Organization

The remainder of this thesis is organized as follows.

In Chapter 2, we review the preliminaries required by this thesis. We introduce algebra knowledge, including group, field and bilinear group. Furthermore, we present some complexity assumptions used throughout this thesis. Subsequently, we describe some basic cryptographic primitives, including hash function, random oracle model, PKE, broadcast encryption, Waters's identity-based encryption, digital signature, commitment scheme and zero-knowledge proof.

In Chapter 3, we develop an identity-based data storage scheme which is secure against collusion attacks. We introduce the background about identity-based proxy re-encryption schemes and point out that they are not suitable to an identity-based data storage scheme. Then, we propose an identity-based data storage scheme. Furthermore, we improve the security of our scheme from being secure against the chosen plaintext attacks to be secure against the chosen ciphertext attacks. Finally, we prove the security of these schemes.

In Chapter 4, we propose an identity-based data storage scheme which can support both intra-domain and inter-domain queries. This scheme is secure against the collusion attacks and suitable to cloud computing.

In Chapter 5, we design a privacy-preserving decentralised key-policy ABE scheme. We first review the knowledge about multiple-authority ABE scheme and point out some problems in the current schemes. Then, we develop a decentralized ABE scheme where multiple authorities can perform independently without any collaboration. Furthermore, we propose a privacy-preserving key extract protocol for our decentralized ABE scheme. Finally, we prove the security of these schemes.

In Chapter 6, we construct an attribute-based data transfer with filtering (ABDTF) scheme. Firstly, we review the background concerning attribute-based data transfer schemes and show the main fault in them is that they cannot resist DoS attacks. Then, we develop an ABDTF scheme which is secure against the DoS attacks and prove its security.

In Chapter 7, we give a generic construction of dynamic single sign-on (SSO) scheme. First, we review the literature of SSO and find that there is no provable dynamic SSO scheme. Then, we give the formal definitions and security models for SSO and dynamic SSO. We show how to use cryptographic primitives to construct a provable dynamic SSO scheme. Finally, we formally prove the security of our

construction.

In Chapter 8, we propose an attribute-based oblivious access control (ABOAC) scheme. We first design an ABE scheme with constant communication and computation cost and prove its security. Then, an ABOAC scheme is constructed by introducing the proposed ABE scheme to an OT scheme. Finally, the security of the proposed ABOAC scheme is proven.

In Chapter 9, an efficient oblivious transfer with access control (AC-OT) scheme is constructed. First, we introduce the background about AC-OT schemes. Then, we introduce oblivious signature-based envelope (OSBE) schemes to an OT scheme to develop AC-OT schemes. Finally, we prove the security of our schemes.

Chapter 10 concludes this thesis.

Chapter 2

Preliminaries

We introduce the preliminaries used throughout this thesis, including foundations of algebra, complexity assumptions and cryptographical tools. More details of cryptography theory can be found in the following books [MVO96, Mao03].

2.1 Miscellaneous Notions

In this thesis, by ℓ, we denote a security parameter. By 1^ℓ, we denote the string of ℓ ones. We say that a function $\epsilon : \mathbb{Z} \to \mathbb{R}$ is negligible if for all $k \in \mathbb{Z}$, there exists $z \in \mathbb{Z}$ such that $\epsilon(x) \leq \frac{1}{x^k}$ for all $x > z$. Unless otherwise specified, by ϵ, we always denote a negligible function. If $n \in \mathbb{Z}^+$, by $[n]$, we denote the set of integers $\{1, 2, \cdots, n\}$. By $p(x) \xleftarrow{R} \mathbb{Z}_p[x]$, we denote the polynomial $p(x)$ is randomly selected from the polynomial ring $\mathbb{Z}_p[x]$ consisting of the polynomials that coefficients are from the finite field $\mathbb{Z}_p$.

A basic notion in complexity theory is probability distributions which should be computationally indistinguishable. Notably, two probability distributions which are statistically distinguishable are computationally distinguishable [Gol90].

Computational Indistinguishability. We say that two distribution families $\Omega_1(\ell)$ and $\Omega_2(\ell)$ are computationally indistinguishable if, for all PPT algorithms $\mathcal{A}$,

$$\left| \Pr_{x \in \Omega_1(\ell)}[\mathcal{A}(x) = 1] - \Pr_{x \in \Omega_2(\ell)}[\mathcal{A}(x) = 1] \right| \leq \epsilon(\ell).$$

Statistical Indistinguishability. We say that two distribution families $\Omega_1(\ell)$ and $\Omega_2(\ell)$ are statistically indistinguishable if

$$\sum_{z} \left| \Pr_{x \in \Omega_1(\ell)}[x = z] - \Pr_{x \in \Omega_2(\ell)}[x = z] \right| \leq \epsilon(\ell).$$

Unless otherwise specified, by indistinguishability, we mean that it is computationally indistinguishable.

2.2 Access Structure

An access structure is used to restrict users' access right. The formal definition of an access structure is as follows:

Definition 2.1 Access Structure [Bei96]. *Let $\mathbb{P} = \{P_1, P_2, \cdots, P_N\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_N\}}$ is monotonic, if $\mathbf{A} \in \mathbb{A}$ and $\mathbf{A} \subseteq \mathbf{B}$ implies that $\mathbf{B} \in \mathbb{A}$. An access structure (respectively, monotonic access structure) is a collection (respectively, monotonic collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \cdots, P_N\}$, namely $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_N\}} - \{\phi\}$. The sets in $\mathbb{A}$ are called as authorized sets, while the sets outside of $\mathbb{A}$ are called as unauthorized sets.*

2.3 Foundations of Algebra

In this section, we review the basic algebra knowledge: group, cyclic group and field.

2.3.1 Group

A group consists of a set of elements and an operation which is executed between any two elements in the set. The formal definition of a group is described as follows:

Definition 2.2 Group. *A group $(\mathbb{G}, \otimes)$ is a set $\mathbb{G}$ equipped with an operation $\otimes$, and satisfies the following properties:*

1. Closure. *For all $g, h \in \mathbb{G}$, $g \otimes h \in \mathbb{G}$;*
2. Associativity. *For all $g, h, \eta \in \mathbb{G}$, $(g \otimes h) \otimes \eta = g \otimes (h \otimes \eta)$;*
3. Identity. *There exists $1_{\mathbb{G}} \in \mathbb{G}$ called the identity of $(\mathbb{G}, \otimes)$, such that $1_{\mathbb{G}} \otimes g = g \otimes 1_{\mathbb{G}} = g$ for all $g \in \mathbb{G}$;*

4. Inverse. *For all $g \in \mathbb{G}$, there exists $g^{-1} \in \mathbb{G}$ called the inverse of g such that $g \otimes g^{-1} = g^{-1} \otimes g = 1_{\mathbb{G}}$.*

For simplicity, a group $(\mathbb{G}, \otimes)$ is often denoted as $\mathbb{G}$ when the operation $\otimes$ is clear. The number of the elements in $\mathbb{G}$ is called the order of $\mathbb{G}$ and denoted as $|\mathbb{G}|$. A group $\mathbb{G}$ is a finite group if $|\mathbb{G}|$ is finite; otherwise, it is an infinite group. A group $\mathbb{G}$ is an Abelian group if for all $g, h \in \mathbb{G}$, $g \otimes h = h \otimes g$.

Let $\mathcal{G}(1^\ell)$ be a group generator which takes as input 1^ℓ and outputs a group $\mathbb{G}$ with order p, namely $\mathcal{G}(1^\ell) \to (p, \mathbb{G})$.

Definition 2.3 Order of Group Element. *Suppose that $g \in \mathbb{G}$, the order of g in $\mathbb{G}$ is the least $i \in \mathbb{Z}^+$ such that $g^i = 1_{\mathbb{G}}$. If for all $i \in \mathbb{Z}^+$, $g^i \neq 1_{\mathbb{G}}$, the order of g is infinite. The order of g is denoted as $ord(g)$.*

Especially, if any element in a group $\mathbb{G}$ can be expressed by a specially element in $\mathbb{G}$, $\mathbb{G}$ is called as a cyclic group. The formal definition of a cyclic group is as follows:

Definition 2.4 Cyclic Group. *A group $\mathbb{G}$ is a cyclic group if there exists $g \in \mathbb{G}$, for all $h \in \mathbb{G}$, there exists $i \in \mathbb{Z}$ such that $h = g^i$. The element g is called as a generator of the group $\mathbb{G}$. $\mathbb{G}$ is said to be generated by g and denoted as $\mathbb{G} = \langle g \rangle$.*

2.3.2 Field

A field consists of a set of elements and two operations defined between any two elements in the set. The formal definition of a field is described as follows.

Definition 2.5 Field. *A field $(\mathbb{F}, \oplus, \otimes)$ consists of a set $\mathbb{F}$ and two operations: addition $\oplus$ and multiplication $\otimes$, and satisfies the following properties.*

1. Addition Group. *$(\mathbb{F}, \oplus)$ is an Abelian group. The identity of the group $(\mathbb{F}, \oplus)$ is denoted as $0_{\mathbb{F}}$ and called additive identity or zero-element;*

2. Multiplication Group. *Let $\mathbb{F}^* = \mathbb{F} - \{0_{\mathbb{F}}\}$. $(\mathbb{F}^*, \otimes)$ is an Abelian group. The identity of the group $(\mathbb{F}^*, \otimes)$ is denoted as $1_{\mathbb{F}}$ and called as multiplicative identity;*

3. Distributivity. *For all $g, h, \eta \in \mathbb{F}$, $(g \oplus h) \otimes \eta = (g \otimes \eta) \oplus (h \otimes \eta)$.*

2.4 Bilinear Groups

In this section, we review the knowledge related to bilinear group.

Definition 2.6 Bilinear Map [BF01]. *Suppose that $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\tau$ are three cyclic groups with the same order p. Let g and h be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear map (pairing) is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\tau$ satisfying the following properties :*

1. Bilinearity. *For all $x \in \mathbb{G}_1, y \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, $e(x^a, y^b) = e(x, y)^{ab}$.*
2. Non-degeneracy. *$e(g, h) \neq 1_{\mathbb{G}_\tau}$ where $1_{\mathbb{G}_\tau}$ is the identity of the group $\mathbb{G}_\tau$.*
3. Computability. *For all $x \in \mathbb{G}_1$ and $y \in \mathbb{G}_2$, there exists an efficient algorithm to compute $e(x, y)$.*

Definition 2.7 Bilinear Groups [GPS08]. *$\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_\tau$ constitute a bilinear group if there exists a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\tau$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_\tau| = p$.*

Galbraith, Paterson and Smart [GPS08] divided pairing operations used in cryptography into three types:

1. $\mathbb{G}_1 = \mathbb{G}_2$;
2. $\mathbb{G}_1 \neq \mathbb{G}_2$, there exists an efficiently computable homomorphism map $\psi : \mathbb{G}_1 \to \mathbb{G}_2$;
3. $\mathbb{G}_1 \neq \mathbb{G}_2$, there are no efficiently computable homomorphism maps between groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

We say that a pairing is symmetric if $\mathbb{G}_1 = \mathbb{G}_2$ and denote the symmetric bilinear group as $(e, p, \mathbb{G}_1, \mathbb{G}_\tau)$. Pairing is often constructed on suitable elliptic curves, so its efficiency is determined by the selected elliptic curves. When selecting elliptic curves for a pairing, two factors must be considered: the group size l of the elliptic curves and the embedding degree d. Generally, to achieve the security of $1,024$-bit RSA, the two parameters l and d should satisfy $l \times d \geq 1,024$ [Lyn06, Bro10].

In the rest of this thesis, we denote $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ as a bilinear group generator which takes as input 1^ℓ and outputs bilinear groups $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ with order p and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\tau$. We denote T_e and T_p as the time of executing one exponential and one pairing, respectively. By $E_{\mathbb{G}_1}$, $E_{\mathbb{G}_2}$, $E_{\mathbb{G}_\tau}$ and $E_{\mathbb{Z}_p}$, we denote the length of one element in the group $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_\tau$ and $\mathbb{Z}_p$, respectively.

2.5 Complexity Assumptions

In this section, we review the complexity assumptions used throughout this thesis.

2.5.1 Discrete Logarithm Assumption

The discrete logarithm (DL) assumption [Odl85] in a finite field is one of the basic assumptions in cryptography research. The DL assumption is defined as follows.

Definition 2.8 Discrete Logarithm (DL) Assumption [Odl85]. *Let* $\mathcal{G}(1^\ell) \rightarrow (p, \mathbb{G})$ *and* $\mathbb{G} = \langle g \rangle$. *Given* $(g, y) \in \mathbb{G}^2$, *we say that the discrete logarithm assumption holds on* $\mathbb{G}$ *if no PPT adversary* $\mathcal{A}$ *can compute a* $x \in \mathbb{Z}_p$ *such that* $y = g^x$ *with the advantage*

$$Adv_{\mathcal{A}}^{DL} = \Pr\left[y = g^x | \mathcal{A}(p, g, y, \mathbb{G}) \rightarrow x\right] \geq \epsilon(\ell)$$

where the probability is taken over the random choice of $y \in \mathbb{G}$ *and the bits consumed by the adversary* $\mathcal{A}$.

2.5.2 Computational Diffie-Hellman Assumption

Diffie and Hellman [DH76] proposed this assumption and constructed a key exchange scheme based on it. This assumption is defined as follows.

Definition 2.9 Computational Diffie-Hellman (CDH) Assumption [DH76]. *Let* $x, y \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{G}(1^\ell) \rightarrow (p, \mathbb{G})$ *and* $\mathbb{G} = \langle g \rangle$. *Given* (g, g^x, g^y), *we say that the computational Diffie-Hellman assumption holds on* $\mathbb{G}$ *if no PPT adversary* $\mathcal{A}$ *can compute* g^{xy} *with the advantage*

$$Adv_{\mathcal{A}}^{CDH} = \Pr\left[\mathcal{A}(g, g^x, g^y) \rightarrow g^{xy}\right] \geq \epsilon(\ell)$$

where the probability is taken over the random choices of $x, y \xleftarrow{R} \mathbb{Z}_p$ *and the bits consumed by the adversary* $\mathcal{A}$.

Maurer [Mau94] discussed the relationships between DL assumption and CDH assumption.

2.5.3 Decisional Diffie-Hellman Assumption

Boneh [Bon98] surveyed the various applications of decisional Diffie-Hellman assumption and demonstrated some results regarding it security.

Definition 2.10 Decisional Diffie-Hellman (DDH) Assumption [Bon98]. *Let* $x, y, z \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{G}(1^\ell) \to (p, \mathbb{G})$ *and* $\mathbb{G} = \langle g \rangle$. *Given* (g, g^x, g^y), *we say that the decisional Diffie-Hellman assumption holds on* $\mathbb{G}$ *if no PPT adversary* $\mathcal{A}$ *can distinguish* $(X, Y, Z) = (g^x, g^y, g^{xy})$ *from* $(X, Y, Z) = (g^x, g^y, g^z)$ *with the advantage*

$$Adv_{\mathcal{A}}^{DDH} = |\Pr[\mathcal{A}(X, Y, g^{xy}) = 1] - \Pr[\mathcal{A}(X, Y, g^z) = 1]| \geq \epsilon(\ell)$$

where the probability is taken over the random choices $x, y, z \xleftarrow{R} \mathbb{Z}_p$ *and the bits consumed by the adversary* $\mathcal{A}$.

2.5.4 Computational Bilinear Diffie-Hellman

Boneh and Franklin [BF01] introduced this assumption. This assumption is as follows.

Definition 2.11 Computational Bilinear Diffie-Hellman (CBDH) Assumption [BF01]. *Let* $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}, \mathbb{G}_\tau)$ *and* $\mathbb{G} = \langle g \rangle$. *We say that the computational bilinear Diffie-Hellman assumption holds on* $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ *if no PPT adversaries* $\mathcal{A}$ *can compute* $e(g, g)^{abc}$ *from* $(A, B, C) = (g^a, g^b, g^c)$ *with the advantage*

$$Adv_{\mathcal{A}}^{CBDH} = \Pr\left[\mathcal{A}(A, B, C) \to e(g, g)^{abc}\right] \geq \epsilon(\ell)$$

where the probability is taken over the random choices of $a, b, c \xleftarrow{R} \mathbb{Z}_p$ *and the bits consumed by* $\mathcal{A}$.

2.5.5 Decisional Bilinear Diffie-Hellman Assumption

Boneh and Franklin [BF01] introduced this assumption and used it to construct an identity-based encryption (IBE) scheme. This assumption is defined as follows.

Definition 2.12 Decisional Bilinear Diffie-Hellman (DBDH) Assumption [BF01]. *Let* $a, b, c, z \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}, \mathbb{G}_\tau)$ *and* $\mathbb{G} = \langle g \rangle$. *We say that the decisional bilinear Diffie-Hellman assumption holds on* $(p, e, \mathbb{G}, \mathbb{G}_\tau)$ *if no PPT adversary* $\mathcal{A}$ *can distinguish* $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ *from* $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ *with the advantage*

$$Adv_{\mathcal{A}}^{DBDH} = \left|\Pr[\mathcal{A}(A, B, C, e(g, g)^{abc}) = 1] - \Pr[\mathcal{A}(A, B, C, e(g, g)^z) = 1]\right| \geq \epsilon(\ell)$$

where the probability is taken over the random choices of $a, b, c, z \xleftarrow{R} \mathbb{Z}_p$ *and the bits consumed by the adversary* $\mathcal{A}$.

2.5.6 q-Strong Diffie-Hellman Assumption

Boneh and Boyen [BB04b] proposed this assumption and used it to develop a short signature scheme. This assumption is defined as follows.

Definition 2.13 q-Strong Diffie-Hellman (q-SDH) Assumption [BB04b]. *Let* $x \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{GG}(1^\ell) \rightarrow (p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, $\mathbb{G}_1 = \langle g \rangle$ *and* $\mathbb{G}_2 = \langle h \rangle$. *Given a* $(q+2)$*-tuple* $(g, h, h^x, \cdots, h^{x^q})$, *we say that the* q*-strong Diffie-Hellman assumption holds on* $(p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *if no PPT adversary* $\mathcal{A}$ *can compute* $(c, g^{\frac{1}{x+c}})$ *with the advantage*

$$Adv_{\mathcal{A}}^{q-SDH} = \Pr\left[\mathcal{A}(g, h, h^x, \cdots, g^{x^q}) \rightarrow (c, g^{\frac{1}{x+c}})\right] \geq \epsilon(\ell)$$

where $c \in \mathbb{Z}_p^*$ *and the probability is taken over the random choice of* $x \xleftarrow{R} \mathbb{Z}_p^*$ *and the bits consumed by the adversary* $\mathcal{A}$.

2.5.7 Chosen-Target Computational Diffie-Hellman Assumption

Boldyreva [Bol03] introduced this assumption and used it to design a blind signature scheme. This assumption is defined as follows.

Definition 2.14 Chosen-Target Computational Diffie-Hellman (CT-CDH) Assumption [Bol03]. *Let* $x \xleftarrow{R} \mathbb{Z}_p$, $\mathcal{G}(1^\ell) \rightarrow (p, \mathbb{G})$ *and* $\mathbb{G} = \langle g \rangle$. *Let* $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{G}$ *be a cryptographic hash function. There are two oracles: target oracle* $T_{\mathbb{G}}(\cdot)$ *and help oracle* $H_{\mathbb{G}}(\cdot)$. $T_{\mathbb{G}}(\cdot)$ *takes as input* $i \in \mathbb{Z}_p$ *and outputs* $g_i \in \mathbb{G}$. $H_{\mathbb{G}}(\cdot)$ *takes as input* $g_i \in \mathbb{G}$ *and outputs* $g_i^x \in \mathbb{G}$. *Let* q_T *and* q_H *denote the number of times that the two oracles are queried, respectively. We say that the chosen-target computational Diffie-Hellman assumption holds on* $(p, \mathbb{G})$ *if no PPT adversary* $\mathcal{A}$ *can have the advantage*

$$Adv_{\mathcal{A}}^{CT-CDH} = \Pr\left[\mathcal{A}^{T_{\mathbb{G}}(\cdot), H_{\mathbb{G}}(\cdot)}(p, \mathcal{H}, g, g^x) \rightarrow \{(i_1, \theta_1), \cdots, (i_{q+1}, \theta_{q+1})\}\right] \geq \epsilon(\ell)$$

where $g_{i_j} \in \{g_1, g_2, \cdots, g_{q+1}\}$, $\theta_j = g_{i_j}^x$ *for* $j = 1, 2, \cdots, q+1$, *and* $q_H \leq q < q_T$.

The CT-CDH assumption is the analogous version of the chosen-target RSA inversion (RSA-CTI) assumption [BNPS02].

2.5.8 EXtended Chosen-Target Computational Diffie-Hellman Assumption

Intuitively, in the above CT-CDH assumption, after the adversary $\mathcal{A}$ queries the help oracle $H_{\mathbb{G}}(\cdot)$ on the elements in $\mathbb{G}$ at most q_H times, he cannot compute a new element in $\mathbb{G}$ to the power of x if its discrete logarithm on the generator and the q_H queried elements are unknown. Based on the CT-CDH assumption, we propose the extended CT-CDH (XCT-CDH) assumption, by replacing the target oracle in CT-CDH assumption with $(q_H + 1)$ random elements of $\mathbb{G}$.

Definition 2.15 EXtended Chosen-Target Computational Diffie-Hellman (XCT-CDH) Assumption. *Let $x \overset{R}{\leftarrow} \mathbb{Z}_p$, $\mathcal{G}(1^\ell) \rightarrow (p, \mathbb{G})$ and $\mathbb{G} = \langle g \rangle$. Suppose that $H_{\mathbb{G}}(\cdot)$ be a help oracle which takes as input $g_i \in \mathbb{G}$ and outputs $g_i^x \in \mathbb{G}$. Let q_H be the number of the times which the oracle is queried. Given a $(q+1)$-tuple $(g^{a_1}, g^{a_2}, \cdots, g^{a_{q+1}})$, we say that the extended chosen-target computational Diffie-Hellman assumption holds on $(p, \mathbb{G})$ if no PPT adversary $\mathcal{A}$ can have the advantage*

$$Adv_{\mathcal{A}}^{XCT-CDH} = \Pr\left[\mathcal{A}^{H_{\mathbb{G}}(\cdot)}(p, g, g^x, g^{a_1}, \cdots, g^{a_{q+1}}) \rightarrow (g^{xa_1}, \cdots, g^{xa_{q+1}})\right] \geq \epsilon(\ell)$$

where $a_i \overset{R}{\leftarrow} \mathbb{Z}_p$ for $i = 1, 2, \cdots, q+1$ and $q_H \leq q$.

We have the following result about CT-CDH assumption and XCT-CDH assumption.

Theorem 2.1 *The extended chosen-target computational Diffie-Hellman assumption and the chosen-target Diffie-Hellman assumption are equivalent.*

Proof: Given the $(q+1)$-tuple $\{g^{a_1}, g^{a_2}, \cdots, g^{a_{q+1}}\}$ where $a_i \overset{R}{\leftarrow} \mathbb{Z}_p$ for $i = 1, 2, \cdots, q+1$, we define a function $\mathcal{H} : j \rightarrow g^{a_{i_j}} \in \mathbb{G}$, where $a_{i_j} \in \{a_1, a_2, \cdots, a_{q+1}\}$ for $j = 1, 2, \cdots, q+1$; otherwise $\mathcal{H} : j \rightarrow g^{b_j}$, where $b_j \overset{R}{\leftarrow} \mathbb{Z}_p$. So, $\mathcal{H}(\cdot)$ is a cryptographic hash function 2.6.2, where the domain is $\mathbb{Z}^+$ and the range is $\mathbb{G}$.

On the one hand, if an adversary $\mathcal{A}$ can break the CT-CDH assumption, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the XCT-CDH assumption as follows. Given $\{g^{a_1}, g^{a_2}, \cdots, g^{a_{q+1}}\}$, for q_T $(q_T \leq q+1)$ target oracle queries, $\mathcal{B}$ responds with $g^{a_{i_1}}, g^{a_{i_2}}, \cdots, g^{a_{i_{q_T}}}$, where $a_{i_j} \in \{a_1, a_2, \cdots, a_{q+1}\}$ for $j = 1, 2, \cdots, q_T$. For q_H $(q_H \leq q)$ help oracle queries, $\mathcal{B}$ queries the help oracle $H_{\mathbb{G}}(\cdot)$ in the XCT-CDH assumption, and responds $\mathcal{A}$ with $\{g^{xa_{i_1}}, g^{xa_{i_2}}, \cdots, g^{xa_{i_{q_H}}}\}$, where $a_{i_j} \in \{a_1, a_2, \cdots, a_{q+1}\}$ for $j = 1, 2, \cdots, q_H$. If $\mathcal{A}$ can output

$\{(i_1, \theta_1), (i_2, \theta_2), \cdots, (i_{q+1}, \theta_{q+1})\}$ where $\theta_j = g_j^x$ for $j = 1, 2, \cdots, q+1$, $\mathcal{B}$ can output $\{g^{xa_1}, g^{xa_2}, \cdots, g^{xa_{q+1}}\}$, where $\mathcal{H}(j) = g_j = g^{a_{i_j}}$ for $j = 1, 2, \cdots, q+1$. So, $\mathcal{B}$ can break the XCT-CDH assumption.

On the other hand, if $\mathcal{A}$ can break the XCT-CDH assumption, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the CT-CDH assumption as follows. When $\mathcal{A}$ queries the help oracle on $\{g^{a_{i_1}}, g^{a_{i_2}}, \cdots, g^{a_{i_{q_H}}}\}$, $\mathcal{B}$ queries the help oracle $H_{\mathbb{G}}(\cdot)$ in CT-CDH assumption, and responds with $\{(i_1, \theta_1), (i_2, \theta_2), \cdots, (i_{q_H}, \theta_{q_H})\}$ where $\theta_j = g_j^x = g^{xa_{i_j}}$ for $j = 1, 2, \cdots, q_H$. If $\mathcal{A}$ can output $\{g^{xa_1}, g^{xa_2}, \cdots, g^{xa_{q+1}}\}$, $\mathcal{B}$ can output $\{(i_1, \theta_1), (i_2, \theta_2), \cdots, (i_{q+1}, \theta_{q+1})\}$ where $\mathcal{H}(j) = g_j = g^{a_{i_j}}$ and $\theta_j = g^{xa_{i_j}} = g_j^x$ for $j = 1, 2, \cdots, q+1$. So, $\mathcal{B}$ can break the CT-CDH assumption.

Therefore, the XCT-CDH assumption is equivalent to CT-CDH assumption.

□

Notably, the XCT-CDH assumption is the CDH assumption if the help oracle $H_{\mathbb{G}}(\cdot)$ in the XCT-CDH assumption is canceled.

2.6 Cryptographical Tools

In this section, we introduce some useful cryptographical tools, including Lagrange interpolation, hash function, random oracle model, commitment, public-key encryption, broadcast encryption, Waters's identity-based encryption, digital signature and zero-knowledge proof.

2.6.1 Lagrange Interpolation

Let $p(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ be a $(k-1)$-degree polynomial. Given any k different values $p(x_1),\ p(x_2),\ \cdots,\ p(x_k)$, the polynomial $p(x)$ can be reconstructed as follows:

$$p(x) = \sum_{x_i \in \mathbf{S}} p(x_i) \prod_{x_j \in \mathbf{S}, x_j \neq x_i} \frac{x - x_j}{x_i - x_j} = \sum_{x_i \in \mathbf{S}} p(x_i) \Delta_{S, x_i}(x)$$

where $\mathbf{S} = \{x_1, x_2, \cdots, x_k\}$. The Lagrange coefficient for x_i in $\mathbf{S}$ is

$$\Delta_{\mathbf{S}, x_i}(x) = \prod_{x_j \in \mathbf{S}, x_j \neq x_i} \frac{x - x_j}{x_i - x_j}.$$

Therefore, given any k different polynomial values, we can compute $p(x)$ for all $x \in \mathbb{Z}_p$. Nevertheless, the other polynomial values are unconditionally secure if only $k-1$ different polynomial values are given.

2.6.2 Hash Function

Carter and Wegman [CW79] introduced the universal classes of hash functions and divided them into tree types. Roughly speaking, a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ is a deterministic function which can map a bit string with any length to a bit string with fixed length λ. A hash function should provide the following properties [Mao03]:

1. Mixing Transformation. The output of $\mathcal{H}$ should be computationally indistinguishable from a uniform binary string in $[0, 2^\lambda]$;

2. Pre-image Resistance. Given a value y, it is computationally infeasible to find a value x such that $y = \mathcal{H}(x)$;

3. Collusion Resistance. It is computationally infeasible to find $x \neq y$ such that $\mathcal{H}(x) = \mathcal{H}(y)$.

Hash function is an important cryptographical primitive and has been used as a building block to design encryption scheme [FOPS01], digital signature scheme [BR93], message authentication code (MAC) scheme [BCK96], *etc.*

2.6.3 Random Oracle Model

A hash function should satisfy the mixing transformation property, namely the output of a hash function is computationally indistinguishable from the uniform distribution over its output's space. If the output of a hash function is uniform distribution over its output's space, it is a very powerful and ideal hash function called *random oracle* [Mao03]. A random oracle is a powerful hash function as it combines the properties: deterministic, efficient and uniform output. Furthermore, a random oracle is an ideal hash function as there are no so powerful computing mechanism or machinery in current computing models.

Bellare and Rogaway [BR93] introduced the notion of *random oracle model*. In this model, a special entity called *Simulator* can simulate every party's behavior. So, whenever a party wants to obtain the output of a random oracle $\mathcal{H}$ on a value x, he must make a *random oracle query* on the value x to the *Simulator*. *Simulator* maintains a $\mathcal{H}$-table consisting of pairs $(z, \mathcal{H}(z))$. For a query on the value x, *Simulator* checks whether x is listed in the table. If it has been in the table, *Simulator* responds with the value $\mathcal{H}(x)$ (*deterministic*); otherwise, *Simulator* creates a

new value $\mathcal{H}(x)$ uniformly at random from the output's space of $\mathcal{H}$, adds the pair $(x, \mathcal{H}(x))$ to the table and responds with $\mathcal{H}(x)$ (*uniform*).

Random oracle model is a very efficient tool to prove the security of cryptographic protocols. Generally, protocols designed in this model are more efficient than those designed in standard model. Whereas, a scheme which is proven to be secure in the random oracle model does not necessarily imply that it is secure in the standard model [CGH98].

Unless otherwise specified, by saying a scheme is secure, we mean that it is secure in the standard model in this thesis.

2.6.4 Commitment Scheme

A commitment scheme is a fundamental component of modern cryptographical protocols. In a commitment scheme, a player can commit his choice to a value called *commitment*. The value *commitment* does not reveal any information about his choice until the player disclose it (*hiding*). Meanwhile, after publishing the commitment, the player cannot change his choice (*binding*). Damgård [Dam99] surveyed various commitment schemes. A commitment scheme is formally defined as follows [BCC88]:

$\mathsf{Setup}(1^\ell) \rightarrow params$. The setup algorithm takes as input 1^ℓ and outputs the public parameters $params$.

$\mathsf{Commit}(params, M) \rightarrow (com, decom)$. The commitment algorithm takes as input the public parameters $params$ and a message M, and outputs a pair $(com, decom)$. The value com is public, while the value $decom$ is kept secret.

$\mathsf{Decommit}(params, M, com, decom) \rightarrow \{0, 1\}$. The decommitment algorithm takes as input the public parameters $params$, the message M and the pair (com, $decom$), and outputs 1 if $decom$ can decommit com to M; otherwise, it outputs 0.

A commitment scheme should provide two properties: *hiding* and *binding*.

Hiding. This property is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ algorithm and sends the public parameters $params$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ outputs two message M_0 and M_1 with equal length. $\mathcal{C}$ computes $\mathsf{Commit}(params, M_b)$ and sends com to $\mathcal{A}$, where $b \in \{0, 1\}$.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 2.16 *We say that a commitment scheme provides hiding property if no PPT adversary $\mathcal{A}$ can win the game with the the advantage*

$$Adv_{\mathcal{A}}^{H} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

Binding. This property is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ algorithm and sends the public parameters $params$ to $\mathcal{A}$.

Output. $\mathcal{A}$ outputs two messages (M_0, M_1). $\mathcal{A}$ wins the game if $M_0 \neq M_1$ and $\mathsf{Commit}(params, M_0) = \mathsf{Commit}(params, M_1)$.

Definition 2.17 *We say that a commitment scheme provides binding property if no PPT adversary $\mathcal{A}$ can win the game with the advantage*

$$Adv_{\mathcal{A}}^{B} = \Pr\left[\mathsf{Commit}(params, M_0) = \mathsf{Commit}(params, M_1)\right] \geq \epsilon(\ell)$$

in the above model.

In this thesis, we use the commitment scheme proposed by Pedersen [Ped92]. This scheme works as follows:

Setup. This algorithm takes as input 1^ℓ and outputs $\mathcal{G}(1^\ell) \rightarrow (p, \mathbb{G})$. Let $g_0, g_1, \cdots, g_\kappa$ be generators of $\mathbb{G}$.

Commit. To commit a set of messages $(m_1, m_2, \cdots, m_\kappa)$, this algorithm takes as input a value $r \xleftarrow{R} \mathbb{Z}_p$ and $(m_1, m_2, \cdots, m_\kappa)$, and outputs $com = g_0^r g_1^{m_1} \cdots g_\kappa^{m_\kappa}$. com is public, while r keeps secret.

Decommit. This algorithm takes as input $(r, m_0, m_1, \cdots, m_\kappa)$ and com, and checks $com \stackrel{?}{=} g_0^r g_1^{m_1} \cdots g_\kappa^{m_\kappa}$. If $com = g_0^r g_1^{m_1} \cdots g_\kappa^{m_\kappa}$, it outputs 1; otherwise, it outputs 0.

The commitment scheme [Ped92] is perfectly hiding and computationally binding if the discrete logarithm assumption holds on the group $\mathbb{G}$.

2.6.5 Public-Key Encryption

Diffie and Hellman [DH76] introduced new research directions in cryptography called *public-key cryptography* (PKC) where two parties can communicate over public channels without compromising the security of the system.

A public-key (asymmetric) encryption (PKE) scheme is a public-key cryptographic scheme used to protect the confidentiality of the transferred massages. In a PKE scheme, a secret-public key pair is generated. Notably, it is computationally infeasible to obtain the secret key from the public key. This is in contrast with a symmetric encryption scheme where both the decryption key and the encryption key are same or it is easy to compute one from the other.

The formal definition of a PKE scheme is as follows [DH76]. A PKE scheme consists of the following four algorithm.

$\mathsf{Setup}(1^\ell) \to params$. The setup algorithm takes as input 1^ℓ and outputs the public parameters $params$.

$\mathsf{KeyGen}(1^\ell) \to (SK, PK)$. The key generation algorithm takes as input 1^ℓ and outputs a secret-public pair $\mathcal{KG}(1^\ell) \to (SK, PK)$.

$\mathsf{Enc}(params, PK, M) \to CT$. The encryption algorithm takes as input the public parameters $params$, the public key PK and a message M, and outputs a ciphertext CT.

$\mathsf{Dec}(params, SK, CT) \to M$. The decryption algorithm takes as input the public parameters $params$, the secret key SK and the ciphertext CT, and outputs the message M.

Definition 2.18 Correctness. *We say that a public-key encryption scheme is correct if*

$$\Pr\left[\mathsf{Dec}(params, SK, CT) \to M \;\middle|\; \begin{array}{l} \mathsf{Setup}(1^\ell) \to params; \\ \mathsf{KeyGen}(1^\ell) \to (SK, PK); \\ \mathsf{Enc}(params, PK, M) \to CT \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by all algorithms in the scheme.

Security Model. The standard notion of the security for a PKE scheme is called *indistinguishability against adaptive chosen ciphertext attacks* (IND-CCA2) [RS92].

This model is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters *params* and sends them to $\mathcal{A}$.

KeyGen. $\mathcal{C}$ runs $\mathsf{KeyGen}(1^\ell)$ to generate the secret-public key pair (SK, PK) and sends the public key PK to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can adaptively query the decryption oracle. $\mathcal{A}$ submits a ciphertext CT to $\mathcal{C}$, where $CT = Enc(param, PK, M)$. $\mathcal{C}$ runs $\mathsf{Dec}(params, SK, CT)$ and responds $\mathcal{A}$ with M. This query can be made multiple times.

Challenger. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{C}$ randomly selects M_b and computes $CT^* = \mathsf{Enc}(params, PK, M_b)$, where $b \in \{0, 1\}$. $\mathcal{C}$ responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can adaptively query the decryption oracle. $\mathcal{A}$ submits a ciphertext CT to $\mathcal{C}$, where the only restrict is $CT \neq CT^*$. Phase 1 is repeated. This query can be made multiple times.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 2.19 IND-CCA2. *We say that a public-key encryption scheme is* $(T, q, \epsilon(\ell))$*-indistinguishable against adaptive chosen ciphertext attacks (IND-CCA2) if no PPT adversary* $\mathcal{A}$ *making* q *decryption queries can win the game with the advantage*

$$Adv_{\mathcal{A}}^{IND-CCA2} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

Another security notion for public-key encryption is called *indistinguishability against adaptive chosen plaintex attacks* (IND-CPA). In this model, the adversary $\mathcal{A}$ is not allowed to query the decryption oracle. The formal definition for this model is as follows.

Definition 2.20 IND-CPA. *We say that a public-key encryption scheme is* $(T, \epsilon(\ell))$*-indistinguishable against adaptive chosen plaintex attacks (IND-CPA) if no PPT*

adversary $\mathcal{A}$ who is restricted to query the decryption oracle can win the game with the advantage

$$Adv_{\mathcal{A}}^{IND-CPA} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

Some well known PKE schemes are ElGamal encryption scheme [ElG85], RSA encryption scheme [RSA78], CS encryption scheme [CS98] and RSA-OAEP encryption scheme [FOPS01]. [ElG85] and [RSA78] schemes are IND-CPA secure. [CS98] scheme is IND-CCA2 secure, while [FOPS01] is IND-CCA2 secure in the random oracle model.

2.6.6 Broadcast Encryption

Proposed by Fiat and Naor [FN94], broadcast encryption is a PKE scheme where a broadcaster encrypts a message for a subset of users. As a result, only the user in the subset can use his secret key to decrypt the ciphertext and obtain the message. The user outside the subset cannot obtain anything about the message.

A broadcast encryption scheme consists of the following four algorithms.

$\mathsf{BSetup}(1^\ell) \rightarrow params$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$.

$\mathsf{BKeyGen}(params, n) \rightarrow \{SK_i\}_{i=1}^n$. The key generation algorithm takes as input the public parameters $params$ and the number of users n, and outputs n secret keys SK_i for $i = 1, 2, \cdots, n$.

$\mathsf{BEnc}(params, \mathbf{S}) \rightarrow (Hdr, K)$. The encryption algorithm takes as input the public parameters $params$ and a subset of users $\mathbf{S} \subseteq \{1, 2, \cdots, n\}$, and outputs (Hdr, K) where Hdr is called the header and $K \in \mathcal{K}$ is a message encryption key. When broadcasting a message $M \in \{0,1\}^*$, the broadcaster runs $\mathsf{BEnc}(params, \mathbf{S}) \rightarrow (Hdr, K)$, computes $CT = \mathcal{E}(params, K, M)$ and broadcasts $(\mathbf{S}, Hdr, CT)$, where $\mathcal{E}(\cdot)$ is a symmetric encryption algorithm.

$\mathsf{BDec}(params, SK_j, Hdr) \rightarrow K$. The decryption algorithm take as input the public parameters $params$, the secret key SK_j with $j \in \mathbf{S}$ and the header Hdr, and outputs the message encryption key K.

Definition 2.21 Correctness. *We say that a broadcast encryption scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{BDec}(params, SK_j, Hdr) \rightarrow K & \begin{array}{l}\mathsf{BSetup}(1^\ell) \rightarrow params; \\ \mathsf{BKeyGen}(params, n) \rightarrow \{SK_i\}_{i=1}^n; \\ \mathsf{BEnc}(params, \mathbf{S}) \rightarrow (Hdr, K); \\ j \in \mathbf{S}\end{array}\end{array}\right] = 1$$

Security Model. The security model of broadcast encryption schemes is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ [BGW05].

Initialization. $\mathcal{A}$ submits a subset $\mathbf{S}^* \subseteq \{1, 2, \cdots, n\}$ with which he wants to be challenged.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and responds $\mathcal{A}$ with $params$.

Phase 1. $\mathcal{A}$ can adaptively query the key generation oracle. $\mathcal{A}$ adaptively submits subsets of users $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{q_1}$ where the only constraints are $\mathbf{S}_j \subset \mathbf{S}^*$ for $j = 1, 2, \cdots, q_1$. $\mathcal{C}$ runs $\mathsf{BKeyGen}(params, n)$ to generates $\{SK_i\}_{i=1}^n$, and responds $\mathcal{A}$ with $\{SK_t\}_{t \in \mathbf{S}_j}$ for $j = 1, 2, \cdots, q_1$.

Challenge. $\mathcal{C}$ runs $\mathsf{Enc}(params, \mathbf{S}^*) \rightarrow (Hdr^*, K^*)$. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ sets $K_b = K^*$ and $K_{1-b} \overset{R}{\leftarrow} \mathcal{K}$. $\mathcal{C}$ responds $\mathcal{A}$ with (Hdr^*, K_0, K_1).

Phase 2. $\mathcal{A}$ can adaptively query the key generation oracle. $\mathcal{A}$ adaptively submits subsets of users $\mathbf{S}_{q_1+1}, \mathbf{S}_{q_1+2}, \cdots, \mathbf{S}_q$ where the only constrains are $\mathbf{S}_j \subset \mathbf{S}^*$ for $j = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b = b'$.

Definition 2.22 B-IND-CCA2. *We say that a broadcast encryption scheme is* $(T, n, q, \epsilon(\ell))$*-secure against chosen ciphertext attacks (or IND-CCA2) if no PPT adversary* $\mathcal{A}$ *making at most* q *key generation queries can win the game with the advantage*

$$Adv_{\mathcal{A}}^{B-IND-CCA2} = \left|\Pr[b' = b] - \frac{1}{2}\right| \geq \epsilon(\ell)$$

in the above model.

2.6.7 Waters's Identity-based Encryption

In this section, we introduce Water's identity-based encryption (IBE) scheme [Wat05]. This IBE scheme is described as follows:

Setup. The setup algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\tau$ and the order p is a prime number. Let g and η be generators of the group $\mathbb{G}$, $u_0 \xleftarrow{R} \mathbb{G}$ and $\mathbf{U} = (u_1, u_2, \cdots, u_n)$ where $u_i \xleftarrow{R} \mathbb{G}$ for $i = 1, 2, \cdots, n$. It selects $\alpha \xleftarrow{R} \mathbb{Z}_p$ and sets $g_1 = g^\alpha$. The public parameters are $params = (e, p, \mathbb{G}, \mathbb{G}_\tau, g, \eta, u_0, \mathbf{U}, g_1)$ and the master secret key is η^α.

KeyGen. Let ID represent an identity which is an n bit binary string, ID_i be the ith bit of ID, and $\mathbf{I}$ be the set which consists of all i with $ID_i = 1$. The key generation algorithm chooses $r \xleftarrow{R} \mathbb{Z}_p$, and computes

$$K_{ID,1} = \eta^\alpha (u_0 \prod_{i \in \mathbf{I}} u_i)^r \quad \text{and} \quad K_{ID,2} = g^r.$$

The secret key for the identity ID is $SK_{ID} = (K_{ID,1}, K_{ID,2})$.

Encryption. To encrypt a message $M \in \mathbb{G}_\tau$, the encryption algorithm chooses $s \xleftarrow{R} \mathbb{Z}_p$ and computes

$$C_1 = M \cdot e(g_1, \eta)^s, \quad C_2 = g^s \quad \text{and} \quad C_3 = (u_0 \prod_{i \in \mathbf{I}} u_i)^s.$$

The ciphertext for the message M is $CT = (C_1, C_2, C_3)$.

Decryption. To decrypt the ciphertext $CT = (C_1, C_2, C_3)$, the decryption algorithm takes as input the secret key $K_{ID} = (K_{ID,1}, K_{ID,2})$ and computes

$$M = C_1 \cdot \frac{e(K_{ID,2}, C_3)}{e(K_{ID,1}, C_2)}$$

Security Model. The security model in [Wat05] was defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs the Setup(1^ℓ) to generate the public parameters $params$ and sends $params$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can adaptively query the key generation oracle. $\mathcal{A}$ can adaptively query secrete keys for identities $ID_1, ID_2, \cdots, ID_{q_1}$. $\mathcal{C}$ runs KeyGen to generates secret keys $SK_{ID_1}, SK_{ID_2}, \cdots, SK_{ID_{q_1}}$, and responds $\mathcal{A}$ with $SK_{ID_1}, SK_{ID_2}, \cdots, SK_{ID_{q_1}}$.

Challenge. $\mathcal{A}$ submitted an identity ID^* and two message M_1 and M_2 with same length, where the only restrict is that $ID^* \notin \{ID_1, ID_2, \cdots, ID_{q_1}\}$. $\mathcal{C}$ flips an unbiased coin with $\{0,1\}$ and obtain a bit $b \in \{0,1\}$. $\mathcal{C}$ runs Encryption to generate a ciphertext CT^* for the message M_b, and sends CT^* to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ can adaptively query secret keys for identities $ID_{q_1+1}, ID_{q_1+2}, \cdots, ID_q$, where the only restrict is $ID^* \notin \{ID_{q_1+1}, ID_{q_1+2}, \cdots, ID_q\}$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 2.23 IBE-IND-CPA. *We say that an identity-based encryption scheme is $(t, q, \epsilon(\ell))$-secure if no PPT adversary $\mathcal{A}$ making q key generation queries can win the game with the advantage*

$$Adv_{\mathcal{A}}^{IBE-IND-CPA} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

The security of this scheme can be reduced to the DBDH assumption.

Theorem 2.2 *This identity-based encryption scheme is $(T, q, \epsilon(\ell))$-secure agasint chosen plaintex attacks (or IND-CPA) if the $(T + \Theta(\epsilon(\ell)^{-2} ln(\epsilon(\ell)^{-1}) \lambda^{-1} ln(\lambda^{-1}))$, $\frac{\epsilon(\ell)}{32q(n+1)})$-decisional bilinear Diffie-Hellman (DBDH) assumption holds on the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $\lambda = \frac{1}{8q(n+1)}$ [Wat05].*

2.6.8 Digital Signature

Digital signature was proposed by Diffie and Hellman [DH76]. It is the electronic version of a handwritten signature. A valid digital signature can convince a verifier that it was generated by a known party for a public message. Especially, a digital signature can provide non-repudiation property, namely a signer cannot deny he has generated the signature.

A digital signature scheme is formally defined as follows [GMR88]. It consists of the following four algorithms.

$\mathsf{Setup}(1^\ell) \to params$. The setup algorithm takes as input 1^ℓ and outputs the public parameters $params$.

$\mathsf{KeyGen}(1^\ell) \to (SK, PK)$. The key generation algorithm takes as input 1^ℓ and outputs a secret-public key pair (SK, PK).

$\mathsf{Sign}(params, SK, M) \to \sigma$. The signature algorithm takes as input the public parameters $params$, the secret ky SK and a message M, and outputs a signature σ on M.

$\mathsf{Verify}(params, M, PK, \sigma) \to True/False$. The verification algorithm takes as input the public parameters $params$, the message M, the public key PK and the signature σ, and outputs $True$ if $\mathsf{Sign}(params, M, SK) \to \sigma$; otherwise, it outputs $False$.

Definition 2.24 Correctness. *We say that a digital signature is correct if*

$$\Pr\left[\mathsf{Verify}(params, M, PK, \sigma) \to True \;\middle|\; \begin{array}{l}\mathsf{Setup}(1^\ell) \to params;\\ \mathsf{KeyGen}(1^\ell) \to (SK, PK);\\ \mathsf{Sign}(params, SK, M) \to \sigma.\end{array}\right] \geq 1 - \epsilon(\ell)$$

and

$$\Pr\left[\mathsf{Verify}(params, M, PK, \sigma) \to False \;\middle|\; \begin{array}{l}\mathsf{Setup}(1^\ell) \to params;\\ \mathsf{KeyGen}(1^\ell) \to (SK, PK);\\ \mathsf{Sign}(params, SK, M) \to \sigma.\end{array}\right] < \epsilon(\ell)$$

where the probability is taken over the random coins consumed by all algorithms in the scheme.

Security Model. A digital signature scheme should achieve the traditional security called *existential unforgeability under adaptive chosen message attacks* (EU-CMA) [GMR88]. This model is formally defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and sends them to $\mathcal{A}$.

KeyGen. $\mathcal{C}$ runs $\mathsf{KeyGen}(1^\ell)$ to generate a secret-public pair (SK, PK) and sends PK to $\mathcal{A}$.

Query. $\mathcal{A}$ can adaptively query the signature oracle. $\mathcal{A}$ sends a message M to $\mathcal{C}$. $\mathcal{C}$ runs $\mathsf{Sign}(params, SK, M)$ to generate a signature σ on M and responds $\mathcal{A}$ with σ. This query can be made multiple times.

Output. $\mathcal{A}$ outputs a message-signature pair (M^*, σ^*). $\mathcal{A}$ wins the game if M^* has not been used to query the signature oracle and $\mathsf{Verify}(params, M^*, PK, \sigma^*) \rightarrow True$.

Definition 2.25 EU-CMA. *We say that a digital signature scheme is $(T, q, \epsilon(\ell))$-existentially unforgeable against adaptive chosen message attacks (EU-CMA) if no PPT adversary $\mathcal{A}$ can win the game with the advantage*

$$Adv_{\mathcal{A}}^{EU-CMA} = \Pr\left[\mathsf{Verify}(params, M^*, PK, \sigma^*) \rightarrow True\right] \geq \epsilon(\ell)$$

in the above model.

An, Dodis and Rabin [ADR02] proposed a stronger definition for the security of digital signature schemes called *strongly existential unforgeability under an adaptive chosen message attack* (SEU-CMA). This model is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and sends them to $\mathcal{A}$.

KeyGen. $\mathcal{C}$ runs $\mathsf{KeyGen}(1^\ell)$ to generate a secret-public pair (SK, PK) and sends PK to $\mathcal{A}$.

Query. $\mathcal{A}$ can adaptively query the signature oracle. $\mathcal{A}$ adaptively sends messages $\{M_1, M_2, \cdots, M_q\}$ to $\mathcal{C}$. $\mathcal{C}$ runs $\mathsf{Sign}(params, SK, M_i)$ to generate a signature σ_i on M_i and responds $\mathcal{A}$ with σ_i, for $i = 1, 2, \cdots, q$.

Output. $\mathcal{A}$ outputs a message-signature pair (M^*, σ^*). $\mathcal{A}$ wins the game if $(M^*, \sigma^*) \notin \{(M_1, \sigma_1), (M_2, \sigma_2), \cdots, (M_q, \sigma_q)\}$ and $\mathsf{Verify}(params, M^*, PK, \sigma^*) \rightarrow True$.

Definition 2.26 SEU-CMA. *We say that a digital signature scheme is $(T, q, \epsilon(\ell))$-strongly existentially unforeable against adaptive chosen message attacks (SEU-CMA) if no PPT adversary $\mathcal{A}$ can win the game with the advantage*

$$Adv_{\mathcal{A}}^{SEU-CMA} = \Pr\left[\mathsf{Verify}(params, M^*, PK, \sigma^*) \rightarrow True\right] \geq \epsilon(\ell)$$

in the above model.

2.6.9 Zero-Knowledge Proof

Introduced by Goldwasser, Micali and Rackoff [GMR86], a zero-knowledge proof is an interactive protocol which can be used by a prover to convince a verifier that a statement is true without releasing any more information than the validity of the statement. A zero-knowledge proof of knowledge (ZK-PoK) is a protocol which can be used by a prover to convince a verifier that he knows a secret value without the verifier knowing anything about the value. There are two parties in a zero-knowledge proof: a prover $\mathcal{P}$ which has unlimited computation ability and a verifier $\mathcal{V}$ which is computationally bound. By $(\mathcal{P} \leftrightarrow \mathcal{V})[x]$, we denote that $\mathcal{P}$ proves to $\mathcal{V}$ that the statement x is correct. The formal definition for a perfect zero-knowledge proof [GMW86] is as follows.

Definition 2.27 *A pair $(\mathcal{P} \leftrightarrow \mathcal{V})$ is an interactive proof system for a language $\mathcal{L}$ if the following properties can be satisfied:*

1. Completeness. *For all $x \in \mathcal{L}$, $\Pr[\mathcal{V}(x,s) = 1 | (\mathcal{P} \leftrightarrow \mathcal{V})[x] \rightarrow s] = 1 - \frac{1}{n^\kappa}$ for each κ and the sufficient large input length n.*

2. Soundness. *For all $x \notin \mathcal{L}$ and $\mathcal{P}'$, $\Pr[\mathcal{V}(x,s) = 1 | (\mathcal{P}' \leftrightarrow \mathcal{V})[x] \rightarrow s] \leq \frac{1}{n^\kappa}$. In other words, if $\mathcal{P}'$ can convince $\mathcal{V}$ that $x \notin \mathcal{L}$ is correct with the advantage ϵ, there exists a* knowledge extractor, *given rewindable black-box access [1] to $\mathcal{P}'$, can output the witness of the statement x with the advantage $\epsilon - \frac{1}{n^\kappa}$.*

3. Zero-Knowledge. *For all $x \in \mathcal{L}$ and $\mathcal{V}$, there exists an simulator $\mathcal{S}$ such that the two outputs $\mathcal{S}_\mathcal{V}(x)$ and $\mathcal{V}(x)$ are indistinguishable, where $\mathcal{S}_\mathcal{V}(x)$ denotes the distribution generated by the simulator $\mathcal{S}$ on input x and $\mathcal{V}(x)$ denotes the distribution generated by the verifier $\mathcal{V}$ who interacts with the prover $\mathcal{P}$ on inputs x.*

All languages in $\mathcal{NP}$ have zero-knowledge proofs if there exist one-way functions [GMW86].

Σ-Protocol. A Σ-protocol is an interactive proof system $(\mathcal{P} \leftrightarrow \mathcal{V})$ with the output of three-tuple (x, c, y) where x and y are computed by $\mathcal{P}$, while c is randomly selected by $\mathcal{V}$. We call the secret value held by $\mathcal{P}$ as *witness*. Given an efficient

[1] *In a rewindable black-box access, the extractor can send any values which its selects to the prover and obtain the corresponding outputs of the prover, without knowing how the prover computes the outputs. It allows the extractor to literally rewind a run of the prover to a previous state.*

computable predicate φ, $\mathcal{V}$ accepts the proof if $\varphi(x, c, y) = 1$. Notably, given two accepted conversations (x, c, y) and (x, c, y') with $(x, c, y) \neq (x, c, y')$, the *witness* can be efficiently extracted. Furthermore, there exists a simulator $\mathcal{S}$ can generate a transcript which is indistinguishable from that generated by $\mathcal{P}$.

Camenisch and Stadler [CS97] proposed a notion which is suitable for various zero-knowledge proof of knowledge of discrete logarithms and proofs of the validity about discrete logarithm statements. Let two cyclic groups be $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\mathbb{G}' = \langle \mathfrak{g} \rangle = \langle \mathfrak{h} \rangle$. The notion is as follows:

$$\text{PoK}\{(x, y, z) : u = g^x h^y \wedge \mu = \mathfrak{g}^x \mathfrak{h}^z \wedge (a \leq x \leq b)\}.$$

This notion can be used to demonstrate a zero-knowledge proof of knowledge of x, y and z such that $u = g^x h^y$ and $\mu = \mathfrak{g}^x \mathfrak{h}^z$ hold simultaneously with $a \leq x \leq b$. Generally, the letters in the parenthesis denote the knowledge which is being proven, while other parameters are known by the verifier.

Part I

Accessed Contents Protection

Chapter 3

Identity-based Distributed Data Storage

In this chapter, we propose two identity-based distributed data storage (IBDDS) schemes. The first scheme is secure against the chosen plaintext attacks (or IND-CPA), while the second scheme is secure against the chosen ciphertext attakcs (or IND-CCA2). Parts of this work appeared in [HSM13b].

3.1 Introduction

Data outsourcing provides users with a convenient service to manage their personal data with the notion called database-as-a-service (DaaS) [HIM02]. In a DaaS scheme, a user can outsource his encrypted data to untrusted proxy servers. Proxy servers can perform some functions on the outsourced ciphertexts without knowing anything about the original data. Unfortunately, this technique has not been extensively exploited . The main reason is that users are especially concerned on the *confidentiality*, *integrity* and *query* of the outsourced data as cloud computing is manipulated by an untrusted third party and a more complicated environment than the local data storage systems. After outsorcing data to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the outsoured data is not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been addressed in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced data from the proxy servers is another concern as the proxy server only maintains the outsourced ciphertexts. Consequently, research around these topics arises significantly.

Confidentiality. Confidentiality is proposed to prevent unauthorized users from accessing the sensitive data as it is subject to be unauthorized disclose and accessed after being outsourced. Since the seminal introduction of DaaS, the confidentiality

of outsourced data has been the primary focus among the data storage research community. To provide confidentiality to the sensitive data, encryption schemes are exploited [AFGH06].

Integrity. Integrity can be used to prevent outsourced data from being replaced and modified. Schemes towards to protect the integrity of the outsourced data have been proposed, such as provable data possession scheme [ABC+07] and proof of retrievability scheme [JJ07]. In these schemes, digital signature schemes and message authentication codes (MAC) are deployed.

Query. Query in a data storage scheme is executed between a requester and a proxy server. The proxy server can perform some functions on the outsourced ciphertexts and convert them to those for the requester. As a result, the requester can access the data outsourced by the owner without releasing anything about it to the proxy server [YGJK10, HN11].

3.1.1 Related Work

In this section, we review the literature related to IBDDS schemes.

Data Storage Systems

A data storage system enables users to store their data to external proxy servers to enhance the access and availability, and reduce the maintainance cost. Samarati and Vimercati [SV10] addressed the privacy issues in data outsouring schemes expanding from the data confidentiality to data utility, and pointed out some main research directions in the protection of the externally stored data. Kher and Kim [KK05] surveyed the data storage schemes comprehensively and classified them into three types based on their secure services: networked file systems, storage-based intrusion detection systems and cryptographic file systems.

Networked File Systems. In these systems, proxy servers are supposed to be trusted. They authenticate clients and validate their access permissions. The interactions between the proxy servers and requesters are executed in a secure channel. Therefore, these systems cannot provide an end-to-end security [SGK+85]. In these schemes, a requester authenticates himself to a proxy server by his password. Then, the proxy server redirects the authentication result to the file owner. The file owner determines whether or not to grant an access permission according to the authentication result.

Storage-based Intrusion Detection Systems. In these systems, an intrusion detection scheme is embedded in proxy servers or the file owner to detect the intruder's behaviors, such as adding backdoors, inserting Trojan horses and tampering with audit logs. These systems can be divided into two types: host-based system and network-based system. In the first kind of systems, an intrusion detection scheme is embedded in the host to detect the local intrusion behaviors [FHS96]. On the contrary, in second kind of systems, an intrusion detection scheme is embedded in proxy servers to detect the external intruder's behaviors. The main advantage of the latter systems is that proxy servers can still detect the intrusion actions even if the host is compromised as they are independent from the host [PSG+03].

Cryptographic File System. In these systems, an end-to-end security is provided by cryptographic protocols which are exploited by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive data. These systems can be classified into two types: shared file systems and non-shared systems. In a shared file system [KRS+03], the owner can share his data with a group of users. Cryptographic techniques exploited in these systems are key sharing, key agreement and key revocation. While, in a non-shared file system [Bla93], in order to share his data with another user, the owner can compute an access key for the user using his secret key. In these two systems, the integrity of the sensitive data is provided by digital signature schemes and MACs.

Identity-based Proxy Re-encryption

Introduced by Mambo and Okamoto [MO97], a proxy cryptosystem was used to delegate the decryption power to a designated decryptor. Then, Blaze, Bleumer, and Strauss [BBS98] proposed an atomic proxy cryptosystem where a semi-trusted proxy server can transfer a ciphertext for the original decryptor to a ciphertext for the designated decryptor without knowing anything about the plaintext. Proxy cryptosystem as an efficient primitive has been used in various scenarios, such as email forwarding, law enforcement and data storage.

Identity-based cryptosystem introduced by Shamir [Sha84] is a system where the public key can be any arbitrary string and the secret key is issued by a trusted third party called private key generator (PKG). Being different from the public-key infrastructure (PKI), two entities can communicate directly without verifying each public-key certificate in an identity-based system. The first secure and practical

identity-based encryption (IBE) scheme was proposed by Boneh and Franklin [BF01] based on pairing.

Ivan and Dodis [ID03] introduced the notion of identity-based proxy encryption (IBPE) and proposed formal definitions and security models for both unidirectional and bidirectional IBPE schemes. In their schemes, the master secret key which is used to extract secret keys for users is split into two parts. One is sent to the proxy server and the other is sent to the user. As a result, the user can collaborate with the proxy server to decrypt a ciphertext for him. Consequently, Ateniese, Fu, Green and Hohenberger [AFGH06] pointed out that these schemes are not secure against the *collusion attacks*, namely the master secret key can be disclosed if the user can compromise the proxy server. The first identity-based proxy re-encryption (IBPRE) was proposed by Green and Ateniese [GA07] where the proxy sever can transfer a ciphertext for the original decryptor to a ciphertext for the designated decryptor after he has obtained a re-encryption key from the former. We classify all IBPRE schemes into the following two types based on the generation of the re-encryption key.

The re-encryption key can be computed by the original decryptor [GA07, CT07, THJ08]. In these schemes, for a decryption request, the original decryptor chooses a random number and randomize his secret key to generate a re-encryption key. Then, he encrypts the chosen random number under the requester's identity. Finally, he sends the re-encryption key and the ciphertext to the proxy server. The proxy server can transfer a ciphertext for the original decryptor to a ciphertext for the designated decryptor by using the re-encryption key. The designated decryptor decrypts the ciphertext using his secret key and gets the random number chosen by the original decryptor. Then, he can decrypt the re-encrypted ciphertext by using the random number. Unfortunately, these schemes are subject to the collusion attacks. If the designated decryptor can compromise the proxy server, they can decrypt the ciphertext, obtain the random number chosen by the original decryptor and compute the secret key of the original decryptor.

The re-encryption key must be computed by the original decryptor with the help from the PKG [Mat07, WWMO10b, WWMO10a]. In these schemes, the original decryptor can cooperate with the PKG to compute a re-encryption key by checking the secret keys of the original decryptor and the designated decryptor.

Identity-based Distributed Data Storage

In an IBDDS scheme, a user's identity can be an arbitrary string and two parties can communicate with each other directly without the necessity to verify the public-key certificates. At first, the file owner encrypts all his files under his identity prior to outsourcing them to multiple proxy servers. Then, he sends the ciphertexts to the proxy servers. Consequently, the proxy servers can transfer a ciphertext encrypted under the identity of the owner to a ciphertext encrypted under the identity of the requester after he has gained an access permission from the owner.

As respect to the confidentiality of the outsouced data, a secure IBDDS scheme should provide the following properties:

1. *Unidirectional.* After receiving an access permission from Alice, the proxy server can transfer a ciphertext for Alice to a ciphertext for Bob, while he cannot transfer a ciphertext for Bob to a ciphertext for Alice. Because, Alice permits Bob to access his file does not mean that Bob permits Alice to access his files.

2. *Non-interactive.* The access permissions can be created by the file owner without any help of the PKG. So, it is very convenient for the owner to create the access permissions.

3. *Key optimal.* The length of the requester's secret key is constant and independent of the delegations which he accepts. This property is important as the security and management of secret keys is a difficult problem for users.

4. *Collusion-safe.* The owner's secret-key keeps nondisclosure even the requester can compromise the proxy server. This is necessary as it is possible that the requester can collude the proxy server in practice.

5. *Non-transitive.* Receiving two access permissions computed by Alice for Bob and Bob for Charlie, the proxy server cannot transfer a ciphertext for Alice to a ciphertext for Charlie. Because Alice permits Bob to access her files and Bob permits Charlie to access his files does not imply that Alice permit Charlie to access her files.

6. *File-based access.* For one request, the requester can only access one of the owner's files even he can compromise the proxy servers. This property is

important as it can improve the security of the outsourced files and is desirable to maintain the access record of the files.

Here, 1-5 are from [AFGH06]. *Proxy invisibility* [AFGH06] is difficult to achieve as the length of the re-encrypted ciphertext is subject to be different from that of the original ciphertext. Furthermore, *original-access* [AFGH06] cannot be guaranteed as the *key escrow* problem, namely the secret key is generated by the PKG, instead of the user himself. Hence, the file owner in an IBDDS scheme has less control on his secret key than that in other public-key encryption schemes.

Although existing IBPRE schemes can provide partial properties of IBDDS, they cannot be exploited in IBDDS systems directly. For example, in the current IBPRE schemes, a requester can collaborate with the proxy servers to access all the owner's files using a re-encryption key as this key is only bound to the requester's identity and independent of the ciphertext. This is undesirable for the file owner to record the accessed number of his files. Furthermore, they are interactive [Mat07, WWMO10b, WWMO10a] or *not* collusion safe [GA07, CT07, THJ08].

Since the PKG can generate a secret key for each user, he can decrypt the ciphertexts and obtain the original files if he knows the identity used to encrypt the files. Therefore, we assume that the PKG is honest and can be trusted by all users in an IBDDS scheme.

3.1.2 Our Contribution

In this chapter, we propose two IBDDS schemes in the standard model. In these schemes, for one request, a requester can only access one of the owner's files, instead of all files. In other words, an access permission (re-encryption key) is bound not only to the requester's identity but also to the requested ciphertext. The access permission can be determined by the owner without any help of the PKG. Furthermore, our schemes are secure against the collusion attacks. Although the first scheme is IND-CPA secure, the second scheme is IND-CCA2 secure. To the best of our knowledge, it is the first IBDDS schemes where an access permissions is made by the owner for an exact file and collusion attacks can be resisted in the standard model.

3.1.3 Chapter Organization

We propose the formal definition and security model of IBDDS in Section 3.2. In Section 3.3, an IND-CPA secure IBDDS scheme is proposed and proven. We propose an IND-CCA2 secure IBDDS scheme and prove its security in Section 3.4. Finally, Section 3.5 summaries this chapter.

3.2 Formal Definition and Security Model

In this section, we introduce the formal definition and security model of IBDDS schemes.

3.2.1 Formal Definition

There are four entities in an IBDDS scheme: private key generator PKG, data owner $\mathcal{O}$, proxy server $\mathcal{PS}$ and requester $\mathcal{R}$. PKG validates $\mathcal{R}$'s identities and issues secret keys to them. $\mathcal{O}$ encrypts his data under his identity and outsources the ciphertexts to the multiple $\mathcal{PS}$s. $\mathcal{PS}$s store the encrypted data and transfer a ciphertext for $\mathcal{O}$ to a ciphertext for $\mathcal{R}$ when he has obtained an access permission from $\mathcal{O}$. $\mathcal{R}$ authenticates himself to $\mathcal{O}$ and decrypts the re-encrypted ciphertext to access the data. An IBDDS scheme consists of the following algorithms:

$\mathsf{Setup}(1^\ell) \rightarrow (params, MSK)$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$ and a master secret key MSK.

$\mathsf{KeyGen}(params, ID, MSK) \rightarrow SK_{ID}$. The key generation algorithm takes as input the public parameters $params$, an identity ID and the master secret key MSK, and outputs a secret key SK_{ID} for the identity ID.

$\mathsf{Enc}(params, ID, M_i) \rightarrow CT_i$. The encryption algorithm takes as input the public parameters $params$, $\mathcal{O}$'s identity ID and a message M_i, and outputs the ciphertext $CT_i = (C_{i,1}, C_{i,2}{}^1)$, for $i = 1, 2, \cdots, m$. It sends the ciphertext $\{CT_i\}_{i=1}^m$ to the multiple $\mathcal{PS}$s.

[1] Suppose that the owner can know which file the requester wants to access from the second part of the cipertext.

$\mathsf{Query}(ID', SK_{ID'}, CT_i) \rightarrow AI$. The query algorithm takes as input $\mathcal{R}$'s identity ID', his secret key $SK_{ID'}$ and the ciphertext CT_i, and outputs an authentication information AI. It sends (ID', AI, CT_i) to the $\mathcal{PS}$. The $\mathcal{PS}$ redirects $(ID', AI, C_{i,2})$ to $\mathcal{O}$ with identity ID.

$\mathsf{Perm}(params, SK_{ID}, ID', C_{i,2}) \rightarrow RK_{ID \rightarrow ID'}$. The permission algorithm verifies the authentication information AI. If it is valid, this algorithm takes as inputs the public parameters $params$, $\mathcal{O}$'s secret key SK_{ID}, $\mathcal{R}$'s identity ID' and the partial ciphertext $C_{i,2}$, and outputs an access permission $RK_{ID \rightarrow ID'}$. It sends $RK_{ID \rightarrow ID'}$ to the $\mathcal{PS}$.

$\mathsf{Re\text{-}enc}(params, RK_{ID \rightarrow ID'}, ID', CT_i) \rightarrow CT_i'$. The re-encryption algorithm takes as input the public parameters $params$, the access permission $RK_{ID \rightarrow ID'}$, $\mathcal{R}$'s identity ID' and the ciphertext CT_i, and outputs a ciphertext $CT_i' = \mathsf{Enc}(params, ID', M_i)$ for $\mathcal{R}$ with identity ID'.

Dec. There are two decryption algorithms. One is for $\mathcal{O}$ and the other is for $\mathcal{R}$.

1. $\mathsf{Dec}_{\mathcal{O}}(params, SK_{ID}, CT_i) \rightarrow M_i$. The owner decryption algorithm takes as input the public parameters $params$, $\mathcal{O}$'s secret key SK_{ID} and the ciphertext CT_i, and outputs the message M_i.
2. $\mathsf{Dec}_{\mathcal{R}}(params, SK_{ID'}, CT_i') \rightarrow M_i$. The requester decryption algorithm takes as input the public parameters $params$, $\mathcal{R}$'s secret key $SK_{ID'}$ and the re-encrypted ciphertext CT_i', and outputs the message M_i.

Definition 3.1 *We say an identity-based distributed data storage scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}_{\mathcal{O}}(params, SK_{ID}, & \mathsf{Setup}(1^{\ell}) \rightarrow (params, MSK); \\ CT_i) \rightarrow M_i & \mathsf{KeyGen}(params, ID, MSK) \rightarrow SK_{ID}; \\ & \mathsf{Enc}(params, ID, M_i) \rightarrow CT_i \end{array}\right] = 1$$

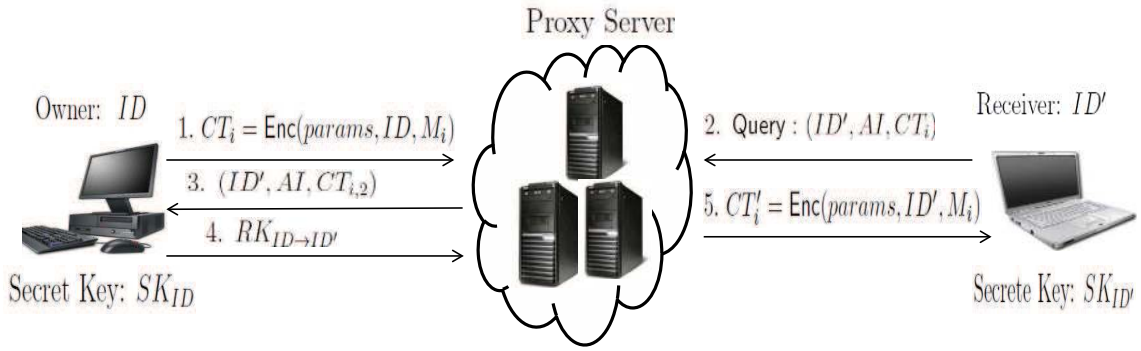

Figure 3.1: The Model of Identity-Based Distributed Data Storage Scheme

and

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}_{\mathcal{R}}(params, SK_{ID'}, CT'_i) \rightarrow M_i & \begin{array}{l}\mathsf{Setup}(1^\ell) \rightarrow (params, MSK);\\ \mathsf{KeyGen}(params, ID, MSK) \rightarrow SK_{ID};\\ \mathsf{KeyGen}(params, ID', MSK) \rightarrow SK_{ID'};\\ \mathsf{Perm}(params, SK_{ID}, ID', C_{i,2})\\ \rightarrow RK_{ID\rightarrow ID'};\\ \mathsf{Re\text{-}enc}(params, RK_{ID\rightarrow ID'}, ID', CT_i)\\ \rightarrow CT'_i\end{array}\end{array}\right] = 1$$

where the probability is taken over the random coins which all the algorithms in the scheme consumes.

Figure 3.1 explains the definition of an IBDDS scheme.

3.2.2 Security Model

The security model of IBDDS schemes is formalized by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and the master secret key MSK, and sends $params$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can adaptively query the following oracles:

1. Secret Key Queries. For a secret key query on an identity ID, $\mathcal{C}$ runs KeyGen $(params, ID, MSK)$ to generate a secret key SK_{ID}. $\mathcal{C}$ responds $\mathcal{A}$ with SK_{ID}. This query can be made multiple times.

2. Permission Queries. For an access permission query on $(ID, ID', C_{i,2})$, $\mathcal{C}$ runs $\mathsf{KeyGen}(params, ID, MSK)$ to generate the secret key SK_{ID} and $\mathsf{Perm}(params, SK_{ID}, ID', C_{i,2})$ to obtain $RK_{ID \to ID'}$. $\mathcal{C}$ responds $\mathcal{A}$ with $RK_{ID \to ID'}$. This query can be made multiple times.
3. Re-encryption Queries. For a re-encryption query on (ID, ID', CT_i), $\mathcal{C}$ runs $\mathsf{KeyGen}(params, ID, MSK)$ to generate a secret key SK_{ID}, and $\mathsf{Perm}(params, SK_{ID}, ID', C_{i,2})$ to obtain $RK_{ID \to ID'}$. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathsf{Re\text{-}enc}\ (params, RK_{ID \to ID'}, ID', CT_i)$. This query can be made multiple times.
4. Owner Decryption Queries. For an owner decryption query on (ID, CT_i), $\mathcal{C}$ runs $\mathsf{KeyGen}(params, ID, MSK)$ to extract the secret key SK_{ID}. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathsf{Dec}_{\mathcal{O}}(params, SK_{ID}, CT_i)$. This query can be made multiple times.
5. Requester Decryption Queries. For a requester decryption query on (ID, ID', CT_i), $\mathcal{C}$ runs $\mathsf{KeyGen}(params, ID, MSK)$ and $\mathsf{KeyGen}(params, ID', MSK)$ to generate secret keys SK_{ID} and SK'_{ID}, $\mathsf{Perm}\ (params, SK_{ID}, ID', C_{i,2})$ to obtain $RK_{ID \to ID'}$ and $\mathsf{Re\text{-}enc}(params, RK_{ID \to ID'}, ID', CT_i)$ to obtain CT'_i. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathsf{Dec}_{\mathcal{R}}(params, SK'_{ID}, CT'_i)$. This query can be made multiple times.

Challenge. $\mathcal{A}$ submits an identity ID^* and two messages M_0 and M_1 with equal length. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ computes $CT^* = \mathsf{Enc}(params, ID^*, M_b)$ and responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can query the oracles as in Phase 1 with the following restricts:

1. Secret Key Queries. $\mathcal{A}$ cannot query $KeyGen(params, ID^*, MSK)$.
2. Permission Queries. $\mathcal{A}$ cannot query $\mathsf{Perm}(ID^*, ID, C_2^*)$ and KeyGen $(params, ID, MSK)$.
3. Re-encryption Queries. $\mathcal{A}$ cannot query re-encryption on (ID^*, ID, CT^*), $\mathsf{Perm}(ID^*, ID, CT_2^*)$ and $\mathsf{KeyGen}(params, ID, MSK)$.
4. Owner Decryption Queries. $\mathcal{A}$ cannot query owner decryption on (ID^*, CT^*).
5. Requester Decryption Queries. $\mathcal{A}$ cannot query re-encryption on $(ID^*, ID,$ $CT^*)$ and requester decryption on $(ID, \widetilde{CT}^*)$, where $\widetilde{CT}^*$ is the re - encrypted ciphertext of CT^*.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 3.2 *An identity-based distributed data storage scheme is* $(T, q_1, q_2, q_3, q_4, q_5, \epsilon(\ell))$*-secure against chosen ciphertext attacks (or IND-CCA2) if no PPT adversary* $\mathcal{A}$ *making at most* q_1 *secret key queries,* q_2 *permission queries,* q_3 *re-encryption queries,* q_4 *owner decryption queries and* q_5 *requester decryption queries can win the game with the advantage*

$$Adv^{IND-CCA2}_{\mathcal{A}-IBDDS} = \left|\Pr[b' = b] - \frac{1}{2}\right| \geq \epsilon(\ell)$$

in the above model.

Definition 3.3 *An identity-based distributed data storage (IBDDS) scheme is* $(T, q_1, q_2, q_3, \epsilon(\ell))$*-secure against chosen plaintext attacks (or IND-CPA) if no PPT adversary* $\mathcal{A}$ *making at most* q_1 *secret key queries,* q_2 *permission queries and* q_3 *re-encryption queries can win the game with the advantage*

$$Adv^{IND-CPA}_{\mathcal{A}-IBDDS} = \left|\Pr[b' = b] - \frac{1}{2}\right| \geq \epsilon(\ell)$$

in the above model.

Theorem 3.1 *An identity-based distributed data storage scheme is unidirectional, nontransitive and collusion safe if it is secure against the chosen plaintext attacks (or IND-CPA) in the above model.*

Proof: Our proof is similar to that in [WWMO10b]. In the above IND-CPA security model, the adversary $\mathcal{A}$ can query secret key oracle, permission oracle and re-encryption oracle.

Collusion-safe. If the scheme is not collusion safe, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the IND-CPA security in the above security model. $\mathcal{A}$ can query a secret key $SK_{ID'}$ for an identity ID' and an access permission $RK_{ID^* \to ID'}$ from an identity ID^* to an identity ID'. After receiving the challenged ciphertext CT^* for the identity ID^*, if $\mathcal{A}$ can compute the secret key SK_{ID^*} from $SK_{ID'}$ and $RK_{ID^* \to ID'}$, $\mathcal{B}$ can use SK_{ID^*} to decrypt CT^* and obtain the message M_b. Hence, $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CPA security in the above model.

Nontransitive. If the scheme is transitive, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the IND-CPA security in the above security model. $\mathcal{A}$ can

query secret keys $SK_{ID'}$ and $SK_{ID''}$ for identities ID' and ID''. Furthermore, $\mathcal{A}$ can query access permissions $RK_{ID^* \to ID'}$ and $RK_{ID' \to ID''}$. After receiving the challenged ciphertext CT^* for the identity ID^*, if $\mathcal{A}$ can compute the permission $RK_{ID^* \to ID''}$ from $RK_{ID^* \to ID'}$ and $K_{ID' \to ID''}$, $\mathcal{B}$ can use $RK_{ID^* \to ID''}$ to transfer the ciphertext CT^* to a ciphertext $\widetilde{CT}$ for the identity ID''. Then, $\mathcal{B}$ can use $SK_{ID''}$ to decrypt $\widetilde{CT}$ and obtain the message M_b. So, $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CPA security in the above security model.

Unidirectional. If the scheme is not unidirectional in the above model, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the IND-CPA security in the above security model. $\mathcal{A}$ can query a secret key $SK_{ID'}$ for an identity ID' and an access permission $RK_{ID' \to ID^*}$ from ID' to an identity ID^*. After receiving the challenged ciphertext CT^* for ID^*, if $\mathcal{A}$ can use $RK_{ID' \to ID^*}$ to transfer CT^* to a ciphertext $\widetilde{CT}$ for ID', $\mathcal{B}$ can use the secret key $SK_{ID'}$ to decrypt $\widetilde{CT}$ and obtain the message M_b. Therefore, $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CPA security in the above model. □

3.3 Identity-based Distributed Data Storage I

In this section, we propose an IBDDS scheme IBDDS-I which is secure against chosen plaintext attacks (or IND-CPA). At first, the file owner encrypts his files and outsources the ciphertexts to the proxy servers. The proxy servers validate the ciphertexts and store them for the owner. For one request, the requester uses his secret key to compute an authentication information (AI) and sends it to the proxy server. The proxy server sends the identity of the requester, AI and the partial intended ciphertext to the owner. Suppose that the owner can detect which file the requester wants to access from the partial ciphertext. Subsequently, the owner validates the received AI. If it is valid, the owner computes an access permission (re-encryption key) using his secret key, the partial ciphertext and the identity of the requester, and sends it to the proxy server. Otherwise, the access request is denied. When receiving an access permission from the owner, the proxy sever re-encrypts the intended ciphertext to a ciphertext for the requester. Finally, the requester can decrypt the re-encrypted ciphertext by his secret key and obtains the original file.

The specific protocol of our scheme IBDDS-I is demonstrated in Figure 3.2. This scheme can be seen as an extension of Water's IBE [Wat05].

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let $g, h, \eta, \mathfrak{g}$ and $\mathfrak{h}$ be the generators of $\mathbb{G}$, $u_0 \stackrel{R}{\leftarrow} \mathbb{G}$ and $\mathbf{U} = (u_1, u_2, \cdots, u_n)$ where $u_i \stackrel{R}{\leftarrow} \mathbb{G}$ for $i = 1, 2, \cdots, n$. It chooses $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and sets $g_1 = g^\alpha$ and $g_2 = \mathfrak{g}^\alpha$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, \eta, \mathfrak{g}, \mathfrak{h}, u_0, \mathbf{U}, g_1, g_2)$ and the master secret key is η^α.

KeyGen. Let ID denote an identity which is an n bit string, ID_i be the ith bit of ID and $\mathbf{I}$ be a set which consists of all the index i with $ID_i = 1$. This algorithm takes as input the master secret key η^α and an identity ID, and computes

$$K_{ID,1} = \eta^\alpha (u_0 \prod_{i \in \mathbf{I}} u_i)^{r_{ID}},\ K_{ID,2} = g^{r_{ID}} \text{ and } K_{ID,3} = \mathfrak{g}^{r_{ID}}.$$

The secret key for a user $\mathcal{U}$ with identity ID is $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$. This secret key can be verified by

$$e(K_{ID,1}, g) \stackrel{?}{=} e(\eta, g_1) \cdot e((u_0 \prod_{i \in \mathbf{I}} u_i), K_{ID,2}) \text{ and } e(K_{ID,2}, \mathfrak{g}) \stackrel{?}{=} e(g, K_{ID,3}).$$

Encryption. To encrypt messages $\{M_1, M_2, \cdots, M_m\}$, the owner $\mathcal{O}$ with identity ID chooses $s_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$C_{i,1} = M_i \cdot e(g_1, \eta)^{s_i},\ C_{i,2} = g^{s_i} \text{ and } C_{i,3} = (u_0 \prod_{i \in \mathbf{I}} u_i)^{s_i}$$

for $i = 1, 2, \cdots, m$. The ciphertext for the message M_i is $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$. $\mathcal{O}$ sends $\{CT_1, CT_2, \cdots, CT_m\}$ to the proxy servers $\mathcal{PS}$s. $\mathcal{PS}$s validate the ciphertexts by checking

$$e((u_0 \prod_{i \in \mathbf{I}} u_i), C_{i,2}) \stackrel{?}{=} e(C_{i,3}, g)$$

for $i = 1, 2, \cdots, m$. If the equations hold, $\mathcal{PS}$s stores the ciphertexts $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$ for $\mathcal{O}$. Otherwise, the ciphertexts are rejected.

Query. If a requester $\mathcal{R}$ with identity ID' wants to access a ciphertext CT_i, he chooses $t \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $K'_{ID',1} = K_{ID',1} \mathfrak{h}^t$ and $\Gamma = \mathfrak{g}^t$. He sends $(ID', K'_{ID',1}, K_{ID',3}, \Gamma)$ to the $\mathcal{PS}$ who stores the ciphertext CT_i. Then, the $\mathcal{PS}$ redirects $(ID', K'_{ID',1}, K_{ID',3}, \Gamma, C_{i,2})$ to $\mathcal{O}$.

Permission. $\mathcal{O}$ checks whether $\mathcal{R}$ has been authenticated by verifying

$$e(K'_{ID',1}, \mathfrak{g}) \stackrel{?}{=} e(\eta, g_2) \cdot e((u_0 \prod_{i \in \mathbf{I}'} u_i), K_{ID',3}) \cdot e(\mathfrak{h}, \Gamma).$$

If it holds, $\mathcal{O}$ chooses $\beta, \rho \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$D_1 = \frac{K_{ID,1}}{K'_{ID',1} \cdot \Gamma^{\rho}} \cdot (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta},\ D_2 = e(C_{i,2}, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta} \text{ and } D_3 = \mathfrak{g}^{\rho}.$$

Then, $\mathcal{O}$ sends $(D_1, D_2, D_3, K_{ID,2})$ to the $\mathcal{PS}$.

Re-encryption. When receiving $(D_1, D_2, D_3, K_{ID,2})$ from $\mathcal{O}$, the $\mathcal{PS}$ computes the re-encrypted ciphertext as

$$C'_{i,1} = D_2 \cdot C_{i,1},\ C'_{i,2} = C_{i,2},\ C'_{i,3} = C_{i,3},$$

$$C'_{i,4} = D_1,\ C'_{i,5} = D_3 \text{ and } C'_{i,6} = K_{ID,2}.$$

The $\mathcal{PS}$ responds $\mathcal{R}$ with $CT'_i = (C'_{i,1}, C'_{i,2}, C'_{i,3}, C'_{i,4}, C'_{i,5}, C'_{i,6})$.

Decryption.

1. To decrypt a ciphertext $CT_i = (C_{i,1}, C_{i,2}, C_{i,3})$, $\mathcal{O}$ computes

$$M_i = C_{i,1} \cdot \frac{e(K_{ID,2}, C_{i,3})}{e(K_{ID,1}, C_{i,2})}.$$

2. To decrypt a re-encrypted ciphertext $CT'_i = (C'_{i,1}, C'_{i,2}, C'_{i,3}, C'_{i,4}, C'_{i,5}, C'_{i,6})$, $\mathcal{R}$ computes

$$K_1 = K'_{ID',1} \cdot C'^{t}_{i,5} \cdot C'_{i,4}$$

and

$$M_i = C'_{i,1} \cdot \frac{e(C'_{i,6}, C'_{i,3})}{e(K_1, C'_{i,2})}.$$

Figure 3.2: IBDDS-I: Identity-Based Distributed Data Storage I

Correctness.

We have

$$\begin{aligned}
& C_{i,1} \cdot \frac{e(K_{ID,2}, C_{i,3})}{e(K_{ID,1}, C_{i,2})} \\
= \; & M_i \cdot e(g_1, \eta)^{s_i} \frac{e(g^{r_{ID}}, (u_0 \prod_{i \in \mathbf{I}} u_i)^{s_i})}{e(\eta^{\alpha} (u_0 \prod_{i \in \mathbf{I}} u_i)^{r_{ID}}, g^{s_i})} \\
= \; & M_i \cdot e(g_1, \eta)^{s_i} \frac{e(g^{r_{ID}}, (u_0 \prod_{i \in \mathbf{I}})^{s_i})}{e(g_1, \eta)^{s_i} \cdot e(g^{r_{ID}}, (u_0 \prod_{i \in \mathbf{I}} u_i)^{s_i})} \\
= \; & M_i \cdot e(g_1, \eta)^{s_i} \cdot \frac{1}{e(g_1, \eta)^{s_i}} \\
= \; & M_i,
\end{aligned}$$

$$\begin{aligned}
K_1 &= K'_{ID',1} \cdot C'^{t}_{i,5} \cdot C'_{i,4} \\
&= K'_{ID',1} \cdot \mathfrak{g}^{\rho t} \cdot \frac{K_{ID,1}}{K'_{ID',1} \cdot \Gamma^{\rho}} \cdot (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta} \\
&= K_{ID,1} \cdot (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta} \\
&= \eta^{\alpha} (u_0 \prod_{i \in \mathbf{I}} u_i)^{r_{ID}} (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta}
\end{aligned}$$

and

$$\begin{aligned}
C'_{i,1} &= D_2 \cdot C_{i,1} \\
&= M_i \cdot e(g_1, \eta)^{s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta s_i}.
\end{aligned}$$

Therefore

$$\begin{aligned}
& C'_{i,1} \cdot \frac{e(C'_{i,6}, C'_{i,3})}{e(K_1, C_{i,2})} \\
= \; & C'_{i,1} \cdot \frac{e(g^{r_{ID}}, (u_0 \prod_{i \in \mathbf{I}} u_i)^{s_i})}{e(\eta^{\alpha} (u_0 \prod_{i \in \mathbf{I}} u_i)^{r_{ID}} (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta}, g^{s_i})} \\
= \; & C'_{i,1} \cdot \frac{e(g, (u_0 \prod_{i \in \mathbf{I}} u_i))^{r_{ID} s_i}}{e(g_1, \eta)^{s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}} u_i))^{r_{ID} s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta s_i}} \\
= \; & C'_{i,1} \cdot \frac{1}{e(g_1, \eta)^{s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta s_i}} \\
= \; & M_i \cdot \frac{e(g_1, \eta)^{s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta s_i}}{e(g_1, \eta)^{s_i} \cdot e(g, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta s_i}} \\
= \; & M_i
\end{aligned}$$

Theorem 3.2 *Our identity-based distributed data storage scheme IBDDS-I is* $(T, q_1, q_2, q_3, \epsilon(\ell))$*-secure against chose plaintext attacks (or IND-CPA) if the* $(T', \epsilon'(\ell))$*-decisional bilinear Diffie-Hellman assumption holds in the bilinear group* $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ *where*

$$T' = T + \Theta(T) \quad \textit{and} \quad \epsilon'(\ell) = \frac{\epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3)(n+1)}.$$

Proof: The proof is similar to that in Waters's IBE [Wat05], except that the permission queries and re-encryption queries from the adversary must be answered.

Suppose that there exists a PPT adversary $\mathcal{A}$ who can $(T, q_1, q_2, q_3, \epsilon(\ell))$ break the IND-CPA security of our IBDDS-I scheme, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the DBDH assumption as follows. The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and chooses a generator $g \xleftarrow{R} \mathbb{G}$. It flips an unbiased coin μ with $\{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, b^c, e(g, g)^{abc})$ to $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \xleftarrow{R} \mathbb{Z}_p$. The algorithm $\mathcal{B}$ will output his guess μ' on μ.

Setup. $\mathcal{B}$ sets $\sigma = 4(q_1 + 2q_2 + 2q_3)$ and chooses an integer $\nu \xleftarrow{R} [n]$. It uniformly selects two integrity vectors $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$ and $\Phi = \{\phi_1, \phi_2, \cdots, \phi_n\}$, where $\pi_i \xleftarrow{R} [\sigma - 1]$ and $\phi_i \xleftarrow{R} \mathbb{Z}_p$ for $i = 1, 2, \cdots, n$. It choose $\pi_0 \xleftarrow{R} [\sigma - 1]$ and $\phi_0 \xleftarrow{R} \mathbb{Z}_p$. Then, $\mathcal{B}$ defines three functions:

$$P(ID) = (p - \sigma\nu) + \pi_0 + \sum_{i \in \mathbf{I}} \pi_i,$$

$$Q(ID) = \phi_0 + \sum_{i \in \mathbf{I}} \phi_i$$

and

$$R(ID) = \begin{cases} 0, & \text{if } \pi_0 + \sum_{i \in \mathbf{I}} \pi_i \equiv 0 \pmod{\sigma} \\ 1, & \text{if } \pi_0 + \sum_{i \in \mathbf{I}} \pi_i \not\equiv 0 \pmod{\sigma} \end{cases}$$

$\mathcal{B}$ sets $g_1 = A$, $\eta = B$, $\mathfrak{g} = g^\theta$, $g_2 = A^\theta$, $u_0 = \eta^{p - \sigma\nu + \pi_0} g^{\phi_0}$, $u_i = \eta^{\pi_i} g^{\phi_i}$ and $\mathbf{U} = \{u_i\}_{i=1}^n$, where $\theta \xleftarrow{R} \mathbb{Z}_p$. It chooses $\mathfrak{h} \xleftarrow{R} \mathbb{G}$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, \eta, \mathfrak{g}, \mathfrak{h}, u_0, \mathbf{U}, g_1, g_2)$, while the master secret key is $\eta^a = g^{ab}$. The distribution of these parameters is identical to that in the real protocol.

Phase 1.

1. Secret Key Queries. For a secret key query on an identity ID, $\mathcal{B}$ checks $R(ID) \stackrel{?}{=} 1$.

   (a) If $R(ID) = 1$, $\mathcal{B}$ chooses $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes

   $$K_{ID,1} = A^{\frac{-Q(ID)}{P(ID)}}(\pi_0 \prod_{i \in \mathbf{I}} \pi_i)^r, \tag{3.1}$$

   $$K_{ID,2} = A^{\frac{-1}{P(ID)}} g^r \tag{3.2}$$

   and

   $$K_{ID,3} = K_{ID,2}^{\theta}. \tag{3.3}$$

   $\mathcal{B}$ responds with $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$.

   (b) If $R(ID) = 0$, $\mathcal{B}$ aborts and outputs his guess μ' randomly.

   We claim that the secret key is generated correctly.

   $$\begin{aligned} K_{ID,1} &= A^{\frac{-Q(ID)}{P(ID)}}(u_0 \prod_{i \in \mathbf{I}} u_i)^r \\ &= g^{\frac{-aQ(ID)}{P(ID)}}(g^{bP(ID)+Q(ID)})^r \\ &= (g^{bP(ID)+Q(ID)})^{\frac{-a}{P(ID)}} g^{ab}(g^{bP(ID)+Q(ID)})^r \\ &= g^{ab}(g^{bP(ID)+Q(ID)})^{r-\frac{a}{P(ID)}} \\ &= \eta^a(u_0 \prod_{i \in \mathbf{I}} u_i)^{r-\frac{a}{P(ID)}} \end{aligned}$$

   Let $\hat{r} = r - \frac{a}{P(ID)}$, we have

   $$K_{ID,1} = \eta^a(u_0 \prod_{i \in \mathbf{I}} u_i)^{\hat{r}},$$

   $$K_{ID,2} = A^{\frac{-1}{P(ID)}} g^r = g^{r-\frac{a}{P(ID)}} = g^{\hat{r}}$$

   and

   $$K_{ID,3} = K_{ID,2}^{\theta} = g^{\theta\hat{r}} = \mathfrak{g}^{\hat{r}}.$$

   Therefore, the secret key is created correctly.

2. Permission Queries. For an access permission on (ID, ID', C_2), $\mathcal{B}$ checks whether he has generated secret keys for identities ID and ID'. If he has not generated secret keys for ID and ID', $\mathcal{B}$ checks whether $R(ID) = R(ID)' = 1$.

(a) If it holds, $\mathcal{B}$ computes $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$ and $K_{ID'} = (K_{ID',1}, K_{ID',2}, K_{ID',3})$. Then, he can compute an access permission (the re-encryption key) as follows. $\mathcal{B}$ chooses $t, \beta, \rho \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$K'_{ID',1} = K_{ID',1}\mathfrak{h}^t, \tag{3.4}$$

$$\Gamma = \mathfrak{g}^t, \tag{3.5}$$

$$D_1 = \frac{K_{ID,1}}{K'_{ID',1}\Gamma^\rho} \cdot (u_0 \prod_{i \in \mathbf{I}'} u_i)^\beta \tag{3.6}$$

$$D_2 = e(C_2, (u_0 \prod_{i \in \mathbf{I}'} u_i))^\beta, \tag{3.7}$$

and

$$D_3 = \mathfrak{g}^\rho. \tag{3.8}$$

$\mathcal{B}$ sends $(D_1, D_2, D_3, K_{ID,2})$ to $\mathcal{A}$.

(b) Otherwise, $\mathcal{B}$ aborts the simulation and outputs his guess μ' randomly.

3. Re-encryption Queries. For a re-encryption query on (ID, ID', C), $\mathcal{B}$ checks whether he has generated an access permission $(D_1, D_2, D_3, K_{ID,2})$ from identity ID to identity ID'. If he has not generated an access permission from ID to ID', $\mathcal{B}$ generate $(D_1, , D_2, D_3, K_{ID,2})$ as above. Otherwise, $\mathcal{B}$ can compute

$$C'_1 = D_2 \cdot C_1,\ C'_2 = C_2,\ C'_3 = C_3,\ C'_4 = D_1,\ C'_5 = D_3 \text{ and } C'_6 = K_{ID,2}.$$

$\mathcal{B}$ responds $\mathcal{A}$ with the re-encrypted ciphertext $CT' = (C'_1, C'_2, C'_3, C'_4, C'_5, C'_6)$.

Challenge. $\mathcal{A}$ submits an identity ID^* and two messages M_0 and M_1 with the equal length. $\mathcal{B}$ checks $R(ID^*) \overset{?}{=} 0$.

1. If $R(ID^*) = 1$, $\mathcal{B}$ aborts and outputs his guess μ' randomly.

2. If $R(ID^*) = 0$, $\mathcal{B}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $\omega \in \{0, 1\}$. $\mathcal{B}$ computes $C^*_1 = M_\omega \cdot Z$, $C^*_2 = C = g^c$ and $C^*_3 = C^{Q(ID^*)} = (u_0 \prod_{i \in \mathbf{I}^*} u_i)^c$. $\mathcal{B}$ sends the ciphertext $CT^* = (C^*_1, C^*_2, C^*_3)$ to $\mathcal{A}$.

Phase 2. Phase 1 is repeated with the following restrictions.

1. Secret Key Queries. $\mathcal{A}$ cannot query secret key for the identity ID^*.
2. Permission Queries. $\mathcal{A}$ cannot query an access permission on (ID^*, ID, C_2^*) and secret keys for identities ID.
3. Re-encryption Queries. $\mathcal{A}$ cannot query re-encryption on (ID^*, ID, C^*), permission on (ID^*, ID, C_2^*) and secret key for ID.

Guess. $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs $\mu' = 0$; otherwise $\mathcal{B}$ outputs $\mu' = 1$.

As shown above, the public parameters and secret keys created in the simulation paradigm are identical to those created in the real protocol. $\mathcal{B}$ does not abort the simulation if and only if the secret keys can be generated correctly and $R(ID^*) = 0$. In q_1 secret key queries, q_2 permission queries and q_3 re-encryption queries, $\mathcal{B}$ needs to create at most $q_1 + 2q_2 + 2q_3$ secret keys. Let $\rho = q_1 + 2q_2 + 2q_3$ and $\{ID^{(1)}, ID^{(2)}, \cdots, ID^{(\rho)}\}$ be the identities selected by $\mathcal{A}$ to query the oracles in our scheme. We compute the bound with which $\mathcal{B}$ does not abort the simulation. This bound is computed using the method exploited in [Wat05].

$$
\begin{aligned}
\Pr\ & [\overline{Abort}] \\
= \ & \Pr\left[(\bigwedge_{i=1}^{\rho} R(ID^{(i)}) = 1) \bigwedge \pi_0 + \sum_{j \in \mathbf{I}^*} \pi_j = k\sigma)\right] \\
= \ & \left(1 - \Pr\left[\bigvee_{i=1}^{\rho} R(ID^{(i)}) = 0\right]\right) \Pr\left[\pi_0 + \sum_{j \in \mathbf{I}^*} \pi_j = k\sigma | \bigwedge_{i=1}^{q} (R(ID^{(i)}) = 1)\right] \\
\geq \ & \left(1 - \sum_{i=1}^{\rho} \Pr\left[R(ID^{(i)}) = 0\right]\right) \Pr\left[\pi_0 + \sum_{j \in \mathbf{I}^*} \pi_j = k\sigma | \bigwedge_{i=1}^{\rho} (R(ID^{(i)}) = 1)\right] \\
= \ & (1 - \frac{\rho}{\sigma}) \Pr\left[\pi_0 + \sum_{j \in \mathbf{I}^*} \pi_j = k\sigma | \bigwedge_{i=1}^{\rho} (R(ID^{(i)}) = 1)\right] \\
= \ & \frac{1}{n+1}(1 - \frac{\rho}{\sigma}) \Pr\left[R(ID^*) = 0 | \bigwedge_{i=1}^{\rho} R(ID^{(i)}) = 1\right] \\
= \ & \frac{1}{n+1}(1 - \frac{\rho}{\sigma}) \frac{\Pr[R(ID^*) = 0]}{\Pr\left[\bigwedge_{i=1}^{\rho} R(ID^{(i)}) = 1\right]} \Pr\left[\bigwedge_{i=1}^{\rho} R(ID^{(i)}) = 1 | R(ID^*) = 0\right] \\
\geq \ & \frac{1}{\sigma(n+1)}(1 - \frac{\rho}{\sigma}) \Pr\left[\bigwedge_{i=1}^{\rho} (R(ID^{(i)}) = 1) | R(ID^*) = 0\right] \\
= \ & \frac{1}{\sigma(n+1)}(1 - \frac{\rho}{\sigma}) \left(1 - \Pr\left[\bigvee_{i=1}^{\rho} R(ID^{(i)}) = 0 | R(ID^*) = 0)\right]\right) \\
\geq \ & \frac{1}{\sigma(n+1)} \left(1 - \frac{\rho}{\sigma}\right)^2 \\
\geq \ & \frac{1}{\sigma(n+1)}(1 - 2\frac{\rho}{\sigma}) \\
\geq \ & \frac{1}{8\rho(n+1)} \\
= \ & \frac{1}{8(q_1 + 2q_2 + 2q_3)(n+1)}.
\end{aligned}
$$

Now, we compute the probability $\Pr[\mu' = \mu]$.

$$\begin{aligned}
\Pr & [\mu' = \mu] \\
= & \Pr\left[\mu' = \mu | \overline{Abort}\right] \Pr\left[\overline{Abort}\right] + \Pr\left[\mu' = \mu | Abort\right] \Pr[Abort] \\
= & \frac{1}{2}\left(\Pr[\overline{Abort} | \omega' = \omega] \Pr[\omega' = \omega] - \Pr[\overline{Abort} | \omega' \neq \omega] \Pr[\omega' \neq \omega]\right) + \frac{1}{2} \\
= & \frac{1}{2}\left(\Pr[\overline{Abort} | \omega' = \omega](\frac{1}{2} + \epsilon(\ell)) - \Pr[\overline{Abort} | \omega' \neq \omega](\frac{1}{2} - \epsilon(\ell))\right) + \frac{1}{2} \\
\geq & \frac{1}{2} \times \frac{3}{2} \times \frac{\epsilon(\ell)}{8(q_1 + 2q_2 + 2q_3)(n+1)} + \frac{1}{2} \\
= & \frac{3 \cdot \epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3)(n+1)} + \frac{1}{2}.
\end{aligned}$$

Therefore, the advantage with which $\mathcal{B}$ can break the DBDH assumption is

$$\left|\Pr[\mu' = \mu] - \frac{1}{2}\right| \geq \frac{3 \cdot \epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3)(n+1)} + \frac{1}{2} - \frac{1}{2} \geq \frac{\epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3)(n+1)}.$$

3.4 Identity-based Distributed Data Storage II

In some complex network environments, such as cloud computing and distributed systems, IND-CPA security cannot satisfy the application requirement as the active adversaries may potentially modify the transmitted ciphertexts. To provide strong security for encryption schemes, chosen-ciphertext security (IND-CCA2) was proposed. This notion can be applied in the presence of active adversaries who can modify the ciphertexts. Schemes which are IND-CCA2 secure can be used as primitives to construct high-level protocols. Therefore, an IBDDS scheme with strong security (IND-CCA2) is desirable. In this section, we propose an IND-CCA2 secure IBDDS scheme IBDDS-II by introducing an existentially unforgeable one-time signature scheme to the IBDDS-I scheme. This idea is derived from [CHK04]. Our IBDDS-II scheme is demonstrated in Figure 3.3.

Correctness. This is the same as in the scheme IBDDS-I.

Theorem 3.3 *Our identity-based distributed data storage scheme IBDDS-II is* $(T, q_1, q_2, q_3, q_4, q_5, \epsilon(\ell))$*-secure against chose ciphertext attacks (or IND-CCA2) if the one-time signature scheme is* $(T', 1, \epsilon'(\ell))$*-existentially unforgeable against adaptive chosen message attacks (EU-CMA) and the* $(T'', \epsilon''(\ell))$ *decisional bilinear Diffie-Hellman*

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let $g, h, \eta, \mathfrak{g}$ and $\mathfrak{h}$ be the generators of $\mathbb{G}$, $u_0 \stackrel{R}{\leftarrow} \mathbb{G}$ and $\mathbf{U} = (u_1, u_2, \cdots, u_n)$ where $u_i \stackrel{R}{\leftarrow} \mathbb{G}$ for $i = 1, 2, \cdots, n$. It chooses $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and sets $g_1 = g^\alpha$ and $g_2 = \mathfrak{g}^\alpha$. It generates an one-time signature scheme $\mathcal{SG}(1^\ell) \rightarrow (\mathsf{SKeyGen}, \mathsf{Sign}, \mathsf{Verify})$, where $\mathsf{SKeyGen}(1^\ell) \rightarrow (sk, vk)$. Let $\mathcal{H} : vk \rightarrow \mathbb{Z}_p$ be a hash function. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, \eta, \mathfrak{g}, \mathfrak{h}, u_0, \mathbf{U}, \mathcal{H}, \mathsf{Sign}, \mathsf{Verify}, g_1, g_2)$ and the master secret key is η^α.

KeyGen. Let ID denote an identity which is an n bit string, ID_i be the ith bit of ID and $\mathbf{I}$ be a set which consists of all the index i with $ID_i = 1$. This algorithm takes as input the master secret key η^α and an identity ID, and computes

$$K_{ID,1} = \eta^\alpha (u_0 \prod_{i \in \mathbf{I}} u_i)^{r_{ID}},\ K_{ID,2} = g^{r_{ID}} \text{ and } K_{ID,3} = \mathfrak{g}^{r_{ID}}.$$

The secret key for the user $\mathcal{U}$ with identity ID is $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$. This secret key can be verified by

$$e(K_{ID,1}, g) \stackrel{?}{=} e(\eta, g_1) \cdot e((u_0 \prod_{i \in \mathbf{I}} u_i), K_{ID,2}) \text{ and } e(K_{ID,2}, \mathfrak{g}) \stackrel{?}{=} e(g, K_{ID,3}).$$

Encryption. To encrypt messages $M_i \in \{M_1, M_2, \cdots, M_m\}$, the owner $\mathcal{O}$ with identity ID runs $\mathsf{SKeyGen}(1^\ell) \rightarrow (sk, vk)$, chooses $s_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$C_{i,1} = M_i \cdot e(g_1, \eta)^{s_i},\ C_{i,2} = g^{s_i},\ C_{i,3} = (u_0 \prod_{i \in \mathbf{I}} u_i)^{s_i},\ C_{i,4} = (g^{\mathcal{H}(vk)} \mathfrak{g})^{s_i}$$

and

$$\sigma_i = \mathsf{Sign}(sk, C_{i,2}, C_{i,3}, C_{i,4})$$

for $i = 1, 2, \cdots, m$. The ciphertext for the message M_i is $CT_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, \sigma_i, vk)$.

$\mathcal{O}$ sends $\{CT_1, CT_2, \cdots, CT_m\}$ to the proxy servers $\mathcal{PS}$s. $\mathcal{PS}$s validate the ciphertexts by verifying

$$\sigma_i \stackrel{?}{=} \mathsf{Verify}(vk, C_{i,2}, C_{i,3}, C_{i,4}),\ e((u_0 \prod_{i \in \mathbf{I}} u_i), C_{i,2}) \stackrel{?}{=} e(C_{i,3}, g)$$

and

$$e(g, C_{i,4}) \stackrel{?}{=} e(C_{i,2}, (g^{\mathcal{H}(vk)} \mathfrak{g}))$$

for $i = 1, 2, \cdots, m$. If the equations hold, $\mathcal{PS}s$ store the ciphertexts $\{CT_i\}_{i=1}^m$ for $\mathcal{O}$. Otherwise, $\mathcal{PS}$s reject the ciphertexts.

Query. If a requester $\mathcal{R}$ with identity ID' wants to access a ciphertext CT_i, he chooses $t \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $K'_{ID',1} = K_{ID',1}\mathfrak{h}^t$ and $\Gamma = \mathfrak{g}^t$. He sends $(ID', K'_{ID',1}, K_{ID',3}, \Gamma)$ to the $\mathcal{PS}$ who stores the ciphertext CT_i. Then, the $\mathcal{PS}$ redirects $(ID', K'_{ID',1}, K_{ID',3}, \Gamma, C_{i,2})$ to $\mathcal{O}$.

Permission. $\mathcal{O}$ checks whether $\mathcal{R}$ has been authenticated by verifying

$$e(K'_{ID',1}, \mathfrak{g}) \overset{?}{=} e(\eta, g_2) \cdot e((u_0 \prod_{i \in \mathbf{I}'} u_i), K_{ID',3}) \cdot e(\mathfrak{h}, \Gamma).$$

If it holds, $\mathcal{O}$ chooses $\beta, \rho \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$D_1 = \frac{K_{ID,1}}{K'_{ID',1} \cdot \Gamma^{\rho}} \cdot (u_0 \prod_{i \in \mathbf{I}'} u_i)^{\beta},\ D_2 = e(C_{i,2}, (u_0 \prod_{i \in \mathbf{I}'} u_i))^{\beta} \text{ and } D_3 = \mathfrak{g}^{\rho}.$$

$\mathcal{O}$ sends an access permission $(D_1, D_2, D_3, K_{ID,2})$ to the $\mathcal{PS}$.

Re-encryption. Receiving $(D_1, D_2, D_3, K_{ID,2})$ from $\mathcal{O}$, the $\mathcal{PS}$ computes the re-encrypted ciphterxt as

$$C'_{i,1} = D_2 \cdot C_{i,1},\ C'_{i,2} = C_{i,2},\ C'_{i,3} = C_{i,3},\ C'_{i,4} = C_{i,4},$$

$$C'_{i,5} = D_1,\ C'_{i,6} = D_3, C'_{i,7} = K_{ID,2} \text{ and } \sigma'_i = \sigma_i.$$

The $\mathcal{PS}$ responds $\mathcal{R}$ with $CT'_i = (C'_{i,1}, C'_{i,2}, C'_{i,3}, C'_{i,4}, C'_{i,5}, C'_{i,6}, C'_{i,7}, \sigma'_i, vk)$.

Decryption.

1. To decrypt a ciphertext $CT_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, \sigma_i, vk)$, $\mathcal{O}$ checks $\sigma_i \overset{?}{=} \mathsf{Verify}(vk, C_{i,2}, C_{i,3}, C_{i,4})$. If it holds, $\mathcal{O}$ computes

$$M_i = C_{i,1} \cdot \frac{e(K_{ID,2}, C_{i,3})}{e(K_{ID,1}, C_{i,2})}.$$

2. To decrypt a re-encrypted ciphertext $CT'_i = (C'_{i,1}, C'_{i,2}, C'_{i,3}, C'_{i,4}, C'_{i,5}, C'_{i,6}, C'_{i,7}, \sigma'_i, vk)$, $\mathcal{R}$ checks $\sigma'_i \overset{?}{=} \mathsf{Verify}(vk, C'_{i,2}, C'_{i,3}, C'_{i,4})$. If it holds, $\mathcal{R}$ computes

$$K_1 = K'_{ID',1} \cdot C'^{t}_{i,6} \cdot C'_{i,5}$$

and

$$M_i = C'_{i,1} \cdot \frac{e(C'_{i,7}, C'_{i,3})}{e(K_1, C'_{i,2})}.$$

Figure 3.3: IBDDS-II: Identity-Based Distributed Data Storage II

assumption holds in the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ *where*

$$T' = T + T' + \Theta(T + T')$$

and

$$\epsilon(\ell) = \epsilon'(\ell) + 32(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5)(n+1)\epsilon''(\ell)$$

Proof: Suppose that there exists a PPT adversary $\mathcal{A}$ can break the IND-CCA2 security of our scheme IBDDS-II with the advantage $Adv_{\mathcal{A}} > \epsilon'(\ell) + 32(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5)(n+1)\epsilon''(\ell)$, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to forge a signature or break the DBDH assumption as follows. The challenger $\mathcal{C}$ generates the bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and chooses a generator $g \overset{R}{\leftarrow} \mathbb{G}$. It flips an unbiased coin μ with $\{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, b^c, e(g, g)^{abc})$ to $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \overset{R}{\leftarrow} \mathbb{Z}_p$. $\mathcal{B}$ will outputs his guess μ' on μ.

Setup. $\mathcal{B}$ sets $\sigma = 4(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5)$ and chooses an integer $\nu \overset{R}{\leftarrow} [n]$. It uniformly selects two integrity vectors $\Pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$ and $\Phi = \{\phi_1, \phi_2, \cdots, \phi_n\}$, where $\pi_i \overset{R}{\leftarrow} [\sigma - 1]$ and $\phi_i \overset{R}{\leftarrow} \mathbb{Z}_p$ for $i = 1, 2, \cdots, n$. It chooses $\pi_0 \overset{R}{\leftarrow} [\sigma - 1]$ and $\phi_0 \overset{R}{\leftarrow} \mathbb{Z}_p$. Then, $\mathcal{B}$ defines three functions:

$$P(ID) = (p - \sigma\nu) + \pi_0 + \sum_{i \in \mathbf{I}} \pi_i,$$

$$Q(ID) = \phi_0 + \sum_{i \in \mathbf{I}} \phi_i$$

and

$$R(ID) = \begin{cases} 0, & \text{if } \pi_0 + \sum_{i \in \mathbf{I}} \pi_i \equiv 0 \pmod{\sigma} \\ 1, & \text{if } \pi_0 + \sum_{i \in \mathbf{I}} \pi_i \not\equiv 0 \pmod{\sigma} \end{cases}$$

$\mathcal{B}$ sets $g_1 = A$, $\eta = B$, $\mathfrak{g} = g^\theta$, $g_2 = A^\theta$, $u_0 = \eta^{p - \sigma\nu + \pi_0} g^{\phi_0}$, $u_i = \eta^{\pi_i} g^{\phi_i}$ and $\mathbf{U} = \{u_i\}_{i=1}^n$, where $\theta \overset{R}{\leftarrow} \mathbb{Z}_p$. It chooses $\mathfrak{h} \overset{R}{\leftarrow} \mathbb{G}$ and an one-time signature scheme $\mathcal{SG}(1^\ell) \rightarrow (\mathsf{SKeyGen}, \mathsf{Sign}, \mathsf{Verify})$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, \eta, \mathfrak{g}, \mathfrak{h}, u_0, \mathbf{U}, \mathsf{Sign}, \mathsf{Verify}, g_1, g_2)$, while the master secret key is $\eta^a = g^{ab}$. The distribution of these parameters is identical to those in the real protocol.

Phase 1.

1. Secret Key Queries. For a secret key query on an identity ID, $\mathcal{B}$ checks $R(ID) \stackrel{?}{=} 1$.
   (a) If $R(ID) = 0$, $\mathcal{B}$ aborts the simulation and outputs his guess μ' randomly.
   (b) If $R(ID) = 1$, $\mathcal{B}$ generates a secret key for ID using the techniques in (3.1), (3.2) and (3.3). $\mathcal{B}$ responds $\mathcal{A}$ with $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$.
2. Permission Queries. For an access permission query on (ID, ID', C_2), $\mathcal{B}$ checks whether $R(ID) = R(ID') = 1$.
   (a) If the equation holds, $\mathcal{B}$ computes an access permission using the techniques in (3.4), (3.5), (3.6), (3.7) and (3.8). $\mathcal{B}$ responds $\mathcal{A}$ with $(D_1, D_2, D_3, K_{ID,2})$.
   (b) Otherwise, $\mathcal{B}$ aborts the simulation and outputs his guess μ' randomly.
3. Re-encryption Queries. For a re-encryption query on (ID, ID', CT) where $CT = (C_1, C_2, C_3, C_4, \sigma, vk)$, $\mathcal{B}$ check whether he has created an access permission for (ID, ID', C_2). If he has not created an access permission, he create an access permission as above to obtain $(D_1, D_2, D_3, K_{ID,2})$ and computes $C'_1 = D_2 \cdot C_1, C'_2 = C_2, C'_3 = C_3, C'_4 = C_4, C'_5 = D_1, C'_6 = D_3, C'_7 = D_3, \sigma' = \sigma, vk' = vk$. $\mathcal{B}$ responds $\mathcal{A}$ with $CT' = (C'_1, C'_2, C'_3, C'_4, C'_5, C'_6, C'_7, \sigma', vk')$.
4. Owner Decryption Queries. For an owner decryption query on (ID, CT) where $CT = (C_1, C_2, C_3, C_4, \sigma, vk)$ is a ciphertext for the identity ID, $\mathcal{B}$ check $R(ID) \stackrel{?}{=} 0$.
   (a) If $R(ID) = 0$, $\mathcal{B}$ aborts and outputs his guess μ' randomly.
   (b) If $R(ID) \neq 0$, $\mathcal{B}$ check the signature $\sigma \stackrel{?}{=} \mathsf{Verify}(vk, C_2, C_3, C_4)$. If the equation holds, $\mathcal{B}$ generates a secret key K_{ID} for ID as (3.1), (3.2) and (3.3), and responds $\mathcal{A}$ with $C_1 \cdot \frac{e(K_{ID,2}, C_3)}{e(K_{ID,1}, C_2)}$.
5. Requester Decryption Queries. For a requester decryption query on (ID, ID', CT), $\mathcal{B}$ checks whether he has created secret keys for ID and ID', an access permission for (ID, ID', C_2) and a re-encryption ciphertext CT'. If he has not done these, he creates secret keys, an access permission and a re-encryption as in the secret key query, permission query and re-encryption query to obtain $(SK_{ID}, SK_{ID'})$, $(D_1, D_2, D_3, K_{ID,2})$ and CT'.

Then, $\mathcal{B}$ computes K_1 as above and responds with $C_1' \cdot \frac{e(C_7', C_3)}{e(K_1, C_2')}$.

Challenge. $\mathcal{A}$ submits an identity ID^* and two messages M_0 and M_1 with the equal length. $\mathcal{B}$ checks $R(ID) \stackrel{?}{=} 0$.

1. If $R(ID) = 1$, $\mathcal{B}$ aborts and outputs his guess μ' randomly.
2. If $R(ID) = 0$, $\mathcal{B}$ flips an unbiased coin with $\{0, 1\}$ and obtains $\omega \in \{0, 1\}$. The challenger runs $\mathsf{SKeyGen}(1^\ell) \to (sk^*, vk^*)$ and computes $C_1^* = M_\omega \cdot Z$, $C_2^* = C = g^c$, $C_3^* = C^{Q(ID^*)} = (u_0 \prod_{i \in \mathbf{I}^*} u_i)^c$, $C_4^* = C^{\mathcal{H}(vk^*)+\theta}$ and $\sigma^* = \mathsf{Sign}(sk^*, C_2^*, C_3^*, C_4^*)$. $\mathcal{B}$ sends the ciphertext $CT^* = (C_1^*, C_2^*, C_3^*, C_4^*, \sigma^*, vk^*)$ to $\mathcal{A}$.

Phase 2. Phase 1 is repeated with the following restricts.

1. Secret Key Queries. $\mathcal{A}$ cannot query a secret key for ID^*.
2. Permission Queries. $\mathcal{A}$ cannot query an access permission on (ID^*, ID, C_2^*) and a secret key for ID.
3. Re-encryption Queries. $\mathcal{A}$ cannot query a re-encryption on (ID^*, ID, C^*), permission on (ID^*, ID, C_2) and secret key for ID.
4. Owner Decryption Queries. $\mathcal{A}$ cannot query the owner decryption on (ID^*, CT^*).
5. Requester Decryption Queries. $\mathcal{A}$ cannot query a re-encryption on (ID^*, ID, CT^*) and requester decryption on $(ID, \widetilde{CT}^*)$, where $\widetilde{CT}^*$ is the re-encrypted ciphertext of CT^*.

Guess. $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs $\mu' = 0$; otherwise $\mathcal{B}$ outputs $\mu' = 1$.

As shown above, the public parameters, public keys and the secret keys created in the simulation paradigm are identical to those created in the real protocol. $\mathcal{B}$ does not abort the simulation if and only if the secret keys can be generated correctly, $R(ID^*) = 0$ and the signatures in the ciphertext are valid. In q_1 secret key queries, q_2 permission queries, q_3 re-encryption queries, q_4 owner decryption queries and q_5 requester decryption queries, $\mathcal{B}$ needs to create at most $q_1 + 2q_2 + 2q_3 + q_4 + 2q_5$ secret keys.

Now, we bound the probability with which $\mathcal{B}$ can break the DBDH assumption. This bound is computed using the method in [BGW05]. If $\mu = 1$, $\mathcal{A}$ cannot obtain anything about ω. Hence, $\mathcal{A}$ can output $\omega' \neq \omega$ with no advantage, namely, $\Pr[\omega' \neq \omega | \mu = 1] = \frac{1}{2}$. Since $\mathcal{B}$ outputs $\mu' = 1$ when $\omega' \neq \omega$, we have $\Pr[\mu' = \mu | \mu = 1] = \frac{1}{2}$. If $\mu = 0$, $\mathcal{A}$ can output $\omega' = \omega$ with the advantage at least $\epsilon(\ell)$, namely $\Pr[\omega' = \omega | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ outputs $\mu' = 0$ when $\omega' = \omega$, we have

$$\begin{aligned} \Pr[\mu' = \mu | \mu = 0] - \frac{1}{2} &\geq Adv_{\mathcal{A}} - \Pr[Abort] \\ &\geq 32(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5) \\ &\quad (n+1)\epsilon''(\ell) + \epsilon'(\ell) - \Pr[Abort] \end{aligned}$$

where $\Pr[Abort]$ is the probability with which $\mathcal{B}$ aborts the simulation. The first inequality is from the case $Z = e(g,g)^{abc}$, so the simulation is performed correctly if $\mathcal{B}$ does not abort. Hence, $\mathcal{B}$ can solve the DBDH assumption with the advantage at least

$$\frac{\epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5)(n+1)} \geq \epsilon''(\ell).$$

It remains to bound the probability with which $\mathcal{B}$ aborts the simulation as a result of $\mathcal{A}$'s decryption queries. We claim that $\Pr[Abort] < \epsilon'(\ell)$. Otherwise, a forged signature can be computed with the advantage at least $\epsilon'(\ell)$. Briefly, receiving the challenged signing key sk^* in the simulation, $\mathcal{A}$ causes an abort by submitting a decryption query which includes a forged signature of one ciphertext under sk^*. Therefore, $\mathcal{B}$ can use the forged signature to break the EU-CMA property of the one-time signature. Notably, $\mathcal{A}$ can only query one signature for the challenged ciphertext. Hence, we have $\Pr[Abort] < \epsilon'(\ell)$.

Therefore, $\mathcal{B}$ can break the DBDH assumption with advantage at least

$$\frac{\epsilon(\ell)}{32(q_1 + 2q_2 + 2q_3 + q_4 + 2q_5)(n+1)}.$$

This finishes our proof. □

We demonstrate the computation cost and communication cost of our IBDDS-I scheme and IBDDS-II scheme in Table 3.1 and Table 3.2, respectively. The comparison of various properties is described in Table 3.3. By T_S, T_V and E_S, we denote the running time of executing one signing algorithm of the one-time signature scheme, the running time of executing one verifying algorithm of the one-time signature and the length of the signature generated by the one-time signature, respectively.

3.5 Chapter Summary

Distributed data storage schemes provide users with a convenience to outsource their files to untrusted proxy servers. Identity-based distributed data storage (IBDDS) schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates. In this chapter, we proposed two new interactive IBDDS schemes in standard model where, for one query, a requester can only access one file, instead of all files. Furthermore, the access permission can be determined by the owner without the help of the PKG. Notably, our schemes are secure against the collusion attacks. The first scheme is IND-CPA secure, while the second one is IND-CCA2 secure.

Table 3.1: The Computation cost of Our IBDDS-I and IBDDS-II Schemes

Scheme	Computation Cost							
	Setup	KeyGen	Encryption	Query	Permission	Re-encryption	$\mathcal{O}$ Decryption	$\mathcal{R}$ Decryption
IBDDS-I	$3T_e$	$3T_e + 4T_p$	$3T_e + 2T_p$	$2T_e$	$3T_e + 5T_p$	0	$2T_p$	$T_e + 2T_p$
IBDDS-II	$3T_e$	$3T_e + 4T_p$	$5T_e + 4T_p + T_S + T_V + T_H$	$2T_e$	$3T_e + 5T_p$	0	$2T_p + T_V$	$T_e + 2T_p + T_V$

Table 3.2: The Communication cost of Our IBDDS-I and IBDDS-II Schemes

Scheme	Communication Cost						
	KeyGen	Encryption	Query		Permission	$\mathcal{O}$ Decryption	$\mathcal{R}$ Decryption
	$PKG \rightarrow \mathcal{U}$	$\mathcal{O} \rightarrow \mathcal{PS}$	$\mathcal{R} \rightarrow \mathcal{PS}$	$\mathcal{PS} \rightarrow \mathcal{O}$	$\mathcal{O} \rightarrow \mathcal{PS}$	$\mathcal{PS} \rightarrow \mathcal{O}$	$\mathcal{PS} \rightarrow \mathcal{R}$
IBDDS-I	$3E_{\mathbb{G}}$	$2E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$3E_{\mathbb{G}}$	$4E_{\mathbb{G}}$	$3E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$2E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$5E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$
IBDDS-II	$3E_{\mathbb{G}}$	$3E_{\mathbb{G}} + E_{\mathbb{G}_\tau} + E_S$	$3E_{\mathbb{G}}$	$4E_{\mathbb{G}}$	$3E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$4E_{\mathbb{G}} + E_{\mathbb{G}_\tau} + E_S$	$6E_{\mathbb{G}} + E_{\mathbb{G}_\tau} + E_S$

Table 3.3: Property Comparison of Related Schemes

Property	[Mat07]	[WWMO10b]	[WWMO10a]	[GA07]	[CT07]	[THJ08]	Our Schemes
Unidirectional	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Noninteractive	No	No	No	Yes	Yes	Yes	Yes
Key optimal	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Collusion-safe	Yes	Yes	Yes	No	No	No	Yes
Nontransitive	Yes	Yes	Yes	Yes	Yes	Yes	Yes
File-based acess	No	No	No	No	No	No	Yes

Chapter 4

Identity-based Data Storage in Cloud Computing

In this chapter, we propose an identity-based data storage (IBDS) scheme where both intra-domain and inter-domain queries are supported. Hence, our scheme is suitable to cloud computing. Parts of this work appeared in [HSM13a].

4.1 Introduction

Cloud computing is a distributed system where users from different domains can share their files with others. In a data storage scheme, to protect the confidentiality of his files, a user encrypts them prior to outsourcing them to an external proxy server. Then, if other users want to access a file of the owner, he must request an access permission from the owner. If he is legal, the owner sends an access permission to the proxy server. Obtaining the access permission for the owner, the proxy server can transfer a ciphertext for the owner to a ciphertext for the requester. Finally, the requester can decrypt the ciphertext and access the file.

Identity-based cryptosystem can provide the advantage that two parties can communicate directly without the necessity to verify the public-key certificate. Although IBDS schemes have been proposed, it is not trivial to construct an IBDS scheme in cloud computing as the secret keys of the users from different domains are generated by different PKGs. Therefore, schemes proposed in Chapter 3 are not suitable to cloud computing as the owner and the requester must come from the same domain. How to guarantee that the files outsourced by a user in a domain can be accessed by other users in different domains is a challenging research problem.

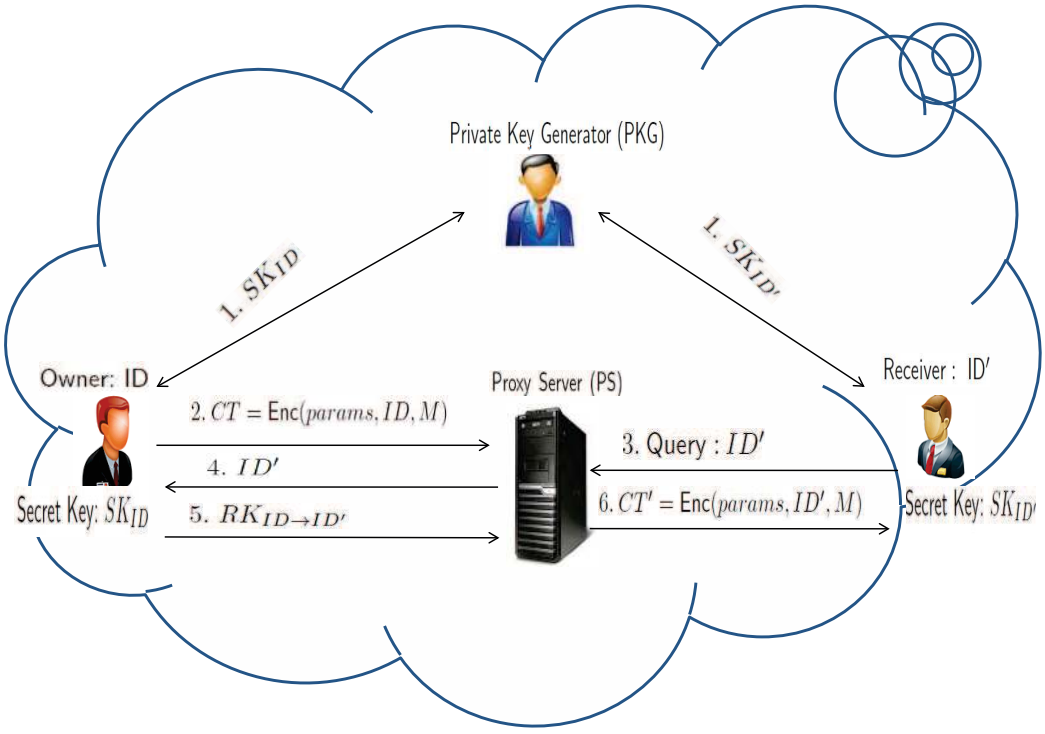

Figure 4.1: Identity-based Data Storage Supporting Intra-Domain Query

4.1.1 Related Work

The background about data storage systems and identity-based proxy re-encryption (IBPRE) schemes have been introduced in Section 3.1.1.

All the previous IBPRE schemes [GA07, CT07, Mat07, WWMO10b, WWMO10a] only addressed the *intra-domain setting*, namely both the original decryptor and the designated decryptor should come from the *same* domain. Tang, Hartel and Jonker [THJ08] first introduced an inter-domain IBPRE scheme where the *inter-domain setting* is considered, namely the proxy server can transfer a ciphertext for the original decryptor in a domain to a ciphertext for a designated decryptor in another domain. Although, this scheme is not secure against the collusion attacks, they made an important step from intra-domain IBPRE to inter-domain IBPRE. We review this scheme in Section 4.3.

To clarify the *intra-domain setting* and *inter-domain setting*, we depict IBDS schemes which support intra-domain query and inter-domain query in Figure 4.1 and Figure 4.2, respectively.

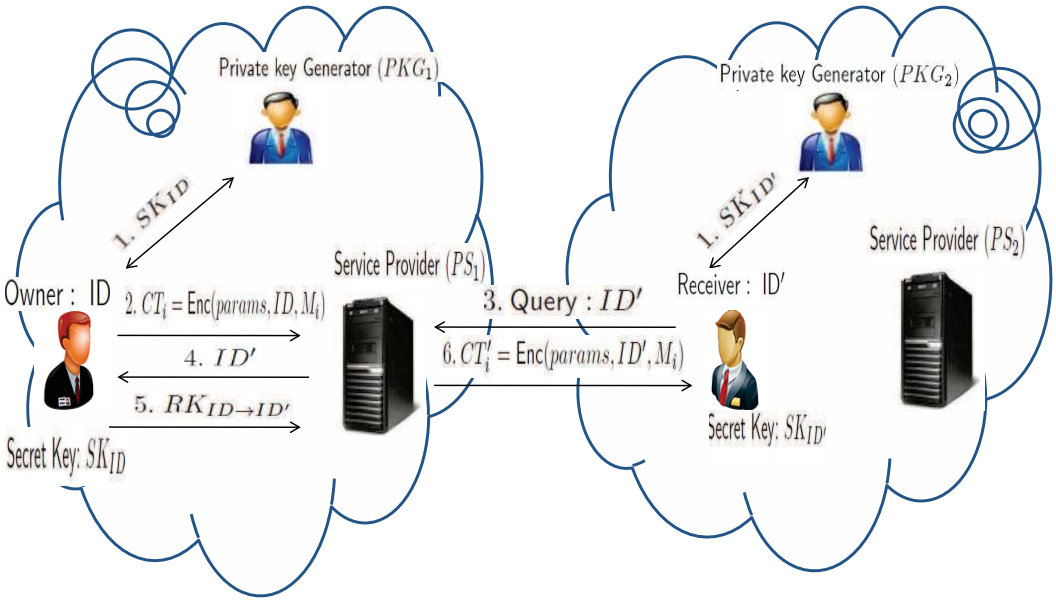

Figure 4.2: Identity-based Data Storage Supporting Inter-Domain Query

4.1.2 Our Contribution

Cloud computing is a distributed system where multiple domains can co-exist. It is desirable that users in different domains can share sensitive data with others. Hence, a sound IBDS scheme in cloud computing should support not only intra-domain queries but also inter-domain queries. However, current IBPRE schemes cannot be exploited in the cloud computing scenario as they cannot support inter-domain queries and resist collusion attacks. We propose an IBDS scheme where both intra-domain and inter-domain queries are supported. In this scheme, an access permission can be made by the owner independently without the help of the PKG. For one query, the requester can only access one file of the owner. Our scheme is secure against the collusion attacks and selective-identity secure in the standard model.

4.1.3 Chapter Organization

In Section 4.2, we introduce the formal definition and security model of IBDS in cloud computing. We review the scheme proposed by Tang, Hartel and Jonker in Section 4.3. In Section 4.4, we propose an IBDS scheme supporting intra-domain and inter-domain queries and prove its security. Section 4.5 summarizes this chapter.

4.2 Formal Definition and Security Model

In this section, we introduce the formal definition and security model of IBDS in cloud computing.

4.2.1 Formal Definition

There are four entities in an IBDS scheme: the private key generator (PKG), the data owner $\mathcal{O}$, the proxy server $\mathcal{PS}$ and the requester $\mathcal{R}$. An IBDS scheme supporting intra-domain and inter-domain queries consists of the following seven algorithms:

$\mathsf{Setup}(1^\ell) \rightarrow (params, (MSK_1, PK_1), (MSK_2, PK_2))$. The setup algorithm takes as input 1^ℓ and outputs the public parameters $params$, master secret-public key pairs (MSK_1, PK_1) and (MSK_2, PK_2) for PKG_1 in domain $\mathfrak{D}_1$ and PKG_2 in domain $\mathfrak{D}_2$, respectively.

$\mathsf{KeyGen}(params, ID, MSK_i) \rightarrow SK_{ID}$. The key generation algorithm takes as input the public parameters $params$, an identity ID in the domain $\mathfrak{D}_i$ and the master secret key MSK_i, and outputs a secret key SK_{ID} for the identity ID, where $i \in \{1, 2\}$.

$\mathsf{Enc}(params, PK_i, ID, M) \rightarrow CT$. The encryption algorithm takes as input the public parameters $params$, the public key PK_i, $\mathcal{O}$'s identity ID and a message M, and outputs a ciphertext $CT = (C_1, C_2)$. It sends CT to the proxy server $\mathcal{PS}_i$ in the domain $\mathfrak{D}_i$, where $i = \{1, 2\}$.

$\mathsf{Query}(ID', SK_{ID'}, CT) \rightarrow AI$. The query algorithm takes as input $\mathcal{R}$'s identity ID', his secret key $SK_{ID'}$ and the ciphertext CT, and outputs an authentication information AI. It sends AI to the $\mathcal{PS}_i$.

1. If both $\mathcal{O}$ and $\mathcal{R}$ are in the same domain, $\mathcal{PS}_i$ sends (ID', AI, C_2) to $\mathcal{O}$.
2. If the $\mathcal{O}$ and $\mathcal{R}$ are in different domains. Suppose that $\mathcal{O}$ is in the domain $\mathfrak{D}_i$ and $\mathcal{R}$ is in the domain $\mathfrak{D}_{3-i}$, where $i = \{1, 2\}$. $\mathcal{PS}_i$ sends (ID', PK_{3-i}, AI, C_2) to $\mathcal{O}$.

$\mathsf{Perm}(params, SK_{ID}, ID', C_2) \rightarrow RK_{ID \rightarrow ID'}$. The permission algorithm verifies the the authentication information AI. If it is valid, this algorithm takes as input the public parameters $params$, $\mathcal{O}$'s secret key SK_{ID}, $\mathcal{R}$'s identity ID' and the

partial ciphertext C_2, and outputs an access permission $RK_{ID \to ID'}$. It sends $RK_{ID \to ID'}$ to the $\mathcal{PS}_i$.

$\mathsf{Re\text{-}enc}(params, RK_{ID \to ID'}, ID', CT) \to CT'$. The re-encryption algorithm takes as inputs the public parameters $params$, the access permission $RK_{ID \to ID'}$, $\mathcal{R}$'s identity ID' and the ciphertext CT, and outputs a re-encrypted ciphertext $CT' = \mathsf{Enc}(params, ID', M)$.

Dec. There are two decryption algorithms. One is for $\mathcal{O}$ and the other is for $\mathcal{R}$.

1. $\mathsf{Dec}_{\mathcal{O}}(params, SK_{ID}, CT) \to M$. The owner decryption algorithm takes as input the public parameters $params$, $\mathcal{O}$'s secret key SK_{ID} and the ciphertext CT, and outputs the message M.
2. $\mathsf{Dec}_{\mathcal{R}}(params, SK_{ID'}, CT') \to M$. The requester decryption algorithm takes as input the public parameters $params$, $\mathcal{R}$'s secret key $SK_{ID'}$ and the re-encrypted ciphertext CT', and outputs the message M.

Definition 4.1 *We say an identity-based data storage scheme supporting intra-domain and inter-domain queries is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}_{\mathcal{O}}(params, SK_{ID}, & \mathsf{Setup}(1^{\ell}) \to (params, (MSK_1, PK_1), \\ CT) \to M & (SK_2, PK_2)); \\ & \mathsf{KeyGen}(params, ID, MSK_i) \to SK_{ID}; \\ & \mathsf{Enc}(params, PK_i, ID, M) \to CT \end{array}\right] = 1$$

and

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}_{\mathcal{R}}(params, SK_{ID'}, & \mathsf{Setup}(1^{\ell}) \to (params, (MSK_1, PK_1), \\ & (MSK_2, PK_2)); \\ & \mathsf{KeyGen}(params, ID, MSK_i) \to SK_{ID}; \\ CT') \to M & \mathsf{KeyGen}(params, ID', MSK_{3-i}) \to SK_{ID'}; \\ & \mathsf{Enc}(params, PK_i, ID, M) \to CT; \\ & \mathsf{Perm}(params, SK_{ID}, ID', C_2); \\ & \to RK_{ID \to ID'}; \\ & \mathsf{Re\text{-}enc}(params, RK_{ID \to ID'}, ID', CT) \\ & \to CT' \end{array}\right] = 1$$

where $i \in \{1, 2\}$ and the probability is taken over the random coins which all the algorithms in the scheme consumes.

4.2.2 Security Model

The following game is used to formalize the security model of IBDS schemes supporting intra-domain and inter-domain queries. This model is derived from the selective-identity secure IBE scheme [BB04a] and defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Initialization. $\mathcal{A}$ submits an identity ID^* with which he wants to be challenged to $\mathcal{C}$. Suppose that ID^* is in the domain $\mathfrak{D}_i$ where $i \in \{1, 2\}$.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and the secret-public key pairs (MSK_1, PK_1) for the PKG_1 in $\mathfrak{D}_1$ and (MSK_2, PK_2) for the PKG_2 in $\mathfrak{D}_2$, respectively. It sends $(params, PK_1, PK_2)$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can make the following queries adaptively:

1. Secret Key Queries. For a secret key query on an identity ID in $\mathfrak{D}_i$ or $\mathfrak{D}_{3-i}$ where $i \in \{1, 2\}$ and the only restrict is $ID \neq ID^*$, $\mathcal{C}$ runs KeyGen $(params, ID, MSK_i)$ to generate a secret key SK_{ID} for ID. $\mathcal{C}$ responds $\mathcal{A}$ with SK_{ID}. This query can be made multiple times.
2. Permission Queries. For an access permission on (ID, ID', CT) where the restricts are $ID \neq ID^*$ and $ID' \neq ID^*$, $\mathcal{C}$ runs $\mathsf{KeyGen}(params, ID,$ $MSK_i)$ to generate a secret key SK_{ID}, then runs $\mathsf{Permission}$ $(params,$ $SK_{ID}, ID', C_2)$ to obtain $RK_{ID \to ID'}$. $\mathcal{C}$ responds $\mathcal{A}$ with $RK_{ID \to ID'}$. This query can be made multiple times.

Challenge. $\mathcal{A}$ submits two messages M_1 and M_2 with equal length. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. Then, $\mathcal{C}$ computes $CT^* = \mathsf{Enc}(params, PK_i, ID^*, M_b)$ and responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can make queries as in Phase 1 with the following constrains.

1. Secret Key Queries. $\mathcal{A}$ cannot query $\mathsf{KeyGen}(parmas, ID^*, MSK_i)$.
2. Permission Queries. $\mathcal{A}$ cannot query $\mathsf{Perm}(ID, ID', CT)$ where $ID = ID^*$ or $ID' = ID^*$.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 4.2 *We say that an identity-based data storage scheme supporting intra-domain and inter-domain queries is $(T, q_1, q_2, \epsilon(\ell))$- secure against selective-identity and adaptively chosen plaintext attacks (or IND-sID-CPA) if no PPT adversary $\mathcal{A}$ making at most q_1 secret key queries and q_2 permission queries can win the game with the advantage*

$$Adv_{\mathcal{A}-IBD}^{IND-sID-CPA} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

4.3 Tang, Hartel and Jonker's Scheme

In this section, we first describe the scheme proposed by Tang, Hartel and Jonker [THJ08], then point out that this scheme cannot resist collusion attacks.

Tang, Hartel and Jonker's scheme [THJ08] works as follows:

Setup_1. The setup algorithm takes as input 1^ℓ, and outputs a bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ and two hash functions $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}$ and $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{G}$. Let g be a generator of $\mathbb{G}$. PKG_1 generates his master secret key $\alpha_1 \overset{R}{\leftarrow} \mathbb{Z}_p$ and public key $y_1 = g^{\alpha_1}$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, y_1, \mathcal{H}_1, \mathcal{H}_2)$ and the master secret key is α_1.

KeyGen_1. The key generation algorithm takes as input the master secret key α_1 and an identity ID, and outputs $SK_{ID} = \mathcal{H}_1(ID)^{\alpha_1}$. The secret key for the identity ID is SK_{ID}.

$\mathsf{Encryption}_1$. To encrypt a message M, the encryption algorithm selects $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$C_0 = M \cdot e(y_1, \mathcal{H}_1(ID))^s \text{ and } C_1 = g^s.$$

The ciphertext is $CT = (C_0, C_1)$.

$\mathsf{Decryption}_1$. To decrypt a ciphertext, the decryption algorithm takes as input the secret key of the original decryptor SK_{ID} and the ciphertext CT, and computes

$$M = \frac{C_0}{e(C_1, SK_{ID})}.$$

PKG_2 generates a master secret-pubic key pair $\mathcal{KG}(1^\ell) \rightarrow (\alpha_2, y_2)$ with $y_2 = g^{\alpha_2}$, and sets up another IBE scheme with $(\mathsf{Setup_2}, \mathsf{KeyGen_2}, \mathsf{Encryption_2}, \mathsf{Decryption_2})$. Suppose that the designated decryptor with identity ID' has registered with PKG_2 and obtained his secret key $SK_{ID'}$.

RKeyGen. To generate a re-encryption key for the ID', the re-encryption key generation algorithm selects $X \stackrel{R}{\leftarrow} \{0,1\}^*$, and computes

$$R_1 = \mathcal{H}_1(ID)^{-\alpha_1} \cdot \mathcal{H}_2(X) \text{ and } R_2 = \mathsf{Encryption_2}(X, ID').$$

The re-encryption key is $RK_{ID \rightarrow ID'} = (R_1, R_2)$.

Re-encryption. To re-encrypt the ciphertext CT, the re-encryption algorithm takes as input the re-encryption key $RK_{ID \rightarrow ID'}$, the identity ID' and the ciphertext CT, and computes

$$C'_1 = \mathsf{Encryption_2}(\mathsf{ID'}, \mathsf{C_1}),\ C'_2 = R_2 \ \text{ and } \ C'_3 = C_0 \cdot e(C_1, R_1 \cdot \mathcal{H}_2(C_1)).$$

The re-encrypted ciphertext is $CT' = (C'_1, C'_2, C'_3)$.

$\mathsf{Decryption_3}$. To decrypt a re-encrypted ciphertext CT', the requester decryption algorithm takes as input the secret key $SK_{ID'}$ and the re-encrypted ciphertext CT', and compute $W_1 = \mathsf{Decryption_2}(\mathsf{SK_{ID'}}, \mathsf{C'_1}),\ W_2 = \mathsf{Decryption_2}(\mathsf{SK_{ID'}}, \mathsf{C'_2})$ and

$$M = \frac{C'_3}{e(W_1, \mathcal{H}_2(W_2) \cdot \mathcal{H}_2(W_1))}.$$

Notably, the security model in [THJ08] is different from our security model. Because, we addressed that, for one query, the requester can only access one file even if he can compromise the proxy server. However, in [THJ08], for one query, the requester can access all the ower's files if he can compromise the proxy server.

Collusion Attacks. If the designated decryptor can compromise the proxy server, he can obtain the re-encryption key $RK_{id \rightarrow ID'} = (R_1, R_2)$. Then, he can use his secret key $SK_{ID'}$ to compute $X = \mathsf{Decryption_2}(\mathsf{SK_{ID'}}, \mathsf{R_2})$. Therefore, he can compute the secret key of the original decryptor by computing $SK_{ID} = \frac{\mathcal{H}_2(X)}{R_1}$.

4.4 Identity-Based Data Storage Scheme in Cloud Computing

In this section, we propose an identity-based data storage scheme which supports intra-domain and inter-domain queries and prove its security. In our scheme, the access permission can be determined by the data owner independently without the need of the PKG. Especially, the access permission is bound to not only the requester's identity but also the requested ciphertext. Furthermore, our scheme is secure against the collusion attacks.

Overview. Suppose that there are two domains: $\mathfrak{D}_1$ and $\mathfrak{D}_2$. At first, the private key generator PKG_i in the domain $\mathfrak{D}_i$ generates his master secret-public pair $(\xi_i, (g_i, h_i))$ where $i \in \{1, 2\}$. Then, PKG_i authenticates users in the domain $\mathfrak{D}_i$ and issues secret keys to them. Prior to outsourcing his files, the data owner $\mathcal{O}$ encrypts them under his identity ID. Then, $\mathcal{O}$ sends the ciphertexts to the proxy server $\mathcal{PS}$. $\mathcal{PS}$ validates the ciphertexts. If they are computed correctly, $\mathcal{PS}$ stores them for $\mathcal{O}$; otherwise, he rejects the ciphertexts. Suppose that $\mathcal{PS}$ can detect which domain the requester $\mathcal{R}$ is from and $\mathcal{O}$ can know which file $\mathcal{R}$ wants to access by the partial ciphertext. If $\mathcal{R}$ wants to access a file stored in $\mathcal{PS}$, he computes an authentication information $(Q, F, K_{ID',3})$ using his secret key $SK_{ID'}$ and sends it to the $\mathcal{PS}$. If $\mathcal{R}$ and $\mathcal{O}$ are in the same domain, the $\mathcal{PS}$ sends $(ID', Q, F, K_{ID',3}, C_2)$ to $\mathcal{O}$, where C_2 is the partial ciphertext. If $\mathcal{R}$ and $\mathcal{O}$ are in different domains, the $\mathcal{PS}$ sends $(ID', Q, F, K_{ID',3}, (g_i, h_i), C_2)$ to $\mathcal{O}$. $\mathcal{O}$ validates $\mathcal{R}$ by verifying $(Q, F, K_{ID',3})$. If the authentication is successful, $\mathcal{O}$ creates an access permission $(P_1, P_2, P_3, K_{ID,2})$ and sends it to the $\mathcal{PS}$. $\mathcal{PS}$ re-encrypts the ciphertext CT and sends the re-encrypted ciphertext CT' to $\mathcal{R}$. At the end, $\mathcal{R}$ decrypts the re-encrypted ciphertext CT' using his secret key $SK_{ID'}$.

In the inter-domain query, suppose that $\mathcal{O}$ is in the domain $\mathfrak{D}_i$ and $\mathcal{R}$ is in the domain $\mathfrak{D}_{3-i}$, where $i \in \{1, 2\}$. In deed, in our scheme, $\mathcal{O}$ uses his secret key to generate an access key[1] for the $\mathcal{R}$. Furthermore, the $\mathcal{PS}_i$ can use the access key to transfer a ciphertext for $\mathcal{O}$ to a ciphertext for $\mathcal{R}$.

Our scheme is based on the IBE scheme [BBH06]. The protocol is described in Figure 4.3.

[1]This key maybe not identical to that generated by the PKG_i for the requester with identity ID'. Here, we just mean that the requester can use it to decrypt the re-encrypted ciphertext.

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \to (e, p, \mathbb{G}, \mathbb{G}_\tau)$ where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\tau$ and p is a prime number. Let $g, h, \eta, \mathfrak{g}, \mathfrak{h}$ be the generators of $\mathbb{G}$.

1. PKG_1 chooses $\alpha_1 \overset{R}{\leftarrow} \mathbb{Z}_p$ and sets $g_1 = g^{\alpha_1}$, $h_1 = \mathfrak{g}^{\alpha_1}$ and $\xi_1 = \eta^{\alpha_1}$. The master secret key is ξ_1 and the public key is $(g, h, \eta, \mathfrak{g}, \mathfrak{h}, g_1, h_1)$.
2. PKG_2 chooses $\alpha_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and sets $g_2 = g^{\alpha_2}$, $h_2 = \mathfrak{g}^{\alpha_2}$ and $\xi_2 = \eta^{\alpha_2}$. The master secret key is ξ_2 and the public key is $(g, h, \eta, \mathfrak{g}, \mathfrak{h}, g_2, h_2)$.

KeyGen. This algorithm takes as input the master secret key ξ_i of PKG_i and an identity $ID \in \mathbb{Z}_p$ in the domain $\mathfrak{D}_i$, and computes

$$K_{ID,1} = \eta^{\alpha_i}(g_i^{ID}h)^{r_{ID}},\ K_{ID,2} = g^{r_{ID}} \text{ and } K_{ID,3} = \mathfrak{g}^{r_{ID}}$$

where $r_{ID} \overset{R}{\leftarrow} \mathbb{Z}_p$ and $i \in \{1, 2\}$. The secret key for a user $\mathcal{U}$ with identity ID is $SK_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$. This secret key can be verified by

$$e(K_{ID,1}, g) \overset{?}{=} e(\eta, g_i) \cdot e(g_i^{ID}h, K_{ID,2}) \text{ and } e(K_{ID,2}, \mathfrak{g}) \overset{?}{=} e(g, K_{ID,3}).$$

Encryption. To encrypt a message M, the owner $\mathcal{O}$ with identity ID chooses $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$C_1 = M \cdot e(g_i, \eta)^s,\ C_2 = g^s \text{ and } C_3 = (g_i^{ID}h)^s$$

where $i \in \{1, 2\}$. The ciphertext for the message M is $CT = (C_1, C_2, C_3)$. $\mathcal{O}$ sends CT to the proxy server $\mathcal{PS}_i$ in the domain $\mathfrak{D}_i$. $\mathcal{PS}_i$ validates the ciphertext by verifying

$$e((g_i^{ID}, h), C_2) \overset{?}{=} e(C_3, g)$$

where $i \in \{1, 2\}$. If the equation holds, $\mathcal{PS}_i$ stores the ciphertext $CT = (C_1, C_2, C_3)$ for $\mathcal{O}$; otherwise, he rejects the ciphertext.

Query. If a requester $\mathcal{R}$ with identity ID' wants to access a ciphertext CT, he chooses $k \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $Q = K_{ID',1}\mathfrak{h}^k$ and $F = \mathfrak{g}^k$. He sends $(Q, F, K_{ID',3})$ to the $\mathcal{PS}_i$ who stores CT. There are two scenarios:

1. Both $\mathcal{O}$ and $\mathcal{R}$ are in the same domain $\mathfrak{D}_i$. $\mathcal{PS}_i$ sends $(ID', Q, F, K_{ID',3}, C_2)$ to $\mathcal{O}$.
2. $\mathcal{O}$ and $\mathcal{R}$ are in different domains. Suppose that $\mathcal{O}$ is in $\mathfrak{D}_i$ and $\mathcal{R}$ is in $\mathfrak{D}_{3-i}$ where $i \in \{1, 2\}$. $\mathcal{PS}_i$ sends $(ID', Q, F, K_{ID',3}, (g_{3-i}, h_{3-i}), C_2)$ to $\mathcal{O}$.

Permission. There are two scenarios:

1. Both $\mathcal{O}$ and $\mathcal{R}$ are in the same domain $\mathfrak{D}_i$. $\mathcal{O}$ checks

$$e(Q, \mathfrak{g}) \stackrel{?}{=} e(\eta, h_i) \cdot e(g_i^{ID'} h, K_{ID',3}) \cdot e(\mathfrak{h}, F).$$

2. $\mathcal{O}$ and $\mathcal{R}$ are in different domains. $\mathcal{O}$ checks

$$e(Q, \mathfrak{g}) \stackrel{?}{=} e(\eta, h_{3-i}) \cdot e(g_{3-i}^{ID'} h, K_{ID',3}) \cdot e(\mathfrak{h}, F).$$

If one of the two equations holds, $\mathcal{O}$ chooses $\beta, \nu \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes

$$P_1 = \frac{K_{ID,1}}{Q \cdot F^{\nu}} \cdot g^{ID'\beta},\ P_2 = \mathfrak{g}^{\nu} \text{ and } P_3 = e(C_2, g)^{ID'\beta}.$$

Then, $\mathcal{O}$ sends the access permission $RK_{ID \to ID'} = (P_1, P_2, P_3, K_{ID,2})$ to the $\mathcal{PS}_i$.

Re-encryption. Receiving $RK_{ID \to ID'} = (P_1, P_2, P_3, K_{ID,2})$ from $\mathcal{O}$, $\mathcal{PS}_i$ re-encrypts the ciphertext as

$$C_1' = P_3 \cdot C_1,\ C_2' = C_2,\ C_3' = C_3,\ C_4' = P_1,\ C_5' = P_2 \text{ and } C_6' = K_{ID,2}.$$

$\mathcal{PS}_i$ responds $\mathcal{R}$ with $CT' = (C_1', C_2', C_3', C_4', C_5', C_6')$.

Decryption.

1. To decrypt a cipherest $CT = (C_1, C_2, C_3)$, $\mathcal{O}$ computes

$$M = C_1 \cdot \frac{e(K_{ID,2}, C_3)}{e(K_{ID,1}, C_2)}.$$

2. To decrypt a re-encrypted ciphertext $CT' = (C_1', C_2', C_3', C_4', C_5', C_6')$, $\mathcal{R}$ computes

$$E = K_{ID',1} \cdot C_4' \cdot \mathfrak{h}^k \cdot C_5'^k$$

and

$$M = C_1' \cdot \frac{e(C_6', C_3')}{e(E, C_2')}.$$

Figure 4.3: Identity-based Data Storage Scheme Supporting Intra-Domain and Inter-Domain Queries

Correctness. The following equations hold.

$$\begin{aligned} C_1 \cdot \frac{e(K_{ID,2}, C_3)}{e(K_{ID,1}, C_2)} &= M \cdot e(g_i, \eta)^s \frac{e(g^{r_{ID}}, (g_i^{ID} h)^s)}{e(\eta^{\alpha_i} (g_i^{ID} h)^{r_{ID}}, g^s)} \\ &= M \cdot e(g_i, \eta)^s \frac{e(g^{r_{ID}}, (g_i^{ID} h)^s)}{e(g_i, \eta)^s \cdot e(g^{r_{ID}}, (g_i^{ID} h)^s)} \\ &= M \cdot e(g_i, \eta)^s \cdot \frac{1}{e(g_i, \eta)^s} \\ &= M. \end{aligned}$$

$$\begin{aligned} P_1 &= \frac{K_{ID,1}}{Q \cdot F^{\nu}} \cdot g^{ID'\beta} \\ &= \frac{\eta^{\alpha_i} (g_i^{ID} h)^{r_{ID}}}{K_{ID',1} \mathfrak{h}^k \cdot \mathfrak{g}^{k\nu}} \cdot g^{ID'\beta}. \end{aligned}$$

$$P_3 = e(C_2, g)^{ID'\beta} = e(g, g)^{s\beta ID'}.$$

$$C_1' = P_3 \cdot C_1 = M \cdot e(g_i, \eta)^s \cdot e(g, g)^{s\beta ID'}.$$

$$\begin{aligned} E &= K_{ID',1} \cdot C_4' \cdot \mathfrak{h}^k \cdot C_5'^k \\ &= K_{ID',1} \cdot \frac{K_{ID,1}}{K_{ID',1} \mathfrak{h}^k \cdot \mathfrak{g}^{k\nu}} \cdot g^{ID'\beta} \cdot \mathfrak{h}^k \cdot \mathfrak{g}^{k\nu} \\ &= K_{ID,1} \cdot g^{ID'\beta} \\ &= \eta^{\alpha_i} \cdot (g_i^{ID} h)^{r_{ID}} \cdot g^{ID'\beta}. \end{aligned}$$

Therefore, we have

$$\begin{aligned} & C_1' \cdot \frac{e(C_6', C_3')}{e(E, C_2)'} \\ = \; & C_1' \cdot \frac{e(g^{r_{ID}}, (g_i^{ID} h)^s)}{e(\eta^{\alpha_i} (g_i^{ID} h)^{r_{ID}} g^{ID'\beta}, g^s)} \\ = \; & C_1' \cdot \frac{e(g, g_i^{ID} h)^{s r_{ID}}}{e(g_i, \eta)^s \cdot e(g, g_i^{ID} h)^{s r_{ID}} \cdot e(g, g)^{s\beta ID'}} \\ = \; & C_1' \cdot \frac{1}{e(g_i, \eta)^s \cdot e(g, g)^{s\beta ID'}} \\ = \; & M \cdot e(g_i, \eta)^s \cdot e(g, g)^{s\beta ID'} \cdot \frac{1}{e(g_i, \eta)^s \cdot e(g, g)^{s\beta ID'}} \\ = \; & M. \end{aligned}$$

Theorem 4.1 *Our identity-based data storage scheme supporting intra-domain and inter-domain queries is $(T, q_1, q_2, \epsilon(\ell))$-secure against selective identity and adaptively chosen plaintext (or IND-sID-CPA) if the $(T', \epsilon(\ell)')$ decisional bilinear Diffie-Hellman assumption holds in the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ where*

$$T' = T + \Theta(T) \quad \textit{and} \quad \epsilon(\ell)' = \frac{1}{2}\epsilon(\ell).$$

Proof: Suppose that there exists a PPT adversary $\mathcal{A}$ who can $(T, q_1, q_2, \epsilon(\ell))$ break the security of our scheme, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the DBDH assumption as follows. The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g be a generator of $\mathbb{G}$. $\mathcal{C}$ flips an unbiased coin μ with $\{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, b^c, e(g, g)^{abc})$ to $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \overset{R}{\leftarrow} \mathbb{Z}_p$. $\mathcal{B}$ will outputs his guess μ' on μ.

Initialization. $\mathcal{A}$ submits an identity ID^* with which he wants to be challenged to $\mathcal{B}$. Let ID^* be in the domain $\mathfrak{D}_i$ where $i \in \{1, 2\}$.

Setup. $\mathcal{B}$ selects $\upsilon, \gamma, \theta \overset{R}{\leftarrow} \mathbb{Z}_p$ and sets $g_i = A$, $g_{3-i} = g^\upsilon$, $\eta = B$, $\mathfrak{g} = g^\theta, h_i = A^\theta, h_{3-i} = g^{\upsilon\theta}$ and $h = g_i^{-ID^*} g^\gamma$. It chooses $\mathfrak{h} \overset{R}{\leftarrow} \mathbb{G}$. The public parameters are $(g, h, \eta, \mathfrak{g}, \mathfrak{h})$. The public keys for the PKG_i in $\mathfrak{D}_i$ and PKG_{3-i} in $\mathfrak{D}_{3-i}$ are (g_i, h_i) and (g_{3-i}, h_{3-i}), respectively. $\mathcal{B}$ sends $\{(e, p, \mathbb{G}, \mathbb{G}_\tau), g, h, \eta, \mathfrak{g}, \mathfrak{h}, g_i, h_i, g_{3-i}, h_{3-i}\}$ to $\mathcal{A}$. The master secret keys for PKG_i and PKG_{3-i} are g^{ab} and $g^{\upsilon b}$, respectively.

Phase 1.

1. Secret Key Queries. For a secret key query on an identity ID where the only restrict is $ID \neq ID^*$, $\mathcal{B}$ works as follows.

   (a) If ID is in $\mathfrak{D}_{3-i}$, $\mathcal{B}$ chooses $r \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

   $$K_{ID,1} = B^\upsilon (g_{3-i}^{ID} h)^r, \; K_{ID,2} = g^r \text{ and } K_{ID,3} = K_{ID,2}^\theta;$$

   (b) If ID is in $\mathfrak{D}_i$, $\mathcal{B}$ chooses $r \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

   $$K_{ID,1} = B^{\frac{-\gamma}{ID - ID^*}} (g_i^{ID} h)^r, \; K_{ID,2} = g^r B^{\frac{-1}{ID - ID^*}} \text{ and } K_{ID,3} = K_{ID,2}^\theta.$$

   $\mathcal{B}$ responds $\mathcal{A}$ with $K_{ID} = (K_{ID,1}, K_{ID,2}, K_{ID,3})$.

We claim that the secret key is computed correctly. We have

$$\begin{aligned} K_{ID,1} &= B^{\frac{-\gamma}{ID-ID^*}}(g_i^{ID}h)^r \\ &= g^{\frac{-b\gamma}{ID-ID^*}}(g^{a(ID-ID^*)+\gamma})^r \\ &= g^{ab}g^{-ab}g^{\frac{-b\gamma}{ID-ID^*}}(g^{a(ID-ID^*)+\gamma})^r \\ &= g^{ab}(g^{a(ID-ID^*)+\gamma})^{\frac{-b}{ID-ID^*}}(g^{a(ID-ID^*)+\gamma})^r \\ &= g^{ab}(g^{a(ID-ID^*)+\gamma})^{r-\frac{b}{ID-ID^*}} \\ &= g^{ab}(g_i^{ID}h)^{r-\frac{b}{ID-ID^*}}. \end{aligned}$$

Let $\hat{r} = r - \frac{b}{ID-ID^*}$, we have $K_{ID,1} = g^{ab}(g_i^{ID}h)^{\hat{r}}$, $K_{ID,2} = g^r B^{\frac{-1}{ID-ID^*}} = g^{r-\frac{b}{ID-ID^*}} = g^{\hat{r}}$ and $K_{ID,3} = K_{ID,2}^{\theta} = g^{\hat{r}\theta} = \mathfrak{g}^{\hat{r}}$. Hence, the distribution of $(K_{ID,1}, K_{ID,2}, K_{ID,3})$ is identical to those generated in the real protocol.

2. Permission Queries. For a permission query on (ID, ID', C_2) where the only restricts are $ID \neq ID^*$ and $ID' \neq ID^*$, $\mathcal{B}$ works as follows.

   (a) If ID' is in $\mathfrak{D}_{3-i}$, $\mathcal{B}$ chooses $r' \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

   $$K_{ID',1} = B^{\upsilon}(g_{3-i}^{ID}h)^{r'}.$$

   (b) If ID' is in $\mathfrak{D}_i$, $\mathcal{B}$ chooses $r' \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes

   $$K_{ID',1} = B^{\frac{-\gamma}{ID'-ID^*}}(g_i^{ID'}h)^{r'}.$$

   $\mathcal{B}$ chooses $t, k, \beta, \nu \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $Q = K_{ID',1}\mathfrak{h}^t$, $F = \mathfrak{g}^k$,

   $$P_1 = \frac{K_{ID,1}}{Q \cdot F^{\nu}} \cdot g^{ID'\beta},\ P_2 = \mathfrak{g}^{\nu} \text{ and } P_3 = e(C_2, g)^{ID'\beta}.$$

   $\mathcal{B}$ responds $\mathcal{A}$ with $RK_{ID\rightarrow ID'} = (P_1, P_2, P_3, K_{ID,2})$.

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with the equal length. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$ and obtains one bit $\omega \in \{0,1\}$. $\mathcal{B}$ computes

$$C_1^* = M_{\omega} \cdot Z,\ C_2^* = C \text{ and } C_3^* = C^{\gamma}.$$

$\mathcal{B}$ responds $\mathcal{A}$ with the ciphertext $CT^* = (C_1^*, C_2^*, C_3^*)$.

Phase 2. Phase 1 is repeated.

Guess. The adversary $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs $\mu' = 0$; otherwise $\mathcal{B}$ outputs $\mu' = 1$.

As shown above, the public parameters, public keys and secret keys generated in the simulation paradigm are identical to those generated in the real protocol. Therefore, we can compute the advantage with which $\mathcal{B}$ can break the DBDH assumption as follows.

If $\mu = 0$, $CT^* = (C_1^*, C_2^*, C_3^*)$ is a legal ciphertext of the message M_ω. Hence, $\mathcal{A}$ can output $\omega' = \omega$ with advantage at least $\epsilon(\ell)$, namely $\Pr[\omega' = \omega | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ outputs $\mu' = 0$ when $\omega' = \omega$, we have $\Pr[\mu' = \mu | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$.

In the case $\mu = 1$, $CT^* = (C_1^*, C_2^*, C_3^*)$ is not a legal ciphertext of the message M_ω. Hence, $\mathcal{A}$ can output $\omega' \neq \omega$ without any advantage, namely $\Pr[\omega' \neq \omega | \mu' = 1] = \frac{1}{2}$. Since $\mathcal{B}$ outputs $\mu' = 1$ when $\omega' \neq \omega$, we have $\Pr[\mu' = \mu | \mu = 1] = \frac{1}{2}$.

Therefore, the advantage with which $\mathcal{B}$ can break the DBDH assumption is $\left|\frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2}\right| \geq \frac{1}{2} \times (\frac{1}{2} + \epsilon(\ell)) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \geq \frac{1}{2}\epsilon(\ell)$. □

Collusion Attacks. It is difficult to give a formal definition of collusion attacks. Therefore, we only provide a heuristic proof that our scheme is collusion resistant. In our scheme, when computing an access permission, $\mathcal{O}$ chooses a random number $\beta \overset{R}{\leftarrow} \mathbb{Z}_p$, randomizes his secret key $K_{ID,1}$ by g^β and computes $P_2 = \mathfrak{g}^\nu$ and $P_3 = e(C_2, g)^{ID'\beta}$. If $\mathcal{R}$ can compromise $\mathcal{PS}_i$, they can obtain $V = K_{ID,1} \cdot g^{ID'\beta} = P_1 \cdot Q \cdot P_2^k$. If he can compute $K_{ID,1}$ from V, he can compute $\Psi = g^\beta = (\frac{V}{K_{ID,1}})^{\frac{1}{ID'}}$. However, this is intractable since the random number β is unknown to the adversary $\mathcal{A}$.

We demonstrate the computation cost and the communication cost of our scheme and [THJ08] in Table 4.1 and Table 4.2, respectively. By T_H, we denote the time of running one hash function.

4.5 Chapter Summary

Cloud computing is a distributed system where users in different domains can share data with others. Identity-based data storage (IBDS) schemes have been proposed to outsource sensitive data from the owner to an external proxy server. Nevertheless, there are some drawbacks in the existing schemes in the literature. For example, they can only support the intra-domain query and the access key is computed with the help of the PKG. Additionally, the proxy server must be trusted. In this chapter, we proposed a new IBDS scheme which is suitable to the cloud computing scenario

as it supports both intra-domain and inter-domain queries. In our scheme, an access key is bound to not only the requester's identity but also the requested ciphertext. Notably, an access key can be computed by the owner independently without any help of the PKG. For one query, the requester can only access one file of the owner, instead of all files. Furthermore, our scheme is secure against the collusion attacks. We proved the IND-CAP security of the proposed scheme in the selective-identity model.

Table 4.1: The Computation Cost of Our IBDS scheme

Scheme	Setup	KeyGen	Encryption	Query	Permission	Re-Encryption	Decryption	
							O	R
[THJ08] scheme	$2T_e$	$T_e + T_H$	$2T_e + T_p + T_H$	0	$3T_e + T_p + 3T_H$	$2T_e + 2T_p + 2T_H$	T_p	$3T_p + 2T_H$
Our scheme	$6T_e$	$5(T_e + T_p)$	$4T_e + 3T_p$	$2T_e$	$5(T_e + T_p)$	0	$2T_p$	$2(T_e + T_p)$

Table 4.2: The Communication Cost of Our IBDS scheme

Scheme	Setup	KeyGen	Encryption	Query			Permission	Re-encryption
					Intra	Inter		
		$PKG \to \mathcal{U}$	$\mathcal{O} \to \mathcal{PS}$	$\mathcal{R} \to \mathcal{PS}$	$\mathcal{PS} \to \mathcal{O}$	$\mathcal{PS} \to \mathcal{O}$	$\mathcal{O} \to \mathcal{PS}$	$\mathcal{PS} \to \mathcal{R}$
[THJ08] scheme	$4E_{\mathbb{G}}$	$E_{\mathbb{G}}$	$E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	0	0	0	$2E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$2E_{\mathbb{G}} + 3E_{\mathbb{G}_\tau}$
Our Scheme	$9E_{\mathbb{G}}$	$3E_{\mathbb{G}}$	$2E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$3E_{\mathbb{G}}$	$3E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$5E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$3E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$	$5E_{\mathbb{G}} + E_{\mathbb{G}_\tau}$

Part II

Personal Information Protection

Chapter 5

Privacy-Preserving Decentralized Key-Policy Attribute-based Encryption

In this chapter, we propose a privacy-preserving decentralized key-policy attribute-based encryption scheme. Parts of this work appeared in [HSMY12c].

5.1 Introduction

In traditional access control schemes [NT94, Sma03], a central authority can determine whether a user can access the sensitive data. We observed the following drawbacks in these schemes, especially in distributed systems. Firstly, in a large distributed system, it is a difficult task for the authority to manage numerous users identities. Secondly, the central authority must be fully trusted. In the scenario that the authority is malicious, he can impersonate any user without being detected. Comparatively, in attribute-based access control schemes [SW05, BSW07], users are identified by their descriptive attributes, instead of their unique identities. Furthermore, a user can share his data with others by specifying an access structure so that all the users whose attributes satisfy it can access the data without knowing their identities. Therefore, attribute-based access control schemes are sound primitives to share data with multiple users without knowing their exact identities. In order to reduce the trust on the central authority, some distributed access control schemes are proposed [Cha07, MKE08, LCLS08, LW11, LHC+11]. Although, distributed attribute-based access control schemes demonstrated lots of metrics, they seldom consider the privacy of users. Especially, a user's attributes could be exposed to the malicious authorities. Thereafter, to provide a sound solution for sharing sensitive data with multiple users, a distributed attribute-based access control with privacy preserving scheme should be addressed.

In an open communication environment, such as the Internet, sensitive data must

be encrypted prior to being transmitted. To achieve this, encryption schemes can be employed to protect the confidentiality of the sensitive data. Nevertheless, traditional encryption schemes cannot express flexible access policies, and additionally, the sender must know all the identities (public keys) of the receivers.

5.1.1 Attribute-based Encryption

Introduced by Sahai and Waters [SW05], attribute-based encryption (ABE) is a more efficient public-key encryption scheme where complex access structures can be implemented. In an ABE scheme, both a user's secret key and a ciphertext are labeled with a set of attributes. An encryptor can encrypt a message under a set of attributes. Prior to decrypting the ciphertext, the receiver must obtain the secret (attribute) keys from a trusted party called central authority (CA). The receiver can decrypt the ciphertext and obtain the data if and only if there is a match between his secret keys and the attributes listed in the ciphertext. The original idea of ABE is to construct a fuzzy (error-tolerant) identity-based encryption (IBE) scheme [Sha84, BF01].

Since its seminal introduction, ABE as an efficient primitive has attracted lots of attention in the public-key cryptography research community. Essentially, ABE schemes can be classified into the following two kinds:

- Key-Policy ABE (KP-ABE): In these schemes, a user's secret key is associated with an access structure, while a ciphertext is labeled with a set of attributes [SW05, GPSW06, OSW07, Cha07, CC09].
- Ciphertext-Policy ABE (CP-ABE): In these schemes, a user's secret key is labeled with a set of attributes, while a ciphertext is associated with an access structure [BSW07, CN07, HLR10, LOS+10, Wat11].

An *access structure* is embedded in a distributed system to control users from accessing the protected resource. Given a universal set $\mathbb{P}$, we say that an access structure is *monotonic* if a subset $S' \subseteq \mathbb{P}$ satisfies the access structure, then all subsets $S \subseteq \mathbb{P}$ which contain S' satisfy the access structure. A (k, n)-*threshold access structure* is an access structure where, given a universal set $\mathbb{P}$ with $|\mathbb{P}| = n$, a subset $S \subseteq \mathbb{P}$ satisfies the access structure if and only if $|S| \geq k$. In an ABE scheme, an access structure is selected by the authority (in KP-ABE) or the encryptor (in CP-ABE) to control who can decrypt a ciphertext. For example, in an KP-ABE

scheme, the authority specifies a (k, n)-threshold access structure and issues secret keys to users according to this access structure. An encryptor encrypts a message under k-out-of-n attributes and lists them in the ciphertext. If a user holds a set of attributes which contains those listed in the ciphertext, he can use his secret key to decrypt the ciphertext and obtain the message. However, if a user does not hold the required attributes specified in the ciphertext, he cannot obtain anything about the plaintext.

The limitation of the original ABE scheme is that it can only express a threshold access structure. Goyal, Pandey, Sahai and Waters [GPSW06] proposed an ABE scheme for fine-grained access policy where any monotonic access structure can be expressed by the *access tree* technique. In an access tree, there is a tree access structure where interior nodes consist of AND and OR gates and the leaf nodes consist of the attributes. Each interior node ω of the tree specifies a threshold gate (k_ω, n_ω), where n_ω is the number of the children of ω and $0 < k_\omega \leq n_\omega$. Thereafter, when $k_\omega = n_\omega$, the gate is an AND gate. When $k_\omega = 1$, the gate is an OR gate. If a set of attributes satisfies the tree access structure, the corresponding secret keys can be used to reconstruct the secret embedded in the vertex of the tree. Subsequently, Ostrovsky, Sahai and Waters [OSW07] proposed an ABE scheme with a *non-monotonic access structure* where a secret key is labeled with a set of attributes including not only the positive but also the negative attributes. Comparatively, an ABE scheme with a non-monotonic access structure can express more complicated access structures.

The first CP-ABE scheme was proposed by Bethencourt, Sahai and Waters [BSW07], and was proven to be secure in the generic group model. In contrast with a KP-ABE scheme, the access structure in a CP-ABE scheme is determined by the encryptor, instead of the CA. Therefore, the encryptor can decide who can decrypt the ciphertext; while, this is decided by the CA in a KP-ABE scheme. Cheung and Newport [CN07] proposed another CP-ABE scheme and reduced the difficulty of breaking their scheme to the DBDH assumption. Both these CP-ABE schemes can only express a threshold access structure. Waters [Wat11] proposed a more generic CP-ABE scheme where any access structure can be expressed by using the *linear secret sharing scheme* (LSSS) technique [Bei96].

Attrapadung and Imai proposed a dual-policy ABE scheme [AI09] which combines a KP-ABE scheme with a CP-ABE scheme. In this scheme, two access structures are exploited. One is for the objective attributes labeled in the ciphertext,

and the other is for the subjective attributes held by the users. However, there is only one access structure in both a KP-ABE scheme and a CP-ABE scheme.

Rial and Preneel [RP10] proposed a blind key extract protocol for the centralized ABE scheme [BSW07]. Hence, this scheme is a blind centralized ABE scheme.

An ABE scheme should be secure against the *collusion attacks* [SW05], namely no group of users can combine their secret keys to decrypt a ciphertext which none of them can decrypt by himself. The most common technique used to prevent the collusion attacks is randomization. The central authority randomizes a user's secret key by selecting a random number [OSW07, CN07] or a random polynomial [SW05, Cha07, CC09].

ABE has been used as a building block to express flexible access structures in practical systems, such as distributed systems [YRL11], data outsourcing systems [HN11] and cloud computing [YWRL10].

5.1.2 Multiple-Authority Attribute-based Encryption

In their seminal work [SW05], Sahai and Waters left an open question that whether it is possible to construct an ABE scheme where a user's secret key can come from multiple authorities. Chase [Cha07] answered this question affirmatively by proposing a multi-authority KP-ABE scheme. In this scheme, there are multiple authorities, one of those is called CA. The CA knows all the secret keys of the other authorities. A user needs to obtain secret keys from all these authorities. Being different from one authority ABE schemes, it is hard to resist collusion attacks in a multi-authority ABE scheme. Especially, if the multiple authorities can work independently, the scheme is subject to this attack. Chase [Cha07] overcame this problem by introducing a global identifier (GID) to a multi-authority ABE scheme. All authorities tie a user's secret keys to his GID. In order to let the ciphertext be independent of the user's GID, the CA must compute a special secret key for the user using his secret key and the other authorities' secret keys. Although this scheme is *not* a decentralized ABE scheme, Chase made an important step from one authority ABE to multi-authority ABE.

Lin, Cao, Liang and Shao [LCLS08] proposed a multi-authority ABE scheme without a central authority based on the distributed key generation (DKG) protocol [GlJK$^+$99] and the joint zero secret sharing (JZSS) protocol [GJKR01]. At the system setup phase, the multiple authorities must collaboratively execute the DKG

protocol and the JZSS protocol twice and k times, respectively, where k is the degree of the polynomial selected by each authority. Each authority must maintain $k + 2$ secret keys. This scheme is k-resilient, namely the scheme is secure if and only if the number of the colluding users is no more than k which is determined at the system setup phase.

M*ü*ller, Katzenbeisser and Eckert [MKE08] proposed a distributed CP-ABE scheme, where the pairing operations executed by the decryption algorithm are constant. This scheme was proven to be secure in the generic group [BSW07], instead of reducing to a complexity assumption. Furthermore, there must be a central authority to generate the global key and issues secret keys to users.

Chase and Chow proposed another multi-authority KP-ABE scheme [CC09] which improved the previous scheme [Cha07] and removed the need of the CA. Notably, they also addressed the privacy issue. In the previous multi-authority ABE schemes [Cha07, LCLS08], a user must submit his GID to each authority to obtain the corresponding secret keys. This will risk the user being traced by a group of corrupted authorities. Chase and Chow provided an anonymous key distribution protocol for the GID, where the 2-party secure computation technique is employed. As a result, a group of authorities cannot cooperate to collect a user's attributes by tracing his GID. However, the multiple authorities must interact to setup the system. Each pair of authorities must execute a 2-party key exchange protocol to share the seeds of the selected pseudorandom functions (PRF) [NPR99]. This scheme is $(N - 2)$-tolerant, namely the scheme is secure if and only if the number of the compromised authorities is no more than $N - 2$, where N is the number of the authorities in the system. The security of this scheme was reduced to DBDH assumption and non-standard complexity assumption (q-decisional Diffie-Hellman inverse (q-DDHI)). Chase and Chow also left an open challenging research problem that how to construct a privacy-preserving multi-authority ABE scheme without the need of cooperations among the authorities.

Lekwo and Waters [LW11]proposed a new multi-authority ABE scheme called decentralizing CP-ABE. This scheme improved the previous multi-authority ABE schemes that require collaborations among multiple authorities to setup the system. In this scheme, no cooperation between the multiple authorities is required in the setup phase and the key generation phase, and there is no central authority. Note that an authority in this scheme can join or leave the system freely without the necessity to re-initialize the system. The scheme was designed in the composite

order ($N = p_1p_2p_3$) bilinear group, and achieves full (adaptive) security in the random oracle model. They also pointed out two methods to create a prime order group variant of their scheme. Unfortunately this scheme is not efficient [Wat11]. Furthermore, a user's attributes can be collected by tracing his GID.

Liu, Cao, Huang, Wong and Yuen [LCH+11] proposed a fully secure multi-authority CP-ABE scheme in the standard model. Their scheme was based on the CP-ABE scheme [LOS+10]. In their scheme, multiple central authorities and attribute authorities co-exist. The central authorities distribute identity-related keys to users and the attribute authorities issue attribute-related keys to users. Prior to obtaining attribute keys from the attribute authorities, a user must obtain secret keys from the multiple central authorities. This multi-authority ABE scheme was designed in the composite order ($N = p_1p_2p_3$) bilinear group.

Li *et al.* [LHC+11] proposed a multi-authority cipher-policy ABE scheme with accountability, where the anonymous key issuing protocol [CC09] was exploited. In their scheme, a user can *only* obtain secret keys anonymously from $N-1$ authorities; while he can be traced when he shares his secret keys with others. Unfortunately, the multiple authorities must initialize the system interactively. Their scheme relied on DBDH assumption, decisional linear (DLIN) assumption and q-DDHI assumption.

5.1.3 Our Contribution

We answered the question left by Chase and Chow [CC09] affirmatively by designing a decentralized KP-ABE scheme with a privacy-preserving key extraction protocol. In our scheme, multiple authorities can perform independently without any cooperation and a central authority. A user's GID is used to tie all his secret keys together, while no group of corrupted authorities can pool the user's attributes by tracing it. Our scheme is $(N-1)$-tolerant for the authorities, where N is the number of the authorities in the system. Our scheme is based on standard complexity assumption (DBDH), instead of any non-standard complexity assumptions (e.g., q-DDHI). To the best of our knowledge, it is *the first* decentralized KP-ABE scheme with privacy-preserving that is based on merely a standard assumption.

5.1.4 Chapter Organization

We review the formal definitions and security models for privacy-preserving decentralized KP-ABE in Section 5.2. In Section 5.3, we propose a privacy-preserving decentralized ABE scheme and prove its security. Finally, Section 5.4 summarizes this chapter.

5.2 Formal Definitions and Security Models

In this section, we introduce the formal definitions and security models of decentralized KP-ABE and privacy-preserving key extraction.

5.2.1 Decentralized Key-Policy Attribute-Based Encryption

A decentralized KP-ABE scheme is defined as follows:

Global-Setup$(1^\ell) \rightarrow params$. The global setup algorithm takes as input 1^ℓ and outputs the public parameters $params$.

Authority-Setup$(1^\ell) \rightarrow (SK_i, PK_i, \mathbb{A}_i)$. The authority setup algorithm takes as input 1^ℓ, and outputs a secret-public key pair $\mathcal{KG}(1)^\ell \rightarrow (SK_i, PK_i)$ and an access structure $\mathbb{A}_i$ for the authority A_i, where $i = 1, 2, \cdots, N$.

KeyGen$(params, GID, \mathbf{A}^i_{GID}, SK_i) \rightarrow SK^i_{\mathcal{U}}$. The key generation algorithm takes as input the public parameters $params$, a global identifier GID, a set of attributes $\mathbf{A}^i_{GID}$ and the authority's secret key SK_i, and outputs a secret key $SK^i_{\mathcal{U}}$ for a user $\mathcal{U}$ with identifier GID, where $\mathbf{A}^i_{GID} = \mathbf{A}_{GID} \bigcap \tilde{\mathbf{A}}_i$, $\mathbf{A}_{GID}$ and $\tilde{\mathbf{A}}_i$ denote the attributes corresponding to the GID and monitored by A_i, respectively.

Enc$(params, \mathbf{A}_C, M) \rightarrow CT$. The encryption algorithm takes as input the public parameters $params$, a set of attributes $\mathbf{A}_C$ and a message M, and outputs a ciphertext CT, where $\mathbf{A}_C = \{\mathbf{A}^1_C, \mathbf{A}^2_C, \cdots, \mathbf{A}^N_C\}$ and $\mathbf{A}^i_C = \mathbf{A}_C \bigcap \tilde{\mathbf{A}}_i$.

Dec$(params, GID, \{SK^i_{\mathcal{U}}\}_{i \in \mathbf{I}_C}, CT)$. The decryption algorithm takes as input the public parameters $params$, the global identifier GID, the secret keys $\{SK^i_{\mathcal{U}}\}_{i \in \mathbf{I}_C}$ and the ciphertext CT, and outputs the message M, where $\mathbf{I}_C$ is the index set of the authorities A_i such that $\mathbf{A}^i_C \neq \{\phi\}$.

Definition 5.1 *We say that a decentralized key-policy attribute-based encryption scheme is correct if*

$$\Pr\left[\begin{array}{l|l} & \mathsf{Global-Setup}(1^\ell) \rightarrow params; \\ \mathsf{Dec}(params, GID, & \mathsf{Authority-Setup}(1^\ell) \rightarrow (SK_i, PK_i, \mathbb{A}_i); \\ \{SK^i_{\mathcal{U}}\}_{i\in\mathbf{I}_C}, CT) & \mathsf{KeyGen}(params, GID, \mathbf{A}^i_{GID}, SK_i) \rightarrow SK^i_{\mathcal{U}}; \\ = M & \mathsf{Enc}(params, \mathbf{A}_C, M) \rightarrow CT; \\ & \{\mathbf{A}_{GID} \bigcap \tilde{\mathbf{A}}_i \in \mathbb{A}_i\}_{i\in\mathbf{I}_C}, \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by all algorithms in the scheme.

5.2.2 Security Model for Decentralized Key-Policy Attribute-based Encryption

As far as the security of ABE schemes are concerned, there are two security models: selective-set model [GPSW06] and full security model [BSW07]. In the selective-set model, an adversary must submit a set of attributes which he wants to be challenged with prior to obtaining the public parameters. This limitation is not required in the full security model. All previous ABE schemes were proven in the selective-set model, except [BSW07] and [LOS+10]. Bethencourt, Sahai and Waters [BSW07] proposed the full security model, and proved their scheme in the generic group model. Lewko *et al.* [LOS+10] first proposed an ABE scheme which is fully secure and can be reduced to the subgroup decision assumptions in composite order bilinear groups. They proved their scheme using the dual system encryption technology [Wat09]. Before the proof, two additional algorithms are constructed, namely semi-functional key algorithm and semi-functional ciphertext algorithm.

Our security model on the decentralized KP-ABE schemes is similar to that proposed in [Cha07, CC09], which is known as the selective-set model. This model is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{A}_C$ which he wants to be challenged with and a list of corrupted authorities $\mathbf{C}_{\mathcal{A}}$, where $|\mathbf{C}_{\mathcal{A}}| < N$. There should exist at least one authority $\mathfrak{A}$ such that $\mathbf{A}_C \bigcap \tilde{\mathfrak{A}}$ does not satisfy the access structure specified by $\mathfrak{A}$.

Global-Setup. $\mathcal{C}$ runs the Global-Setup(1^ℓ) algorithm to generate the public parameters $params$, and responds $\mathcal{A}$ with $params$.

Authority-setup.

1. For the authorities $A_i \in \mathbf{C}_\mathcal{A}$, $\mathcal{C}$ sends the secret-public key pair (SK_i, PK_i) and the access structure $\mathbb{A}_i$ to $\mathcal{A}$.
2. For the authorities $A_i \notin \mathbf{C}_\mathcal{A}$, $\mathcal{C}$ sends the public key PK_i and the access structure $\mathbb{A}_i$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_{GID_1}, \mathbf{A}_{GID_2}, \cdots, \mathbf{A}_{GID_{q_1}}$, where the only constraint is $\mathbf{A}_C \nsubseteq \mathbf{A}_{GID_i}$ for $i = 1, 2, \cdots, q_1$. $\mathcal{C}$ responses $\mathcal{A}$ with KeyGen$(params, GID_i, \mathbf{A}^i_{GID}, SK_j)$ for $j = 1, 2, \cdots, N$.

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ computes $CT^* =$ Enc$(params, \mathbf{A}_C, M_b)$ and responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_{GID_{q_1+1}}, \mathbf{A}_{GID_{q_1+2}}, \cdots, \mathbf{A}_{GID_q}$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 5.2 *We say that a decentralized key-policy attribute-based encryption scheme is (T, q, ϵ) secure in the selective-set model if no PPT adversary $\mathcal{A}$ making at most q secret key queries can win the game with the advantage*

$$Adv^{DKP-ABE}_\mathcal{A} = \left| \Pr[b' = b] - \frac{1}{2} \right| > \epsilon(\ell)$$

in the selective-set model.

5.2.3 Privacy-Preserving Decentralized KP-ABE

A privacy-preserving decentralized KP-ABE scheme has the same algorithms Global-Setup, Authority-Setup, Enc and Dec with a decentralized KP-ABE scheme. The only difference is that the algorithm KeyGen in the decentralized KP-ABE scheme is replaced by the algorithm BlindKeyGen. In a privacy-preserving decentralized KP-ABE scheme, the authorities do not know a user's GID nor can cause failures using the information of the GID. This concept is derived from blind IBE schemes [GH07, CKRS09]. The algorithm BlindKeyGen is defined as follows:

$\mathsf{BlindKeyGen}(\mathcal{U}(params, PK_i, GID, decom) \leftrightarrow A_i(params, SK_i, PK_i, \mathbb{A}_i, com)) \rightarrow (SK^i_{\mathcal{U}}, \mathsf{empty})$. In this interactive algorithm, a user $\mathcal{U}$ runs the commitment algorithm $\mathsf{Commit}(params, GID) \rightarrow (com, decom)$ and sends com to the authority A_i. Then, $\mathcal{U}$ and A_i take as input $(params, GID, PK_i, decom)$ and $(params, SK_i, PK_i, \mathbb{A}_i, com)$, respectively. If $\mathsf{Decommit}(params, GID, com, decom) \rightarrow 1$, this algorithm outputs a secret key $SK^i_{\mathcal{U}}$ for $\mathcal{U}$ and empty for A_i, respectively; otherwise it outputs error messages $(\perp, \perp)$ for both $\mathcal{U}$ and A_i.

A sound algorithm BlindKeyGen should satisfy the following two properties: *leak-freeness* and *selective-failure blindness* [GH07, CKRS09]. Leak-freeness requires that, by executing the algorithm BlindKeyGen with a honest authority, a malicious user cannot obtain anything which he cannot obtain by executing the algorithm KeyGen with the honest authoritiy. Selective-failure blindness requires that a malicious authority cannot know anything about the user's GID and cause the algorithm BlindKeyGen to selectively fail depending on the user's GID. These two properties are formally defined by the following two games.

Leak-freeness. This game is defined by a real experiment and an ideal experiment:

Real Experiment: Runs $\mathsf{Setup}(1^\ell) \rightarrow params$ and $\mathsf{Authority\text{-}Setup}(1^\ell) \rightarrow (SK_i, PK_i, \mathbb{A}_i)$. As many times as the distinguisher $\mathcal{D}$ wants, a malicious user $\mathcal{U}^*$ chooses a GID^* and executes BlindKeyGen with an authority A_i: $\mathsf{BlindKeyGen}(\mathcal{U}^*(params, PK_i, GID^*, decom) \leftrightarrow A_i(params, SK_i, PK_i, \mathbb{A}_i, com))$.

Ideal Experiment: Runs $\mathsf{Setup}(1^\ell) \rightarrow params$ and $\mathsf{Authority\text{-}Setup}(1^\ell) \rightarrow (SK_i, PK_i, \mathbb{A}_i)$. As many times as the distinguisher $\mathcal{D}$ wants, the simulator $\hat{\mathcal{U}}^*$ chooses a GID^* and queries a trusted party to obtain the output of the algorithm $\mathsf{KeyGen}(GID, A^i_{GID}, SK_i)$ if $\mathsf{Decommit}(params, GID, com, decom) \rightarrow 1$, and $\perp$ otherwise.

Definition 5.3 *We say that an algorithm* $\mathsf{BlindKeyGen}(\mathcal{U}(\boxplus) \leftrightarrow A_i(\boxdot))$ *associated with a decentralized KP-ABE scheme* $\prod =($Global-Setup, Authority-Setup, KeyGen, Encr, Dec*) is leak-free if for all PPT adversaries* $\mathcal{U}^*$*, there exists an efficient simulator* $\hat{\mathcal{U}}^*$ *such that for the security parameter* ℓ*, no efficient distinguisher* $\mathcal{D}$ *can distinguish whether* $\mathcal{U}^*$ *is executing* Real Experiment *or* Ideal Experiment *with non-negligible advantage, where* $\boxplus = (params, PK_i, GID, decom)$ *and* $\boxdot = (params, SK_i, PK_i, \mathbb{A}_i, com)$.

Selective-failure Blindness. This game is defined as follows:

1. The malicious authority $\mathcal{A}_i^*$ outputs his public key PK_i^* and a pair of global identifiers (GID_0, GID_1).

2. A random bit $b \in \{0, 1\}$ is selected randomly.

3. A_i^* is given the two commitments $com_b = \mathsf{Commit}(params, GID_b)$ and $com_{1-b} = \mathsf{Commit}(params, GID_{1-b})$, and can black-box access the two oracles $\mathcal{U}(params, PK_i, GID_b, com_b)$ and $\mathcal{U}(params, PK_i, GID_{1-b}, com_{1-b})$.

4. The algorithm $\mathcal{U}$ generates secret keys $SK^i_{\mathcal{U},b}$ for GID_b and $SK^i_{\mathcal{U},1-b}$ for GID_{1-b}, respectively.

5. If $SK^i_{\mathcal{U},b} \neq \perp$ and $SK^i_{\mathcal{U},1-b} \neq \perp$, A_i^* is given $(SK^i_{\mathcal{U},b}, SK^i_{\mathcal{U},1-b})$. If $SK^i_{\mathcal{U},b} \neq \perp$ and $SK^i_{\mathcal{U},1-b} = \perp$, A_i^* is given $(\epsilon, \perp)$. If $SK^i_{\mathcal{U},b} = \perp$ and $SK^i_{\mathcal{U},1-b} \neq \perp$, A_i^* is given $(\perp, \epsilon)$. If $SK^i_{\mathcal{U},b} = \perp$ and $SK^i_{\mathcal{U},1-b} = \perp$, A_i^* is given $(\perp, \perp)$.

6. Finally, A_i^* outputs his guess b' on b.

Definition 5.4 *We say that an algorithm* $\mathsf{BlindKeyGen}(\mathcal{U}(\boxplus) \leftrightarrow A_i(\boxdot))$ *associated with a decentralized KP-ABE scheme* $\prod = (\mathsf{Global\ Setup}, \mathsf{Authority\ Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is selective-failure blind if no PPT adversary* A_i^* *can win the game with the advantage*

$$Adv^{SFB}_{A_i^*} = \left| \Pr[b' = b] - \frac{1}{2} \right| > \epsilon(\ell)$$

in the above model, where $\boxplus = (params, PK_i, GID, decom)$ *and* $\boxdot = (\ params, SK_i, PK_i, \mathbb{A}_i, com)$.

Definition 5.5 *We say that a privacy-preserving decentralized KP-ABE scheme* $\tilde{\prod} = (\mathsf{Global\text{-}Setup}, \mathsf{Authority\text{-}Setup}, \mathsf{BlindKeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is secure in the selective-set model if and only if: (1)* $\prod = (\mathsf{Global\text{-}Setup}, \mathsf{Authority\text{-}Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a secure decentralized KP-ABE scheme in the selective-set model; and (2) the algorithm* $\mathsf{BlindKeyGen}$ *is leak-free and selective-failure blind.*

5.3 Privacy-Preserving Decentralized Key-Policy Attribute-based Encryption

In this section, we first propose a decentralized KP-ABE scheme and prove its security. Then, a privacy-preserving key extraction protocol for the proposed decentralized KP-ABE is constructed.

In our privacy-preserving decentralized KP-ABE scheme, a user executes a 2-party secure computation protocol with an authority to possess his secret keys. As a result, the user can obtain his secret key without releasing anything about his identifier to the multiple authorities. As pointed in [CC09], an anonymous credential system [Cha85, CL02] can be used by a user to convince the authorities that he holds the corresponding attributes without revealing his identifier to them. In an anonymous credential system, a user can prove that he has obtained a credential anonymously. Furthermore, the user can interact with different partners using different pseudonyms [LRSW99] such that no partner can link the pseudonyms to the same user. Meanwhile, the user can convince a partner that he has obtained multiple credentials which correspond to the same identifier without releasing his identifier. Hence, this technique can be embedded in our scheme to allow a user to convince the authorities that he hold the corresponding attributes without revealing his identifier to them.

5.3.1 Decentralized Key-Policy Attribute-based Encryption

Our decentralized KP-ABE scheme is described in Figure 5.1. This idea is inspired by the IBE schemes [BB04a, Wat05] and the multi-authority ABE schemes [Cha07, CC09, LW11].

Overview. Suppose that there are N authorities: $A_1, A_2, \cdots, A_N$. A_i manages a set of attributes $\tilde{\mathbf{A}}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,n_i}\}$ and specifies a (k_i, n_i)-threshold access structure $\mathbb{A}_i$ for $i = 1, 2, \cdots, N$. A_i generates a secret-public key pair $((\alpha_i, \beta_i), (Y_i, Z_i))$ and publishes (Y_i, Z_i). For each attribute $a_{i,j} \in \tilde{A}_i$, A_i generates a secret-public key pair $(t_{i,j}, T_{i,j})$ and publishes $T_{i,j}$. The secret keys and public keys of A_i are $(\alpha_i, \beta_i, \{t_{i,j}\}_{a_{i,j} \in \tilde{\mathbf{A}}_i})$ and $(Y_i, Z_i, \{T_{i,j}\}_{a_{i,j} \in \tilde{\mathbf{A}}_i})$, respectively. To issue secret keys to a user $\mathcal{U}$ with a set of attributes $\mathbf{A}_\mathcal{U}$, A_i chooses a random number $r_i \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes D_i by using r_i, his secret keys (α_i, β_i) and $\mathcal{U}$'s identifier u. Hence, $\mathcal{U}$'s identifer u is tied to all his secret keys. D_i is used to protect against the collusion

attacks. Otherwise, two users with identifier u_1 and u_2 can combine their secret keys from A_i and A_j together. A_i chooses a $(k_i - 1)$-degree polynomial $p_i(x)$ with $p_i(0) = r_i$. For each attribute $a_{i,j} \in \mathbf{A}_{\mathcal{U}} \bigcap \tilde{\mathbf{A}}_i$, A_i computes a secret key $D_{i,j}$ by using the value $p_i(a_{i,j})$ and $t_{i,j}$. To encrypt a message $M \in \mathbb{G}_\tau$ under a set of attributes $\mathbf{A}_C = \{\mathbf{A}_C^1, \mathbf{A}_C^2, \cdots, \mathbf{A}_C^N\}$ where $\mathbf{A}_C^i = \mathbf{A}_C \bigcap \tilde{\mathbf{A}}_i$ for $i = 1, 2, \cdots, N$, a random number $s \overset{R}{\leftarrow} \mathbb{Z}_p$ is chosen to hide M in $C_1 = M \cdot \prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i}$. The ciphertext is $CT = (C_1, C_2, C_3, \{C_{i,j}\}_{a_{i,j} \in \mathbf{A}_C})$, where $\{C_{i,j}\}_{a_{i,j} \in \mathbf{A}_C}$ are computed by using s and the public keys $\{T_{i,j}\}_{a_{i,j} \in \mathbf{A}_C}$. If a user holds the attributes listed in the ciphertext, he can use his secret keys D_i and C_2 to compute $E = \prod_{i \in \mathbf{I}_C} e(D_i, C_2) = \prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i} \prod_{i \in \mathbf{I}_C} e(g,h)^{sr_i} \prod_{i \in \mathbf{I}_C} e(g,h_1)^{us\beta_i}$, use $\{D_{i,j}\}_{a_{i,j} \in \mathbf{A}_C}$ and $\{C_{i,j}\}_{a_{i,j} \in \mathbf{A}_C}$ to reconstruct the exponential r_i, and compute $F_i = \prod_{i \in \mathbf{I}_C} e(g,h)^{sr_i}$. Using C_3 and his identifier u, $\mathcal{U}$ can compute $V = \prod_{i \in \mathbf{I}_C} e(g,h_1)^{us\beta_i}$. Consequently, $\mathcal{U}$ can obtain $\prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i}$ by removing $\prod_{i \in \mathbf{I}_C} F_i$ and V from E. Finally, $\mathcal{U}$ can obtain M by removing $\prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i}$ from C_1.

Correctness. We have

$$\begin{aligned} E &= \prod_{i \in \mathbf{I}_C} e(D_i, C_2) \\ &= \prod_{i \in \mathbf{I}_C} e(g^{\alpha_i} h^{r_i} h_1^{u\beta_i}, g^s) \\ &= \prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i} \prod_{i \in \mathbf{I}_C} e(g,h)^{sr_i} \prod_{i \in \mathbf{I}_C} e(g,h_1)^{us\beta_i}, \end{aligned}$$

$$V = e(C_3, h_1^u) = \prod_{i \in \mathbf{I}_C} e(g,h_1)^{us\beta_i},$$

and

$$\begin{aligned} F_i &= \prod_{a_{i,j} \in \mathbf{A}_C^i} e(C_{i,j}, D_{i,j})^{\Delta_{a_{i,j}, \mathbf{A}_C^i}(0)} \\ &= \prod_{a_{i,j} \in \mathbf{A}_C^i} e(g^{st_{i,j}}, h^{\frac{p_i(a_{i,j})}{t_{i,j}}})^{\Delta_{a_{i,j}, \mathbf{A}_C^i}(0)} \\ &= \prod_{a_{i,j} \in \mathbf{A}_C^i} e(g,h)^{sp_i(a_{i,j})\Delta_{a_{i,j}, \mathbf{A}_C^i}(0)} \\ &= e(g,h)^{s\sum_{a_{i,j} \in \mathbf{A}_C^i} p_i(a_{i,j})\Delta_{a_{i,j}, \mathbf{A}_C^i}(0)} \\ &= e(g,h)^{sr_i}. \end{aligned}$$

Global-Setup. This algorithm takes as input 1^ℓ and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let g, h, and h_1 be the generators of $\mathbb{G}$.

Suppose that there are N authorities: $A_1, A_2, \cdots, A_N$. A_i monitors a set of attributes $\tilde{\mathbf{A}}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,n_i}\}$ for $i = 1, 2, \cdots, N$. Let the set of universal attributes $\mathbb{U} = \bigcup_{i=1}^{N} \tilde{\mathbf{A}}_i$.

Authority-Setup. A_i generates a secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (\alpha_i, \beta_i, Y_i, Z_i)$, where $Y_i = e(g,g)^{\alpha_i}$ and $Z_i = g^{\beta_i}$. For each $a_{i,j} \in \tilde{\mathbf{A}}_i$, it chooses $t_{i,j} \xleftarrow{R} \mathbb{Z}_p$, and computes $T_{i,j} = g^{t_{i,j}}$. The public keys and secret keys of A_i are $PK_i = \{Y_i, Z_i, T_{i,1}, T_{i,2}, \cdots, T_{i,n_i}\}$ and $SK_i = \{\alpha_i, \beta_i, t_{i,1}, t_{i,2}, \cdots, t_{i,n_i}\}$, respectively, for $i = 1, 2, \cdots, N$. A_i specifies an (k_i, n_i)-threshold access structure $\mathbb{A}_i$, where $k_i \leq n_i$.

KeyGen. Suppose that a user $\mathcal{U}$ has a global identifier $u \in \mathbb{Z}_p$ and a set of attributes $\mathbf{A}_\mathcal{U}$. To generate a key for an attribute $a_{i,j} \in \tilde{\mathbf{A}}_i \bigcap \mathbf{A}_\mathcal{U}$, A_i selects $r_i \xleftarrow{R} \mathbb{Z}_p$ and a $(k_i - 1)$-degree polynomial $p_i(x) \xleftarrow{R} \mathbb{Z}_p[x]$ with $p_i(0) = r_i$, and computes

$$D_i = g^{\alpha_i} h^{r_i} h_1^{u\beta_i} \text{ and } \{D_{i,j} = h^{\frac{p_i(a_{i,j})}{t_{i,j}}}\}_{a_{i,j} \in \mathbf{A}_\mathcal{U}^i}$$

where $\mathbf{A}_\mathcal{U}^i = \mathbf{A}_\mathcal{U} \bigcap \tilde{\mathbf{A}}_i$, for $i = 1, 2, \cdots, N$. The secret key for $\mathcal{U}$ is $SK_\mathcal{U} = (D_i, \{D_{i,j}\}_{a_{i,j} \in \mathbf{A}_\mathcal{U}^i})$.

Encryption. To encrypt a message $M \in \mathbb{G}_\tau$, this algorithm takes as input a set of attributes $\mathbf{A}_C = \{\mathbf{A}_C^1, \mathbf{A}_C^2, \cdots, \mathbf{A}_C^N\}$ and a random number $s \xleftarrow{R} \mathbb{Z}_p$, and outputs a ciphertext as follows

$$C_1 = M \cdot \prod_{i \in \mathbf{I}_C} e(g,g)^{\alpha_i s},\ C_2 = g^s,\ C_3 = \prod_{i \in \mathbf{I}_C} g^{\beta_i s},\ \{C_{i,j} = T_{i,j}^s\}_{a_{i,j} \in \mathbf{A}_C}$$

where $\mathbf{A}_C^i = \mathbf{A}_C \bigcap \tilde{\mathbf{A}}_i$ and $\mathbf{I}_C$ is the index set of the authorities A_i such that $\mathbf{A}_C^i \neq \{\phi\}$, for $i = 1, 2, \cdots, N$.

Decryption. To decrypt a ciphertext $CT = (C_1,\ C_2,\ C_3,\ \{C_{i,j}\}_{a_{i,j} \in \mathbf{A}_C})$, $\mathcal{U}$ computes

$$E = \prod_{i \in \mathbf{I}_C} e(D_i, C_2),\ V = e(C_3, h_1^u),$$

$$F_i = \prod_{a_{i,j} \in \mathbf{A}_C^i} e(C_{i,j}, D_{i,j})^{\Delta_{a_{i,j}, \mathbf{A}_C^i}(0)} (i \in \mathbf{I}_C)$$

and

$$M = C_1 \cdot \frac{V \cdot \prod_{i \in \mathbf{I}_C} F_i}{E}$$

Figure 5.1: Decentralized Key-Policy Attribute-based Encryption

Therefore,

$$\begin{aligned} C_1 \cdot \frac{V \cdot \prod_{i \in \mathbf{I}_C} F_i}{E} &= M \cdot \frac{\prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i} e(g,h_1)^{us\beta_i} e(g,h)^{sr_i}}{\prod_{i \in \mathbf{I}_C} e(g,g)^{s\alpha_i} e(g,h)^{sr_i} e(g,h_1)^{us\beta_i}} \\ &= M \end{aligned}$$

Theorem 5.1 *Our decentralized key-policy attribute-based encryption scheme is* $(T, q, \epsilon(\ell))$*-secure against chosen plaintext attacks (or IND-CPA) in the selective-set model if the* $(T', \epsilon'(\ell))$ *decisional bilinear Diffie-Hellman assumption holds in* $(e, p, \mathbb{G}, \mathbb{G}_\tau)$*, where*

$$T' = T + \Theta(T) \text{ and } \epsilon'(\ell) = \frac{1}{2}\epsilon(\ell).$$

Proof: Suppose that there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break our decentralized KP-ABE scheme, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the DBDH assumption as follows.

The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g be a generator of $\mathbb{G}$. He flips an unbiased coin μ with $\{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g,g)^{abc})$ to $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g,g)^z)$ to $\mathcal{B}$, where $z \xleftarrow{R} \mathbb{Z}_p$. $\mathcal{B}$ will output his guess μ' on μ.

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{A}_C = \{\mathbf{A}_C^1, \mathbf{A}_C^2, \cdots, \mathbf{A}_C^N\}$ which he wants to be challenged with and a list of corrupted authorities $\mathbf{C}_\mathcal{A}$. Suppose that $\mathbf{A}_C$ is mapped to a user with the global identifier u^*.

Global-Setup. $\mathcal{B}$ selects $\gamma, \eta \xleftarrow{R} \mathbb{Z}_p$, and sets $h = Ag^\gamma$ and $h_1 = g^\eta$. $\mathcal{B}$ sends $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, h_1)$ to $\mathcal{A}$.

Authority-Setup. There should be at least one authority $\mathfrak{A} \notin \mathbf{C}_\mathcal{A}$ where $\mathcal{A}$ can only query secret keys for the attributes in $\mathbf{A}_C$ less than the specified threshold value. Suppose that $\mathfrak{A}$ specifies an (k, n)-threshold access structure and $|A_C \bigcap \tilde{\mathfrak{A}}| = k - 1$, where $\tilde{\mathfrak{A}}$ denotes the set of attributes monitored by $\mathfrak{A}$.

1. For the authority $A_k \in \mathbf{C}_\mathcal{A}$, $\mathcal{B}$ selects $v_k, \beta_k, w_{k,j} \xleftarrow{R} \mathbb{Z}_p$, and sets:

$$Y_k = e(g,g)^{v_k}, Z_k = g^{\beta_k} \text{ and } \{T_{k,j} = g^{w_{k,j}}\}_{a_{k,j} \in \tilde{A}_k}.$$

This implies that the secret key for $A_k \in \mathbf{C}_\mathcal{A}$ is $(v_k, \beta_k, w_{k,j})$. $\mathcal{B}$ responds $\mathcal{A}$ with $(v_k, \beta_k, w_{k,j})$ and $(Y_k, Z_k, T_{k,j})$.

2. For the authority $A_k \notin \mathbf{C}_{\mathcal{A}}$ and $A_k \neq \mathfrak{A}$, $\mathcal{B}$ selects $v_k, \beta_k, w_{k,j} \overset{R}{\leftarrow} \mathbb{Z}_p$, and sets:

$$Y_k = e(g,g)^{bv_k},\ Z_k = g^{\beta_k},\ \{T_{k,j} = g^{w_{k,j}}\}_{a_{k,j} \in \mathbf{A}_C \bigcap \tilde{\mathbf{A}}_i}$$

and

$$\{T_{k,j} = h^{w_{k,j}} = g^{(a+\gamma)w_{k,j}}\}_{a_{k,j} \in \tilde{\mathbf{A}}_i - \mathbf{A}_C}.$$

This implies that the secret key for $A_k \notin \mathbf{C}_{\mathcal{A}}$ is $(bv_k, \beta_k, w_{k,j})$. $\mathcal{B}$ responds $\mathcal{A}$ with $(Y_k, Z_k, T_{k,j})$.

3. For the authority $\mathfrak{A}$, $\mathcal{B}$ selects $w_j, \beta \overset{R}{\leftarrow} \mathbb{Z}_p$, and sets

$$Y = e(g,g)^{ab} \prod_{A_k \notin \mathbf{C}_{\mathcal{A}}} e(g,g)^{-v_k b} \prod_{A_k \in \mathbf{C}_{\mathcal{A}}} e(g,g)^{-v_k},$$

$$Z = g^{\beta},\ \{T_j = g^{w_j}\}_{a_j \in \mathbf{A}_C \bigcap \tilde{\mathfrak{A}}}$$

and

$$\{T_j = h^{w_j} = g^{(a+\gamma)w_j}\}_{a_j \in \tilde{\mathfrak{A}} - \mathbf{A}_C}.$$

This implies that the secret key for $\mathfrak{A}$ is $(v,\ \beta,\ \omega_j)$, where $v = ab - \sum_{A_k \notin \mathbf{C}_{\mathcal{A}}} v_k b - \sum_{A_k \in \mathbf{C}_{\mathcal{A}}} v_k$. $\mathcal{B}$ responds $\mathcal{A}$ with (Y, Z, T_j).

Phase 1. For a secret key query on a global identifier u' with a set of attributes $\mathbf{A}_{\mathcal{U}'}$ where $\mathbf{A}_C \nsubseteq \mathbf{A}_{\mathcal{U}'}$, $\mathcal{B}$ works as follows.

1. For $A_k \in \mathbf{C}_{\mathcal{A}}$, $\mathcal{B}$ can use $(v_k, \beta_k, w_{k,j})$ to compute secret keys for $a_{k,j} \in \tilde{\mathbf{A}}_k \bigcap \mathbf{A}_{\mathcal{U}'}$.
2. For $A_k \notin \mathbf{C}_{\mathcal{A}}$ and $A_k \neq \mathfrak{A}$, $\mathcal{B}$ selects $r_k \overset{R}{\leftarrow} \mathbb{Z}_p$ and a random $(k_k - 1)$-degree polynomial $p_k(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ with $p_k(0) = r_k$. It computes

$$D_k = B^{v_k} h^{r_k} h_1^{u'\beta_k}.$$

(a) If $a_{k,j} \in \mathbf{A}_C \bigcap \tilde{\mathbf{A}}_k$, it computes

$$D_{k,j} = h^{\frac{p_k(a_{k,j})}{w_{k,j}}}.$$

(b) If $a_{k,j} \in \tilde{\mathbf{A}}_k - \mathbf{A}_C$, it computes

$$D_{k,j} = h^{\frac{p_k(a_{k,j})}{(a+\gamma)w_{k,j}}} = g^{\frac{p_k(a_{k,j})}{w_{k,j}}}.$$

3. For $\mathfrak{A}$, it chooses $r, e_1, e_2, \cdots, e_{k-1} \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$D = B^{-\gamma} h^r h_1^{u'\beta} \prod_{A_k \notin \mathbf{C}_\mathcal{A}} B^{-v_k} \prod_{A_k \in \mathbf{C}_\mathcal{A}} g^{-v_k}.$$

(a) If $a_j \in \mathbf{A}_C \bigcap \tilde{\mathfrak{A}}$, it computes

$$D_j = h^{\frac{e_j}{w_j}}.$$

(b) If $a_j \in \tilde{\mathfrak{A}} - \mathbf{A}_C$, it computes

$$D_j = (g^r B^{-1})^{\frac{\Delta_{0,\mathbf{S}}(a_j)}{w_j}} \prod_{i=1}^{k-1} g^{\frac{e_i \Delta_{i,\mathbf{S}}(a_j)}{w_j}}$$

where $\mathbf{S} = (\mathbf{A}_C \bigcap \tilde{\mathfrak{A}}) \bigcup \{0\}$.

We claim that D and D_j are correctly distributed.

$$\begin{aligned}
D &= B^{-\gamma} h^r h_1^{u'\beta} \prod_{A_k \notin \mathbf{C}_\mathcal{A}} B^{-v_k} \prod_{A_k \in \mathbf{C}_\mathcal{A}} g^{-v_k} \\
&= g^{-b\gamma} (g^a g^\gamma)^r h_1^{u'\beta} g^{-(\sum_{A_k \notin \mathbf{C}_\mathcal{A}} bv_k + \sum_{A_k \in \mathbf{C}_\mathcal{A}} v_k)} \\
&= (g^a g^\gamma)^{-b} g^{ab} (g^a g^\gamma)^r h_1^{u'\beta} g^{-(\sum_{A_k \notin \mathbf{C}_\mathcal{A}} bv_k + \sum_{A_k \in \mathbf{C}_\mathcal{A}} v_k)} \\
&= g^{ab} (g^a g^\gamma)^{r-b} h_1^{u'\beta} g^{-(\sum_{A_k \notin \mathbf{C}_\mathcal{A}} bv_k + \sum_{A_k \in \mathbf{C}_\mathcal{A}} v_k)} \\
&= g^{ab - (\sum_{A_k \notin \mathbf{C}_\mathcal{A}} bv_k + \sum_{A_k \in \mathbf{C}_\mathcal{A}} v_k)} h^{r-b} h_1^{u'\beta}.
\end{aligned}$$

Let $r' = r - b$, we have

$$D = g^{ab - (\sum_{A_k \notin \mathbf{C}_\mathcal{A}} bv_k + \sum_{A_k \in \mathbf{C}_\mathcal{A}} v_k)} h^{r'} h_1^{u'\beta}.$$

By selecting $e_1, e_2, \cdots, e_{k-1}$, $\mathcal{B}$ implicitly defines a $(k-1)$-degree polynomial $p(x) \in \mathbb{Z}_p[x]$, such that $p(0) = r'$ and $p(i) = e_i$. So, $\mathcal{B}$ can compute any value of $p(x)$ by interpolation as follows:

$$p(x) = r' \Delta_{0,\mathbf{S}}(x) + \sum_{i=1}^{k-1} e_i \cdot \Delta_{i,\mathbf{S}}(x),$$

where $\mathbf{S} = (\mathbf{A}_C \bigcap \tilde{\mathfrak{A}}) \bigcup \{0\}$.

Hence, for $a_j \in \tilde{\mathfrak{A}} - \mathbf{A}_C$,

$$\begin{aligned} D_j &= h^{\frac{p(a_j)}{(\alpha+\gamma)w_j}} = g^{\frac{p(a_j)}{w_j}} = g^{\frac{r'\Delta_{0,\mathbf{S}}(a_j)}{w_j}} \prod_{i=1}^{k-1} g^{\frac{e_i\cdot\Delta_{i,\mathbf{S}}(a_j)}{w_j}} \\ &= (g^r B^{-1})^{\frac{\Delta_{0,\mathbf{S}}(a_j)}{w_j}} \prod_{i=1}^{k-1} g^{\frac{e_i\cdot\Delta_{i,\mathbf{S}}(a_j)}{w_j}}. \end{aligned}$$

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$, and obtains one bit $\omega \in \{0,1\}$. $\mathcal{B}$ computes

$$C_1 = Z \cdot M_\omega,\ C_2 = C,\ C_3 = \prod_{i\in\mathbf{I}_C} C^{\beta_i},\ C_{i,j} = \{C^{w_{i,j}}\}_{a_{i,j}\in\mathbf{A}_C}.$$

$\mathcal{B}$ responds $\mathcal{A}$ with the challenged ciphertext $CT^* = (C_1, C_2, C_3, \{C_{i,j}\}_{a_{i,j}\in\mathbf{A}_C})$. So, CT^* is a valid ciphertext of M_ω with correct distribution whenever $Z = e(g,g)^{abc}$.

Phase 2. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs $\mu' = 0$; otherwise $\mathcal{B}$ outputs $\mu' = 1$.

As shown above, the public parameters and the secret keys created in the simulation paradigm are identical to those in the real protocol. Now, we compute the advantage with which $\mathcal{B}$ can break the BDDH assumption.

If $\mu = 0$, $(C_1, C_2, C_3, \{C_{i,j}\}_{a_{i,j}\in\mathbf{A}_C})$ is a correct ciphertext of M_ω. Therefore, $\mathcal{A}$ can output $\omega' = \omega$ with the advantage at least $\epsilon(\ell)$, namely $\Pr[\omega' = \omega|\mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ outputs $\mu' = 0$ when $\omega' = \omega$, we have $\Pr[\mu' = \mu|\mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$.

If $\mu = 1$, $\mathcal{A}$ cannot get any information about ω. Hence, $\mathcal{A}$ can output $\omega' \neq \omega$ with no advantage, namely $\Pr[\omega' \neq \omega|\mu = 1] = \frac{1}{2}$. Since $\mathcal{B}$ outputs $\mu' = 1$ when $\omega' \neq \omega$, we have $\Pr[\mu' = \mu|\mu = 1] = \frac{1}{2}$.

Therefore, the advantage with which $\mathcal{A}$ can break the BDDH assumption is $\left|\frac{1}{2}\Pr[\mu' = \mu|\mu = 0] + \frac{1}{2}\Pr[\mu' = \mu|\mu = 1] - \frac{1}{2}\right| \geq \frac{1}{2} \times (\frac{1}{2} + \epsilon(\ell)) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \geq \frac{1}{2}\epsilon(\ell)$. □

We compare our scheme with other multi-authority schemes in Table 5.1 and Table 5.2. By $|\mathbb{U}|$, $|\mathbf{A}_\mathcal{U}|$ and $|\mathbf{A}_C|$, we denote the number of the universal attributes, the attributes held by a user $\mathcal{U}$ and the attributes required by a ciphertext, respectively. $\mathbf{I}_\mathcal{U}$ and $\mathbf{I}_C$ denote the index set of the authorities such that $\mathbf{A}^i_\mathcal{U} \neq \{\phi\}$ and

$\mathbf{A}_C^i \neq \{\phi\}$, respectively. N denotes the number of the authorities in the systems. By d, we denote the number of the central authorities in [LCH+11].

Collusion Resistance. To be secure against the collusion attacks, a user's identifier u is bound with his secret keys and the second secret keys of the authorities so that his secret keys can be tied together. When encrypting a message, all the second public keys of the authorities A_i with $i \in \mathbf{I}_C$ are aggregated and randomized by the value s. Hence, only the secret keys generated for the same identifier can be used to decrypt the ciphertext. The secret keys of different identifiers cannot be combined as C_3 cannot be split by the malicious users. Suppose that $\mathbf{I}_C = \mathbf{I}_{C'} \bigcup \mathbf{I}_{C''}$ and two users $\mathcal{U}_1$ and $\mathcal{U}_2$ obtain secret keys for the attributes which satisfy the access structures specified by the authorities with the indexes in $\mathbf{I}_{C'}$ and $\mathbf{I}_{C''}$, respectively. If they cooperate to decrypt the ciphertext, they must compute $C_3' = \prod_{i \in \mathbf{I}_{C'}} g^{\beta_i s}$ and $C_3'' = \prod_{i \in \mathbf{I}_{c''}} g^{\beta_i s}$. Unfortunately, both C_3' and C_3'' cannot be obtained from C_3 as the exponent s is unknown.

Fine-Grained Access control. In our decentralized KP-ABE scheme, a threshold access structure can be implemented. In order to express any access structure, we exploit the *access tree* technique introduced by Goyal, Pandey, Sahai and Waters [GPSW06]. Let $\mathcal{T}$ be a tree which specifies an access structure, and defines an ordering between the children of every node τ from 1 to n_τ, where n_τ denotes the number of the children of the node τ. Each non-leaf node in $\mathcal{T}$ represents a threshold gate which consists of the number of its children n_τ and a threshold value k_τ with $1 \leq k_\tau \leq n_\tau$. When $k_\tau = 1$, the threshold gate is an OR gate. While, if $k_\tau = n_\tau$, the threshold gate is an AND gate. Furthermore, each leaf node in $\mathcal{T}$ is labeled with an attribute and a threshold value $k_\tau = 1$. Given an access structure, a polynomial $p_\tau(x) \in \mathbb{Z}_p[x]$ is selected for each node in $\mathcal{T}$ following the way in a top-down manner. Beginning from the root node ϱ, set the degree d_τ of the polynomial to be $k_\tau - 1$. In our case, we can set $p_\varrho(0) = r_i$ for the authority A_i. For other nodes in $\mathcal{T}$, we can set $q_\tau(0) = q_{parent(\tau)}(index(\tau))$, where $parent(\tau)$ denotes the parent node of τ, and $index(\tau)$ denotes the number labeled to the node τ.

5.3.2 Privacy-Preserving Key Extract Protocol

A privacy-preserving key extract protocol for the proposed decentralized KP-ABE is described in Figure 5.2.

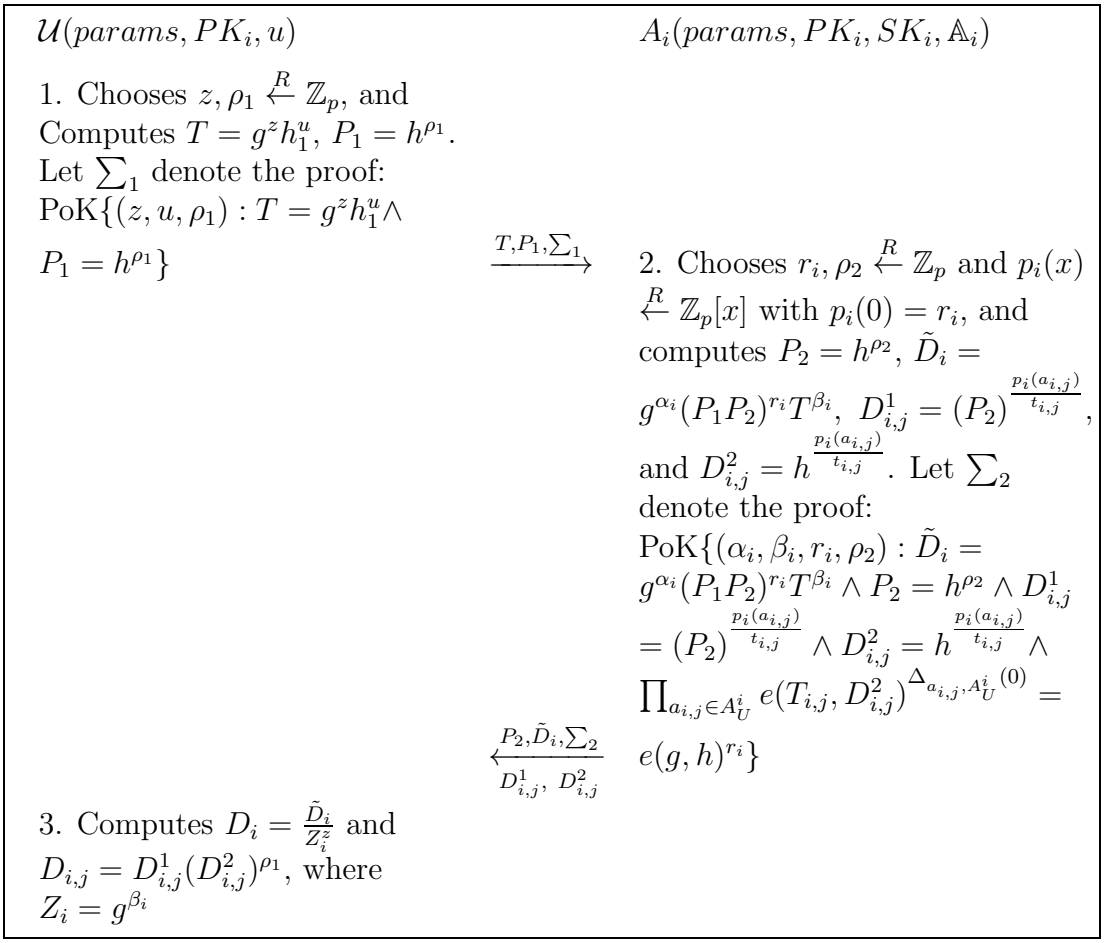

Figure 5.2: Privacy-Preserving Key Extract Protocol BlindKeyGen for Our Decentralized KP-ABE Scheme in Figure 5.1

Overview. In Figure 5.1, the secret keys for a user $\mathcal{U}$ with an identifier u are $D_i = g^{\alpha_i} h^{r_i} h_1^{u\beta_i}$ and $\{D_{i,j} = h^{\frac{p_i(a_{i,j})}{t_{i,j}}}\}_{a_{i,j} \in \mathbf{A}^i_{\mathcal{U}}}$. To obtain secret keys from an authority A_i blindly, $\mathcal{U}$ should prove that he holds the identifier u in zero-knowledge. Notably, if the random number r_i is chosen by A_i independently, he can detect the user by computing $h_1^u = (\frac{D_i}{g^{\alpha_i} h^{r_i}})^{\beta_i^{-1}}$ since the identifer u is public. Hence, the random number used to generate secret keys for $\mathcal{U}$ should be computed by executing a 2-party secure computing between $\mathcal{U}$ and A_i. As a result, $\mathcal{U}$ can obtain his secret keys from A_i blindly without releasing anything about his identifier to him.

In our scheme, $\mathcal{U}$ chooses $z, \rho_1 \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $\Gamma = g^z h_1^u$ and $P_1 = h^{\rho_1}$. Actually, Γ is a commitment of the identifier u and can be used by $\mathcal{U}$ to prove that u has been included in it in zero-knowledge. P_1 will be used to execute a 2-party secure computing with A_i. The user proves that he knows z, u, ρ_1 to A_i in zero-knowledge.

If the proof is correct, A_i selects $r_i, \rho_2 \overset{R}{\leftarrow} \mathbb{Z}_p$ and a (k_i-1)-degree polynomial $p_i(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ with $p_i(0) = r_i$. A_i computes $P_2 = h^{\rho_2}$, $\tilde{D}_i = g^{\alpha_i}(P_1P_2)^{r_i}\Gamma^{\beta_i} = g^{\alpha_i}h^{r_i(\rho_1+\rho_2)}h_1^{u\beta_i}g^{z\beta_i}$, $D_{i,j}^1 = (P_2)^{\frac{p_i(a_{i,j})}{t_{i,j}}}$ and $D_{i,j}^2 = h^{\frac{p_i(a_{i,j})}{t_{i,j}}}$. Actually, P_1 and P_2 are used to compute the exponential $r_i(\rho_1+\rho_2)$ by executing a 2-party secure computing. In this case, the secret key for an attribute $a_{i,j} \in \mathbf{A}_{\mathcal{U}} \bigcap \tilde{\mathbf{A}}_i$ should be $D_{i,j} = h^{\frac{(\rho_1+\rho_2)p_i(a_{i,j})}{t_{i,j}}}$. Unfortunately, A_i does not know ρ_1. Therefore, he computes $D_{i,j}^1 = (P_2)^{\frac{p_i(a_{i,j})}{t_{i,j}}} = h^{\frac{\rho_2 p_i(a_{i,j})}{t_{i,j}}}$ and $D_{i,j}^2 = h^{\frac{p_i(a_{i,j})}{t_{i,j}}}$ so that $\mathcal{U}$ can compute $D_{i,j}$ from $D_{i,j}^1$, $D_{i,j}^2$ and ρ_1. A_i responds $\mathcal{U}$ with $(P_2, \tilde{D}_i, \{D_{i,j}^1, D_{i,j}^2\}_{a_{i,j}\in\mathbf{A}_{\mathcal{U}}^i})$ and proves that he knows $(r_i, \rho_2, \alpha_i, p_i(x), \{t_{i,j}\}_{a_{i,j}\in\mathbf{A}_{\mathcal{U}}^i})$ in zero-knowledge.

If the proof is correct, $\mathcal{U}$ can compute his secret keys as $D_i = \frac{\tilde{D}_i}{Z_i^z} = g^{\alpha_i}h^{r_i(\rho_1+\rho_2)}h_1^{u\beta_i}$ and $D_{i,j} = D_{i,j}^1(D_{i,j}^2)^{\rho_1} = h^{\frac{(\rho_1+\rho_2)p_i(a_{i,j})}{t_{i,j}}}$.

In the BlindKeyGen protocol, $\mathcal{U}$ obtains his secret key $SK_{\mathcal{U}}^i = (D_i, \{D_{i,j}\}_{a_{i,j}\in\mathbf{A}_{\mathcal{U}}^i})$ from A_i, where $D_i = g^{\alpha_i}h^{r_i(\rho_1+\rho_2)}h_1^{u\beta_i}$ and $D_{i,j} = h^{\frac{p_i(a_{i,j})(\rho_1+\rho_2)}{t_{i,j}}}$. The value $r_i(\rho_1+\rho_2)$ is computed by $\mathcal{U}$ and A_i executing a 2-party secure computing, where α_i, β_i, r_i and ρ_2 are from A_i and ρ_1 is from $\mathcal{U}$. Hence, from the view of A_i, D_i and $D_{i,j}$ are identically distributed in the group $\mathbb{G}$.

The details of the protocol in Figure 5.2 are as follows:

1. $\mathcal{U}$ selects $\rho_1, z, z_1, z_2, z_3 \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $\Gamma = g^z h_1^u$, $P_1 = h^{\rho_1}$, $\Gamma' = g^{z_1}h_1^{z_2}$ and $P_1' = h^{z_3}$. $\mathcal{U}$ sends $(\Gamma, P_1, \Gamma', P_1')$ to A_i.

2. A_i chooses $c \overset{R}{\leftarrow} \mathbb{Z}_p$, and responds $\mathcal{U}$ with c.

3. $\mathcal{U}$ computes $s_1 = z_1 - cz$, $s_2 = z_2 - cu$ and $s_3 = z_3 - c\rho_1$, and responds A_i with (s_1, s_2, s_3).

4. A_i verifies $\Gamma' \overset{?}{=} g^{s_1}h_1^{s_2}\Gamma^c$ and $P_1' \overset{?}{=} h^{s_3}P_1^c$. If so, A_i selects $r_i, \rho_2, w, b_1, b_2, b_3, d_j \overset{R}{\leftarrow} \mathbb{Z}_p$ and a (k_i-1)-degree polynomial $p_i(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ with $p_i(0) = r_i$, and computes $P_2 = h^{\rho_2}$, $P_2' = h^w$, $\tilde{D}_i = g^{\alpha_i}(P_1P_2)^{r_i}\Gamma^{\beta_i}$, $D_{i,j}^1 = P_2^{\frac{p_i(a_{i,j})}{t_{i,j}}}$, $D_{i,j}^2 = h^{\frac{p_i(a_{i,j})}{t_{i,j}}}$, $\tilde{D}_i' = g^{b_1}(P_1P_2)^{b_2}\Gamma^{b_3}$, $Z_i' = g^{b_3}$, $Z' = e(g,h)^{b_2}$, $V_j^1 = P_2^{d_j}$ and $V_j^2 = h^{d_j}$. A_i responds $\mathcal{U}$ with $(D_{i,j}^1, D_{i,j}^2, P_2, P_2', Z_i', Z', \tilde{D}_i, \tilde{D}_i', V_j^1, V_j^2)$. Otherwise, A_i aborts.

5. $\mathcal{U}$ selects $c' \overset{R}{\leftarrow} \mathbb{Z}_p$, and responds A_i with c'.

6. A_i computes $\gamma_1 = b_1 - c'\alpha_i$, $\gamma_2 = b_2 - c'r_i$, $\gamma_3 = b_3 - c'\beta_i$, $\gamma_4 = w - c'\rho_2$, and $\eta_j = d_j - c'\frac{p_i(a_{i,j})}{t_{i,j}}$. A_i responds $\mathcal{U}$ with $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \eta_j)$.

7. $\mathcal{U}$ computes $Z = \prod_{a_{i,j} \in A_U^i} e(T_{i,j}, D_{i,j}^2)^{\Delta_{a_{i,j}, A_U^i}(0)}$, and verifies $P_2' \stackrel{?}{=} h^{\gamma_4} P_2^{c'}$, $Z' \stackrel{?}{=} e(g,h)^{\gamma_2} Z^{c'}$, $V_j^1 \stackrel{?}{=} P_2^{\eta_j} (D_{i,j}^1)^{c'}$, $V_j^2 \stackrel{?}{=} h^{\eta_j} (D_{i,j}^2)^{c'}$ and $\tilde{D}_i' \stackrel{?}{=} g^{\gamma_1} (P_1 P_2)^{\gamma_2} \Gamma^{\gamma_3} \tilde{D}_i^{c'}$. If so, $\mathcal{U}$ computes $D_i = \frac{\tilde{D}_i}{Z_i^z}$ and $D_{i,j} = D_{i,j}^1 (D_{i,j}^2)^{\rho_1}$. Otherwise, $\mathcal{U}$ aborts.

The computation cost and communication cost of the privacy-preserving key extract protocol BlindKeyGen is described in Table 5.3.

Theorem 5.2 *The proposed privacy-preserving key extract protocol* BlindKeyGen *in Figure 5.2 is both leak-free and selective-failure blind.*

Proof: We first prove that BlindKeyGen is leak-free.
Leak freeness. Suppose that there exists a PPT adversary $\mathcal{U}^*$ in the real experiment (where $\mathcal{U}^*$ is interacting with an honest authority A_i running the BlindKeyGen protocol), there will exist a simulator $\hat{\mathcal{U}}^*$ in the ideal experiment (where $\hat{\mathcal{U}}^*$ can access the trusted party running the ideal KeyGen protocol) so that no efficient distinguisher $\mathcal{D}$ can distinguish the real experiment from the ideal experiment. $\hat{\mathcal{U}}^*$ simulates the communication between the distinguisher $\mathcal{D}$ and the adversary $\mathcal{U}^*$ by passing the input of $\mathcal{D}$ to $\mathcal{U}^*$ and the output of $\mathcal{U}^*$ to $\mathcal{D}$. $\hat{\mathcal{U}}^*$ works as follows:

1. $\hat{\mathcal{U}}^*$ sends the public key PK_i of A_i to $\mathcal{U}^*$.

2. $\mathcal{U}^*$ must submit two values Γ and P_1, and prove PoK$\{(z,\, u, \rho_1) :\ \Gamma = g^z h_1^u \wedge P_1 = h^{\rho_1}\}$. If the proof fails, $\hat{\mathcal{U}}^*$ aborts the simulation. Otherwise, $\hat{\mathcal{U}}^*$ can obtain (z, u, ρ_1) using the rewinding technique.

3. $\hat{\mathcal{U}}^*$ sends u to the trusted party. The trusted party runs KeyGen to generates $(D_i, \{D_{i,j}\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$, and responds $\hat{\mathcal{U}}$ with them.

4. $\hat{\mathcal{U}}^*$ selects $\lambda \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $\rho_2 = \lambda - \rho_1$, $P_2 = h^{\rho_2}$, $\tilde{D}_i = D_i Z_i^z$, $D_{i,j}^1 = (D_{i,j})^{\frac{\rho_2}{\lambda}}$ and $D_{i,j}^2 = D_{i,j}^{\frac{1}{\lambda}}$. $\hat{\mathcal{U}}^*$ responds $\mathcal{U}^*$ with $(P_2, \tilde{D}_i, D_{i,j}^1, D_{i,j}^2)$.

Therefore, if $(D_i, \{D_{i,j}\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ are correct secret keys from the trusted party in the ideal experiment, $(P_2, \tilde{D}_i, \{D_{i,j}^1, D_{i,j}^2\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ are correct secret keys from A_i in the real experiment. So, $(D_i, \{D_{i,j}\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ and $(P_2, \tilde{D}_i, \{D_{i,j}^1, D_{i,j}^2\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ are distributed identically. Hence, no efficient distinguisher $\mathcal{D}$ can distinguish the real experiment from the ideal experiment.

Now, we prove that BlindKeyGen is selective-failure blind.

Selective-failure blindness. The PPT adversary A_i^* submits the public key PK_i and two global identifiers u_0 and u_1. Then, a bit $b \in \{0,1\}$ is chosen randomly. A_i^* can black-box access $\mathcal{U}(params, PK_i, u_b)$ and $\mathcal{U}(params, PK_i, u_{1-b})$. Subsequently, $\mathcal{U}$ executes the BlindKeyGen protocol with A_i^*, where A_i^* plays the role of the authority A_i. $\mathcal{U}$ outputs secret keys $SK_{\mathcal{U}}^b$ and $SK_{\mathcal{U}}^{1-b}$ for the global identifiers u_b and u_{1-b}, respectively. If $SK_{\mathcal{U}}^b \neq \perp$ and $SK_{\mathcal{U}}^{1-b} \neq \perp$, A_i^* is given $(SK_{\mathcal{U}}^b,\ SK_{\mathcal{U}}^{1-b})$. If $SK_{\mathcal{U}}^b \neq \perp$ and $SK_{\mathcal{U}}^{1-b} = \perp$, A_i^* is given $(\epsilon, \perp)$. If $SK_{\mathcal{U}}^b = \perp$ and $SK_{\mathcal{U}}^{1-b} \neq \perp$, A_i^* is given $(\perp, \epsilon)$. If $SK_{\mathcal{U}}^b = \perp$ and $SK_{\mathcal{U}}^{1-b} = \perp$, A_i^* is given $(\perp, \perp)$. Finally, A_i^* outputs his prediction b' on b.

In the BlindKeyGen protocol, $\mathcal{U}$ sends A_i^* two random values $\Gamma, P_1 \in \mathbb{G}$ and the proof $\text{PoK}\{(z, u_b, \rho_1) : \Gamma = g^z h_1^{u_b} \wedge P_1 = h^{\rho_1}\}$. Supposed that A_i^* runs one or both of the oracles up to this point. Now, it is A_i^*'s turn to respond. So far, A_i^*'s view on the two oracles is computationally indistinguishable. Otherwise, the hiding property of the commitment scheme and the witness undistinguishable property of the zero-knowledge proof will be broken. Suppose that A_i^* uses any computing strategy to output the secret keys $(P_2, \tilde{D}_i, \{D_{i,j}^1, D_{i,j}^2\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ for the first oracle. In the following, we will show that A_i^* can predict $SK_{\mathcal{U}}^b$ of $\mathcal{U}$ without interaction with the two oracles:

1. A_i^* checks

$$\text{PoK}\left\{ (\alpha_i, \beta_i, r_i, \rho_2) : \begin{array}{l} \tilde{D}_i = g^{\alpha_i}(P_1 P_2)^{r_i}\Gamma^{\beta_i} \wedge P_2 = h^{\rho_2} \wedge \\ D_{i,j}^1 = (P_2)^{\frac{p_i(a_{i,j})}{t_{i,j}}} \wedge D_{i,j}^2 = h^{\frac{p_i(a_{i,j})}{t_{i,j}}} \wedge \\ \prod_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i} e(T_{i,j}, D_{i,j}^2)^{\Delta_{a_{i,j}, \mathbf{A}_{\mathcal{U}}^i}(0)} = e(g,h)^{r_i} \end{array} \right\}.$$

   If the proof fails, A_i^* sets $SK_{\mathcal{U}}^0 = \perp$.

2. A_i^* generates different $(P_2, \tilde{D}_i, \{D_{i,j}^1, D_{i,j}^2\}_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i})$ for the second oracle and a proof of knowledge:

$$\text{PoK}\left\{ (\alpha_i, \beta_i, r_i, \rho_2) : \begin{array}{l} \tilde{D}_i = g^{\alpha_i}(P_1 P_2)^{r_i}\Gamma^{\beta_i} \wedge P_2 = h^{\rho_2} \wedge \\ D_{i,j}^1 = (P_2)^{\frac{p_i(a_{i,j})}{t_{i,j}}} \wedge D_{i,j}^2 = h^{\frac{p_i(a_{i,j})}{t_{i,j}}} \wedge \\ \prod_{a_{i,j} \in \mathbf{A}_{\mathcal{U}}^i} e(T_{i,j}, D_{i,j}^2)^{\Delta_{a_{i,j}, \mathbf{A}_{\mathcal{U}}^i}(0)} = e(g,h)^{r_i} \end{array} \right\}.$$

   A_i^* checks the proof. If it fails, A_i^* sets $SK_{\mathcal{U}}^1 = \perp$.

3. Finally, A_i^* outputs his predication on $(u_0,\ u_1)$ with $(SK_{\mathcal{U}}^0, SK_{\mathcal{U}}^1)$, if $SK_{\mathcal{U}}^0 \neq \perp$ and $SK_{\mathcal{U}}^1 \neq \perp$; $(\epsilon, \perp)$, if $SK_{\mathcal{U}}^0 \neq \perp$ and $SK_{\mathcal{U}}^1 = \perp$; $(\perp, \epsilon)$, if $SK_{\mathcal{U}}^0 = \perp$ and $SK_{\mathcal{U}}^1 \neq \perp$; $(\perp, \perp)$, if $SK_{\mathcal{U}}^0 = \perp$ and $SK_{\mathcal{U}}^1 = \perp$.

The predication on $(u_0,\ u_1)$ is correct, and has the identical distribution with the oracle. Because A_i^* performs the same check as the honest $\mathcal{U}$, it outputs the valid secret keys as $\mathcal{U}$ obtains from $\mathsf{BlindKeyGen}(\mathcal{U}(params, PK_i, u) \leftrightarrow A_i(params, PK_i, SK_i))$ when both checks are valid. Hence, if A_i^* can predict the final outputs of the two oracles, the advantage of him in distinguishing $\mathcal{U}(params, PK_i, u_b)$ from $\mathcal{U}(params,\ PK_i, u_{1-b})$ is the same without the final outputs. Therefore, the advantage of A_i^* should come from the received Γ, P_1 and the proof $\mathrm{PoK}\{(z, u_b, \rho_1) :\ \Gamma = g^z h_1^{u_b} \wedge P_1 = h^{\rho_1}\}$. Due to the hiding property of the commitment scheme and witness undistinguishable property of the zero-knowledge proof, A_i^* cannot distinguish one from the other with non-negligible advantage.

Therefore, the following theorem can be derived from Theorem 5.1 and Theorem 5.2.

Theorem 5.3 *Our privacy-preserving decentralized attribute-based encryption scheme* $\tilde{\prod} =$*(*Global-Setup, Authority-Setup, BlindKeyGen, Encryption, Decryption*) is secure in the selective-set model under the decisional bilinear Diffie-Hellman assumption.*

5.4 Chapter Summary

Decentralized ABE schemes have attracted a lot of research attention as they can reduce the trust on merely a single centralized authority. In a decentralized ABE scheme, a global identifier GID is used to tie all the user's secret keys from multiple authorities together to resist the collusion attacks. However, this will risk the user being traced and impersonated by the compromised authorities. In this chapter, we proposed a privacy-preserving decentralized ABE scheme to protect the user's privacy. In our scheme, all the user's secret keys are tied to his identifier to resist the collusion attacks, while the multiple authorities have no idea on the user's identifier. Notably, each authority can join or leave the system freely without the necessity of re-initializing the system and there is no central authority. Furthermore, any access structure can be expressed in our scheme using the access tree technique. Finally, our scheme relies on the standard complexity assumption (e.g., DBDH), rather than the non-standard complexity assumption (e.g., DDHI).

Table 5.1: The Comparison of Computation Cost

Schemes	Authority setup	KeyGen	Encryption	Decryption
Chase's scheme [Cha07]	$(\|\mathbb{U}\|+1)T_e$	$(\|\mathbf{A}_\mathcal{U}\|+1)T_e$	$(\|\mathbf{A}_C\|+2)T_e$	$\|\mathbf{A}_C\|T_e+(\|\mathbf{A}_C\|+1)T_p$
MKE's scheme [MKE08]	$2\|\mathbb{U}\|T_e$	$\|\mathbf{A}_\mathcal{U}\|T_e$	$3\|\mathbf{I}_C\|T_e$	$2T_p$
CC scheme [CC09]	$(\|\mathbb{U}\|+2N)T_e$	$(\|\mathbb{U}\|+\|\mathbf{I}_\mathcal{U}\|^2)T_e$	$(\|\mathbf{A}_C\|+2)T_e$	$\|\mathbf{A}_C\|T_e+(\|\mathbf{A}_C\|+1)T_p$
LW's scheme [LW11]	$2NT_e$	$2\|\mathbf{A}_\mathcal{U}\|T_e$	$(5\|\mathbf{A}_C\|+1)T_e$	$3\|\mathbf{A}_C\|(T_e+T_p)$
LCHWY scheme [LCH$^+$11]	$(\|\mathbb{U}\|+N)T_e)$	$(4d+\|\mathbf{A}_\mathcal{U}\|)T_e+\|\mathbf{I}_\mathcal{U}\|T_p$	$(3\|\mathbf{A}_C+2\|T_e)$	$(\|\mathbf{A}_C\|+1)T_e+2\|\mathbf{A}_C\|T_p$
Our scheme	$(\|\mathbb{U}\|+2N)T_e$	$(\|\mathbf{A}_\mathcal{U}\|+3\|\mathbf{I}_\mathcal{U}\|)T_e$	$(\|\mathbf{A}_C\|+3)T_e$	$\|\mathbf{A}_C\|T_e+(\|\mathbf{A}_C\|+\|\mathbf{I}_C\|+1)T_p$

Table 5.2: The Comparison of Type, Central Authority, Security Model and the Length of Ciphertext

Schemes	KP/CP-ABE	Central Authority	Security Model	Length of Ciphertext
Chase's scheme [Cha07]	KP-ABE	Yes	Selective-set	$(\|\mathbf{A}_C\|+1)E_\mathbb{G}+E_{\mathbb{G}_\tau}$
MKE's [MKE08]	CP-ABE	Yes	Full security	$2\|\mathbf{I}_C\|E_\mathbb{G}+\|\mathbf{I}_C\|E_{\mathbb{E}_\tau}$
CC scheme [CC09]	KP-ABE	No	Selective-set	$(\|\mathbf{A}_C\|+1)E_\mathbb{G}+E_{\mathbb{G}_\tau}$
LW's scheme [LW11]	CP-ABE	No	Full security	$2\|\mathbf{A}_C\|E_\mathbb{G}+(\|\mathbf{A}_C\|+1)E_{\mathbb{G}_\tau}$
LCHWY scheme [LCH$^+$11]	CP-ABE	No	full security	$(2\|\mathbf{A}_C\|+1)E_\mathbb{G}+E_{\mathbb{G}_\tau}$
Our scheme	KP-ABE	No	Selective-set	$(\|\mathbf{A}_C\|+2)E_\mathbb{G}+E_{\mathbb{G}_\tau}$

Table 5.3: The Computing Cost and Communication Cost of the Privacy-Preserving Key Extract Protocol

Scheme	Computation Cost		Communication Cost	
	U	A_i	$U\rightarrow A_i$	$U\leftarrow A_i$
Our scheme	$(14+5\|\mathbf{A}^i_\mathcal{U}\|)T_e+\|\mathbf{A}_\mathcal{U}\|T_p$	$(15+4\|\mathbf{A}^i_\mathcal{U}\|)T_e$	$4E_\mathbb{G}+4E_{\mathbb{Z}_p}$	$(5+\|\mathbf{A}^i_\mathcal{U}\|)E_{\mathbb{Z}_p}+(5+4\|\mathbf{A}^i_\mathcal{U}\|)E_\mathbb{G}+E_{\mathbb{G}_\tau}$

Chapter 6

Attribute-based Data Transfer with Filtering in Distributed Systems

In this chapter, we proposed an attribute-based data transfer with filtering (AB-DTF) scheme in distributed systems.

6.1 Introduction

In complicated data transfer systems, such as cloud computing [YWRL10] and wireless sensor networks (WSN) [AK04, NN08], the confidentiality and efficiency of the transferred data have been primarily focused. Some schemes toward to provide these two properties have been proposed in the literature [Sha84, BF01, ZSJN04, SW05, LLZ+10, Wat11]. While, in order to send sensitive data to the intended receivers, a sender must know all the identities (or public keys) of the receivers and communicate with them separately [Sha84, BF01, ZSJN04, LLZ+10]. Furthermore, a receiver cannot determine whether a message is from a legal sender [Sha84, BF01, SW05, Wat11], since anyone who knows his identity can send messages to him. These problems are particularly serious in the systems with numerous users. To clarify these issues, we provide the following scenario. In cloud computing, a user is unable to know and communicate with all the other users as the number of users in the system are very large. In the scenario that a user would like to purchase a personal computer with attributes $PC = \{Brand = Apple,\ Year = 2011\}$, he must set conversations with the multiple unknown sellers. A sound solution is that the user can specify an access structure such that only the sellers whose product attributes satisfy this access structure can contact him and negotiate with him. This system will not only protect the user's privacy, but also reduce the communication cost. Meanwhile, if a seller sells the machine with attributes $PC = \{Brand = Apple,\ Price \leq 5000,\ Type =$

$Student,\ Year = 2011\}$, he will not do any deal with the buyer who is *not* a student. Otherwise, he will face the denial-of-service (DoS) attacks [MVS01], as many buyers who do not hold the attributes required by the sell can also contact him. DoS attacks are initialized by malicious adversaries to consume the resource of the host or network so that legal users cannot be serviced. DoS attacks can be classified into two types [MVS01]: logic attacks and flooding attacks. In the logic attacks, the adversaries use the flaws in the exploited software to degrade its performance. While, in the flood attacks, the adversaries send or inject lots of false messages to consume a user's resource or paralyze the system. Consequently, filtering schemes [Blo70, Far75] are proposed to resist DoS attacks. A receiver can efficiently filter out the false messages prior to processing them.

In this chapter, we introduce a filtering scheme to an attribute-based data transfer scheme to save the receiver from the DoS attacks and protect the sender's privacy.

6.1.1 Related Work

In this section, we introduce the literature about attribute-based data transfer with filtering (ABDTF) scheme.

Attribute-based Encryption

This is referred to Section 5.1.1.

Data Transfer with filtering schemes

Filtering is an efficient tool to help a receiver filter out the false data [Blo70, Lit74, Yue77, Mit02], and has been used to resist DoS attacks in distributed systems.

Bloom [Blo70] proposed a filtering scheme using the hashing-code methods to detect the membership in a set of messages. Subsequently, Mitzenmacher [Mit02] proposed a compressed filtering scheme to improve the perfermance and transmission of the scheme [Blo70].

Little [Lit74] proposed an efficient algorithm for nonrecursive and recursive digital filtering, where the filtering speed depends on the memory space and the filtering time is independent of the order of the filtering. Yuen [Yue77] improved Little's scheme by expressing the data with two complement forms, instead of the *biased* form.

Filtering schemes which can be used to filter out the false reports in a WSN system have been proposed. To name a few, Ye, Luo, Lu and Zhang [YLLZ04] proposed a statistical en-route filtering scheme to filter out the false reports during the forwarding process in a WSN system. In their scheme, each sensor creates a keyed message authentication code (MAC). For an event report, multiple MACs are attached to it. As the report is forwarded, the sensors validate the MACs probabilistically and determine whether it is false.

Zhu, Setia, Jajodia and Ning [ZSJN04] proposed an interleaved hop-by-hop authentication scheme. In their scheme, a false report can be detected by the base station (sink) if no more than a certain number of sensors are corrupted. In the scenario that the number of the compromised sensors are under the certain number, they gave an upper bound for the number of hops that a false report can be forwarded prior to being detected.

Yang, Ye, Yuan, Lu and Arbaugh [YYY+05] proposed a location-based filtering scheme where a key is bound to the geographic location to resist the compromised sensors to compute a false report. Ren, Lou and Zhang [RLZ06] proposed a location-aware end-to-end data transfer scheme where not only the false reports can be detected, but also end-to-end security can be provided. Both [YYY+05] and [RLZ06] are based on symmetric-key systems where each sensor must share a key with his upper and lower sensors. Zhang, Liu, Lou and Fang [ZLLF06] proposed a location-based compromise-tolerant filtering scheme based on public-key systems.

Yu and Guan [YG10] proposed a dynamic en-rout filtering scheme where each sensor validates reports by a keyed hash chain. They exploited the *hill climbing* key distribution technique to guarantee that the sensors close to the sink have strong filtering ability, and broadcast property to resist DoS attacks.

Lu, Lin, Zhu, Liang and Shen [LLZ+10] proposed a bandwidth-efficient cooperative authentication mechanism with filtering. They introduced a random graph characteristics of sensor nodes and a cooperative bit-compressed authentication scheme to a WSN system. As a result, the energy of detecting a false report can be saved and the burden of the sink can be reduced.

6.1.2 Our Contribution

ABE schemes have been used as a building block to design data transfer schemes in distributed systems as they do not depend on the public-key infrastructure

(PKI). However, distributed systems are subject to DoS attacks. These attacks not only consume users' resources, but also paralyze the system. Therefore, it is an interesting work to construct an ABDTF scheme where DoS attacks can be resisted. In this chapter, we formalize the definition and security model of ABDTF schemes, and propose an efficient ABDTF scheme. In our scheme, a sender can encrypt a message under a set of attributes such that only the receivers who hold these attributes can obtain the message. Furthermore, the receiver can also specify an access structure such that only the senders whose attributes satisfy this access structure can send messages to him. Prior to processing the received messages, the receiver can efficiently filter out the false ones. This idea is related to the idea of attribute-based authenticated encryption. Notably, the receiver can update his access structure dynamically without the necessity of re-initializing the system and re-issuing the secret keys to the users. Finally, the authentication key stored by the receiver and the authentication information from the sender are short. Especially, the authentication key and the authentication information can be computed off-line by the receiver and the sender, respectively. To the best of our knowledge, it is the *first* time that a provable ABDTF scheme is proposed. Therefore, our work provides a formal treatment on the research of ABDTF schemes.

6.1.3 Chapter Organization

In Section 6.2, we formalize the definition and security model of ABDTF schemes. An ABDTF scheme is proposed and proven in Section 6.3. Finally, Section 6.4 summaries this chapter.

6.2 Formal Definition and Security Model

In this section, we introduce the formal definition and security model of ABDTF schemes.

6.2.1 Formal Definition

An ABDTF scheme consists of the following five algorithms:

$\mathsf{Setup}(1^\ell) \to (params, MSK)$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$ and a master secret key MSK.

$\mathsf{KeyGen}(params, \mathbf{A}_{\mathcal{U}}, MSK) \rightarrow SK_{\mathcal{U}}$. The key generation algorithm takes as input the public parameters $params$, a set of attributes $\mathbf{A}_{\mathcal{U}}$ and the master secret key MSK, and outputs a secret key $SK_{\mathcal{U}}$ for a user $\mathcal{U}$ with attributes $\mathbf{A}_{\mathcal{U}}$.

$\mathsf{Receiver\text{-}Policy}(params, \mathbf{R}) \rightarrow (\mathbb{A}_{\mathcal{R}}, AK_{\mathcal{R}})$. The receiver policy algorithm takes as input the public parameters $params$ and a set of attributes $\mathbf{R}$, and outputs an access structure $\mathbb{A}_{\mathcal{R}}$ and an authentication key $AK_{\mathcal{R}}$.

$\mathsf{Enc}(params, \mathbf{A}_C, M, \mathbb{A}_R, SK_{\mathcal{S}}) \rightarrow (CT, AI)$. The encryption algorithm takes as input the public parameters $params$, a set of attributes $\mathbf{A}_C$, a message M, an access structure $\mathbb{A}_R$ and the sender's secret key $SK_{\mathcal{S}}$, and outputs a ciphertext CT and an authentication information AI. CT can be decrypted by the receiver who holds a set of attributes $\mathbf{A}_{\mathcal{R}}$ if $\mathbf{A}_C \subseteq \mathbf{A}_{\mathcal{R}}$.

Dec.

1. $\mathsf{Filter}(params, AK_{\mathcal{R}}, AI) \rightarrow True/False$. The filtering algorithm takes as input the public parameters $params$, the authentication key $AK_{\mathcal{R}}$ and the authentication information AI, and outputs $True$ if the attributes of the sender $\mathbf{A}_{\mathcal{S}}$ satisfy the access structure $\mathbb{A}_{\mathcal{R}}$. Otherwise, it outputs $False$ and aborts the protocol.
2. $\mathsf{Dec}(params, SK_{\mathcal{R}}, CT) \rightarrow M$. The decryption algorithm takes as input the public parameters $params$, the receiver's secret key $SK_{\mathcal{R}}$ and the ciphertext CT, and outputs the message M.

Definition 6.1 *We say that an attribute-based data transfer with filtering scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}(params, SK_{\mathcal{R}}, CT) = M & \begin{array}{l} \mathsf{Setup}(1^{\ell}) \rightarrow (params, MSK); \\ \mathsf{KeyGen}(params, \mathbf{A}_{\mathcal{U}}, MSK) \rightarrow SK_{\mathcal{U}}; \\ \mathsf{Receiver-Policy}(params, \mathbf{R}) \rightarrow (\mathbb{A}_{\mathcal{R}}, AK_{\mathcal{R}}); \\ \mathsf{Enc}(params, \mathbf{A}_C, M, \mathbb{A}_{\mathcal{R}}, SK_{\mathcal{S}}) \rightarrow (CT, AI); \\ \mathsf{Filter}(params, AK_{\mathcal{R}}, AI) \rightarrow True; \\ \mathbf{A}_C \subseteq \mathbf{A}_{\mathcal{R}} \text{ and } \mathbf{A}_{\mathcal{S}} \in \mathbb{A}_{\mathcal{R}} \end{array} \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by the algorithms in the scheme.

6.2.2 Security Model

The security model of ABDTF schemes is formalized by the following two games executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The first game is about the security of the ciphertext. It is similar to the selective-set security model in [SW05]. The second game is about the security of the filtering. It is used to formalize the model that only the sender $\mathcal{S}$ whose attributes satisfy the access structure $\mathbb{A}_{\mathcal{R}}$ specified by the receiver $\mathcal{R}$ can send messages to him.

Game 1: Selective-set Model.

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{A}_C$ which he wants to be challenged with.

Setup. $\mathcal{C}$ runs the $\mathsf{Setup}(1^\ell)$ algorithm to generate the public parameters *params* and a secret key MSK. $\mathcal{C}$ responds $\mathcal{A}$ with **params**.

Phase 1. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_{q_1}$, where the only constraint is $\mathbf{A}_C \notin \mathbf{A}_i$ for $j = 1, 2, \cdots, q_1$. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathsf{KeyGen}(params, A_i, MSK)$ for $i = 1, 2, \cdots, q_1$.

Challenge. $\mathcal{A}$ submits two message M_0 and M_1 with equal length. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ computes $CT^* = \mathsf{Ecn}(params, \mathbf{A}_C, M_b)$, and responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_{q_1+1}, \mathbf{A}_{q_1+2}, \cdots, \mathbf{A}_q$, where the only constraint is $\mathbf{A}_C \notin \mathbf{A}_i$ for $i = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 6.2 *An attribute-based data transfer with filtering scheme is $(T, q, \epsilon(\ell))$-secure against chosen plaintexts attacks (or IND-CPA) if no PPT adversary $\mathcal{A}$ making at most q secret key queries can win the game with the advantage*

$$Adv_{\mathcal{A}-ABDTF}^{IND-CPA} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above selective-set model.

Game 2: Filtering Security Model.

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{R}$ which he wants to be challenged with.

Setup. $\mathcal{C}$ runs the $\mathsf{Setup}(1^\ell)$ algorithm to generate the public parameters $params$ and a secret key MSK. $\mathcal{C}$ responds $\mathcal{A}$ with $params$.

Phase 1. $\mathcal{A}$ can adaptively query authentication information for sets of attributes $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_{q_1}$, where the only constraint is $\mathbf{R} \notin \mathbf{A}_i$ for $i = 1, 2, \cdots, q_1$. $\mathcal{C}$ responds with the corresponding authentication information AI_i for $i = 1, 2, \cdots, q_1$.

Challenge. $\mathcal{C}$ runs the $\mathsf{Receiver\text{-}Policy}(params, \mathbf{R})$ algorithm to generates $(\mathbb{A}_\mathcal{R}, AK_\mathcal{R})$, where $\mathbb{A}_\mathcal{R}$ is an access structure for the attributes in $\mathbf{R}$ and $AK_\mathcal{R}$ is an authentication key for $\mathbb{A}_\mathcal{R}$. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathbb{A}_\mathcal{R}$.

Phase 2. $\mathcal{A}$ can adaptively query authentication information for sets of attributes $\mathbf{A}_{q_1+1}, \mathbf{A}_{q_1+2}, \cdots, \mathbf{A}_q$, where $\mathbf{R} \notin \mathbf{A}_i$ for $i = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated

Output. $\mathcal{A}$ outputs an authentication information $AI_\mathcal{R}$ for the access structure $\mathbb{A}_\mathcal{R}$. $\mathcal{A}$ wins the game if $\mathsf{Filter}(AK_\mathcal{R}, AI_\mathcal{R}) \to True$.

Definition 6.3 *An attribute-based data transfer with filtering scheme is $(T, q, \epsilon(\ell))$-filtering secure if no PPT adversary $\mathcal{A}$ making at most q authentication information queries can win the game with the advantage*

$$Adv^{Filtering}_{\mathcal{A}-ABDTF} = \Pr\left[\mathsf{Filter}(AK_\mathcal{R}, AI_\mathcal{R}) \to True\right] \geq \epsilon(\ell)$$

in the above model.

6.3 Attribute-based Data Transfer With Filtering

In this section, we propose an ABDTF scheme and prove its security in the proposed security model.

Overview. We introduce a filtering scheme to a KP-ABE scheme to resist the DoS attacks. In our scheme, at first, the CA specifies a (k, n)-threshold access structure $\mathbb{A}$. Then, he issues secret keys to users according their attributes. Suppose that a receiver $\mathcal{R}$ and a sender $\mathcal{S}$ hold sets of attributes $\mathbf{A}_\mathcal{R}$ and $\mathbf{A}_\mathcal{S}$, respectively. To resist

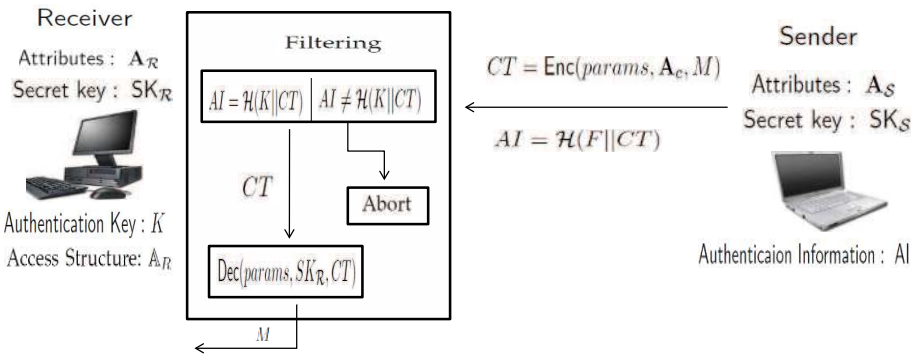

Figure 6.1: The Model of Attribute-based Data Transfer With Filtering Scheme

the DoS attacks, $\mathcal{R}$ specifies a (k, ρ)-threshold access structure $\mathbb{A}_{\mathcal{R}}$ such that only the users (senders) whose attributes satisfies $\mathbb{A}_{\mathcal{R}}$ can send messages to him, where $1 \leq \rho \leq n$. $\mathcal{R}$ selects an authentication key K for $\mathbb{A}_{\mathcal{R}}$ and encapsulates K in $\mathbb{A}_{\mathcal{R}}$. If $\mathcal{S}$ wants to send a message to $\mathcal{R}$, he must check whether $\mathbf{A}_{\mathcal{S}} \in \mathbb{A}_{\mathcal{R}}$. If $\mathbf{A}_{\mathcal{S}} \in \mathbb{A}_{\mathcal{R}}$, $\mathcal{S}$ can reconstruct K using his secret keys. Subsequently, $\mathcal{S}$ encrypts a message under a set of attributes $\mathbf{A}_C$ and computes an authentication information AI which is the hash value of K and the ciphertext CT. Actually, AI is a MAC of K and CT. $\mathcal{S}$ sends CT and AI to $\mathcal{R}$. Receiving (AI, CT) from $\mathcal{S}$, $\mathcal{R}$ validates AI using K and CT. If the AI is valid, $\mathcal{R}$ checks whether he holds the attributes listed in CT and decrypt it useing his secret keys. Otherwise, $\mathcal{R}$ treats the received (AI, CT) as a false message and aborts. We explain our model in Figure 6.1.

We describe our ABDTF scheme in Figure 6.2.

Correctness. The following equations hold.

$$\begin{aligned} F_{\mathcal{S}} &= e(D_{\mathcal{S}}, W) \\ &= e(g^{\alpha} h^{\sigma_s}, g^{w}) \\ &= e(g, g)^{\alpha w} \cdot e(g, h)^{w\sigma_s}, \end{aligned}$$

$$\begin{aligned} F_{v_j} &= e(D_{\mathcal{S}, v_j}, E_{v_j}) \\ &= e(h^{\frac{p(a_{v_j})}{t_{v_j}}}, g^{w t_{v_j}})^{\Delta_{\mathcal{Q}, a_{v_j}}(0)} \\ &= e(g, h)^{w p(a_{v_j}) \Delta_{\mathcal{Q}, a_{v_j}}(0)}, \end{aligned}$$

$$\begin{aligned} e(D_{\mathcal{R}}, C_1) &= e(g^{\alpha} h^{\sigma_r}, g^{s}) \\ &= e(g, g)^{\alpha s} \cdot e(g, h)^{s\sigma_r} \end{aligned}$$

and

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. It also generates a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\lambda$, where λ is linear in ℓ. Let g and h be generators of $\mathbb{G}$. Suppose that the universal attribute set $\mathbb{U} = \{a_1, a_2, \cdots, a_n\} \in \mathbb{Z}_p^n$. It selects $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $Y = e(g,g)^\alpha$. For each attribute $a_i \in \mathbb{U}$, it chooses $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $T_i = g^{t_i}$. The master secret key is $(\alpha, t_1, t_2, \cdots, t_n)$, while the public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, h, Y, T_1, \cdots, T_n)$.

KeyGen. To generate a secret key for a user $\mathcal{U}$ with a set of attributes $\mathbf{A}_\mathcal{U}$, this algorithm chooses $\sigma_u \overset{R}{\leftarrow} \mathbb{Z}_p$ and a $(k-1)$ degree polynomial $p(x) \overset{R}{\leftarrow} \mathbb{Z}_p[x]$ with $p(0) = \sigma_u$ and computes

$$D_\mathcal{U} = g^\alpha h^{\sigma_u} \text{ and } \{D_{\mathcal{U},i} = h^{\frac{p(a_i)}{t_i}}\}_{a_i \in \mathbf{A}_\mathcal{U}}.$$

The secret key for $\mathcal{U}$ is $SK_\mathcal{U} = (D_\mathcal{U}, \{D_{\mathcal{U},i}\}_{a_i \in \mathbf{A}_\mathcal{U}})$.

Receiver-Policy. If a receiver $\mathcal{R}$ only wants to receive messages from senders who hold k-out-of-ρ attributes $\mathbf{R} = \{a_{j_1}, a_{j_2}, \cdots, a_{j_k}, \cdots, a_{j_\rho}\} \subseteq \mathbb{U}$, he chooses $w \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $K = e(g,g)^{\alpha w}$, $W = g^w$ and $\{E_{j_c} = T_{j_c}^w\}_{c=1}^\rho)$, where $1 \leq k \leq \rho \leq n$. $\mathcal{R}$ keeps K as an authentication key, and publishes the access structure $\mathbb{A}_\mathcal{R} = (W,\ \{a_{j_c},\ E_{j_c}\}_{c=1}^\rho)$.

Encryption. To encrypt a message $M \in \mathbb{G}_\tau$ under a set of attributes $\mathbf{A}_C$, a sender $\mathcal{S}$ selects $s \overset{R}{\leftarrow} \mathbb{Z}_p$, a set of attributes $\mathcal{Q} = \{a_{v_1}, a_{v_2}, \cdots, a_{v_k}\} \subseteq \mathbf{A}_\mathcal{S} \bigcap \mathbf{R}$ and computes

$$C_0 = M \cdot e(g,g)^{\alpha s},\ C_1 = g^s,\ \{C_x = T_x^s\}_{a_x \in \mathbf{A}_C},$$

$$F_S = e(D_\mathcal{S}, W),\ \{F_{v_j} = e(D_{\mathcal{S},v_j}, E_{v_j})^{\Delta_{\mathcal{Q},a_{v_j}}(0)}\}_{a_{v_j} \in \mathcal{Q}} \text{ and } F = \frac{F_S}{\prod_{a_{v_j} \in \mathcal{Q}} F_{v_j}}$$

where $\mathbf{A}_\mathcal{S}$ and $(D_\mathcal{S}, \{D_{\mathcal{S},v_j}\}_{a_{v_j} \in \mathcal{Q}})$ are the set of attributes held by $\mathcal{S}$ and his partial secret keys, respectively.

The ciphertext is $CT = (C_0,\ C_1,\ \{C_x\}_{a_x \in A_C})$ and the authentication information is $\Gamma = \mathcal{H}(F||CT)$.

Decryption.

1. Filter. Receiving $CT = (C_0,\ C_1,\ \{C_x\}_{a_x \in \mathbf{A}_C})$ and Γ, $\mathcal{R}$ checks whether $\Gamma = \mathcal{H}(K||CT)$ and $\mathbf{A}_\mathcal{R} \in \mathbf{A}_C$. If it is, $\mathcal{R}$ goes to the next step. Otherwise, he aborts.
2. Decrypt. $\mathcal{R}$ computes $M = C_0 \cdot \frac{\prod_{a_x \in \mathbf{A}_C} e(D_{\mathcal{R},x}, C_x)^{\Delta_{\mathbf{A}_C,a_x}(0)}}{e(D_\mathcal{R}, C_1)}$.

Figure 6.2: Attribute-based Data Transfer with Filtering Scheme

$$
\begin{aligned}
& e(D_{\mathcal{R},x}, C_x)^{\Delta_{\mathbf{A}_C,a_x}(0)} \\
&= e(h^{\frac{p(a_x)}{t_x}}, g^{st_x})^{\Delta_{\mathbf{A}_C,a_x}(0)} \\
&= e(g,h)^{sp(a_x)\Delta_{\mathbf{A}_C,a_x}(0)}.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
F &= \frac{F_S}{\prod_{a_{v_j}\in\mathcal{Q}} F_{v_j}} \\
&= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{\prod_{a_{v_j}\in\mathcal{Q}} e(g,h)^{wp(a_{v_j})\Delta_{\mathcal{Q},v_j}(0)}} \\
&= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{e(g,h)^{w\sum_{a_{v_j}\in\mathcal{Q}} p(a_{v_j})\Delta_{\mathcal{Q},a_{v_j}}(0)}} \\
&= \frac{e(g,g)^{\alpha w} e(g,h)^{w\sigma_s}}{e(g,h)^{w\sigma_s}} \\
&= e(g,g)^{\alpha w}
\end{aligned}
$$

and

$$
\begin{aligned}
& C_0 \cdot \frac{\prod_{a_x\in\mathbf{A}_C} e(D_{\mathcal{R},x}, C_x)^{\Delta_{\mathbf{A}_C,a_x}(0)}}{e(D_{\mathcal{R}}, C_1)} \\
&= C_0 \cdot \frac{\prod_{a_x\in\mathbf{A}_C} e(g,h)^{sp(a_x)\Delta_{\mathbf{A}_C,a_x}(0)}}{e(g,g)^{\alpha s}\cdot e(g,h)^{s\sigma_r}} \\
&= C_0 \cdot \frac{e(g,h)^{s\sum_{a_x\in\mathbf{A}_C} p(a_x)\Delta_{\mathbf{A}_C,a_x}(0)}}{e(g,g)^{\alpha s}\cdot e(g,h)^{s\sigma_r}} \\
&= C_0 \cdot \frac{e(g,h)^{s\sigma_r}}{e(g,g)^{\alpha s}\cdot e(g,h)^{s\sigma_r}} \\
&= M \cdot \frac{e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} \\
&= M.
\end{aligned}
$$

Although both the computation costs of the access structure $\mathbb{A}_R$ and the authentication information Γ are linear with the number of the required attributes, $(K,\ W,\ \{E_{j_c}\}_{a_{j_c}\in R})$ and $(F,\ F_S,\ \{F_{v_j}\}_{a_{v_j}\in\mathcal{Q}})$ can be computed by $\mathcal{R}$ and $\mathcal{S}$ off-line, respectively. Notably, $\mathcal{R}$ can update the access structure $\mathbb{A}_{\mathcal{R}}$ dynamically without re-initializing the system and re-issuing secret keys to the users.

In the practical scenario, the filtering algorithm can be executed by a proxy server. $\mathcal{R}$ can determine an access structure, select an authentication key K and delegate it to the proxy server. When receiving a message (Γ, CT), the proxy server

checks $\Gamma \stackrel{?}{=} \mathcal{H}(K||CT)$. If so, he sends CT to $\mathcal{R}$. Otherwise, he filters it out on behalf of $\mathcal{R}$. This is especially necessary in e-mail systems [JP94], where the filtering algorithm can help a user filter out the junk e-mails and reduce jams. The authentication key K can be stored in a software with a limited memory space as it is only one element (512 bites) in the bilinear group.

We compare the computation cost and communication cost of our ABDTF scheme with related schemes in Table 6.1 and Table 6.2. By $|\mathbf{A}_{\mathcal{U}}|$ and $|\mathbf{A}_C|$, we denote the number of the attributes held by a user $\mathcal{U}$ and the number of the attributes listed in the ciphertext, respectively. By $--$, we denote that the item is not suitable for the scheme.

Theorem 6.1 *Our attribute-based data transfer with filtering scheme is $(T, q, \epsilon(\ell))$ secure against the chosen plaintext attacks (or IND-CPA) in the selective-set security model if the $(T', \epsilon'(\ell))$ decisional bilinear Diffie-Hellman assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$, where*

$$T' = T + \Theta(T) \quad \textit{and} \quad \epsilon'(\ell) = \frac{\epsilon(\ell)}{2}.$$

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the ciphertext security of our ABDTF scheme, we can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the DBDH assumption as follows.

The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g be a generator of the group $\mathbb{G}$. He flips an unbiased coin with $\{0, 1\}$ and obtains one bit $\mu \in \{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ to $\mathcal{B}$. Otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \stackrel{R}{\leftarrow} \mathbb{Z}_p$. $\mathcal{B}$ will output his guess μ' on μ.

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{A}_C$ which he wants to be challenged with.

Setup. $\mathcal{B}$ sets $Y = e(g, g)^{ab}$ and $h = Ag^\eta$, where $\eta \stackrel{R}{\leftarrow} \mathbb{Z}_p$. If $a_i \in \mathbf{A}_C$, he selects $t_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = g^{t_i}$. Otherwise, he selects $t_i \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = h^{t_i} = g^{t_i(a+\eta)}$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$, while the implicit master secret key is $(ab, \{t_i\}_{a_i \in \mathbf{A}_c}, \{t_i(a+\eta)\}_{a_i \notin \mathbf{A}_c})$. $\mathcal{B}$ responds $\mathcal{A}$ with $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$.

Phase 1. For a secret key query on a set of attributes $\mathbf{A}$, the only restriction is $\mathbf{A}_C \nsubseteq \mathbf{A}$. Suppose that $\mathbf{A} \bigcap \mathbf{A}_C = \{a_{i_1}, a_{i_2}, \cdots, a_{i_l}\}$, where $0 \leq l < k$. $\mathcal{B}$

selects $r, y_{i_1}, y_{i_2}, \cdots, y_{i_l}, \cdots, y_{i_{k-1}} \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$D = B^{-\eta} h^r \tag{6.1}$$

$$\{D_{i_j} = h^{\frac{y_{i_j}}{t_{i_j}}}\}_{a_{i_j} \in \mathbf{A}_C} \tag{6.2}$$

and

$$\{D_{i_j} = (B^{-1} g^r)^{\frac{\Delta_{\mathbf{S},0}(a_{i_j})}{t_{i_j}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j} \Delta_{\mathbf{S},a_{i_j}}(a_{i_j})}{t_{i_j}}}\}_{a_{i_j} \in \mathbf{A} - \mathbf{A}_C} \tag{6.3}$$

where $\mathbf{S} = \{a_{i_1}, a_{i_2}, \cdots, a_{i_l}, \cdots, a_{i_{k-1}}\} \bigcup \{0\}$.

We claim that D and D_{i_j} are correctly generated.

$$\begin{aligned} D &= B^{-\eta} h^r \\ &= g^{-b\eta} g^{r(a+\eta)} \\ &= g^{ab} g^{-ab-b\eta} g^{r(a+\eta)} \\ &= g^{ab} g^{-b(a+\eta)} g^{r(a+\eta)} \\ &= g^{ab} g^{(a+\eta)(r-b)} \\ &= g^{ab} h^{r-b}. \end{aligned}$$

Let $r' = r - b$, we have $D = g^{ab} h^{r'}$. By the choices of $y_{i_1}, y_{i_2}, \cdots, y_{i_{k-1}}$, we implicitly defined a $(k-1)$ degree polynomial $p(x) \in \mathbb{Z}_p[x]$ with $p(0) = r'$ and $p(a_{i_j}) = y_{i_j}$ for $a_{i_j} \in \mathbf{A} \bigcap \mathbf{A}_C$. Hnece, we can reconstruct $p(x)$ using Lagrange interpolation as follows:

$$p(x) = (r-b)\Delta_{\mathbf{S},0}(x) + \sum_{j=1}^{k-1} y_{i_j} \Delta_{\mathbf{S},a_{i_j}}(x).$$

Therefore, for $a_{i_j} \in \mathbf{A} - \mathbf{A}_C$, we have

$$\begin{aligned} D_{i_j} &= h^{\frac{p(a_{i_j})}{t_{i_j}(a+\eta)}} = g^{\frac{p(a_{i_j})}{t_{i_j}}} \\ &= g^{\frac{(r-b)\Delta_{\mathbf{S},0}(a_{i_j})}{t_{i_j}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j} \Delta_{\mathbf{S},a_{i_j}}(a_{i_j})}{t_{i_j}}} \\ &= (B^{-1} g^r)^{\frac{\Delta_{\mathbf{S},0}(a_{i_j})}{t_{i_j}}} \prod_{j=1}^{k-1} g^{\frac{y_{i_j} \Delta_{\mathbf{S},a_{i_j}}(a_{i_j})}{t_{i_j}}}. \end{aligned}$$

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$ and obtains one bit $\omega \in \{0,1\}$. $\mathcal{B}$ computes

$$C_0 = M_\omega \cdot Z,\ C_1 = C,\ \text{and}\ \{C_x = C^{t_x}\}_{a_x \in \mathbf{A}_C}.$$

$\mathcal{B}$ responds $\mathcal{A}$ with the challenged ciphertext $CT^* = (C_0, C_1, \{C_x\}_{a_x \in \mathbf{A}_C})$. Hence, $(C_0, C_1, \{C_x\}_{a_x \in \mathbf{A}_C})$ is a valid ciphertext of M_ω whenever $Z = e(g,g)^{abc}$.

Phase 2. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs his guess $\mu' = 0$; otherwise $\mathcal{B}$ outputs his guess $\mu' = 1$.

The public parameters and secret keys generated in the simulation paradigm are identical to those in the real protocol. Hence, the advantage with which $\mathcal{B}$ can use $\mathcal{A}$ to break the DBDH assumption can be computed as follows.

If $\mu = 0$, $(C_0, C_1, \{C_x\}_{a_x \in \mathbf{A}_C})$ is a valid ciphertext of M_ω. Therefore, $\mathcal{A}$ can output $\omega' = \omega$ with advantage at least $\epsilon(\ell)$, namely $\Pr[\omega' = \omega | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ outputs $\mu' = 0$ when $\omega' = \omega$, we have $\Pr[\mu' = \mu | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$.

If $\mu = 1$, $\mathcal{A}$ cannot obtain any information about ω. In other words, $\mathcal{A}$ can output $\omega' \neq \omega$ with no advantage, namely $\Pr[\omega' \neq \omega | \mu = 1] = \frac{1}{2}$. Since $\mathcal{B}$ outputs $\mu = 1$ when $\omega' \neq \omega$, we have $\Pr[\mu' = \mu | \mu = 1] = \frac{1}{2}$.

Therefore, the advantage with which $\mathcal{B}$ can break the DBDH assumption is $\left|\frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2}\right| \geq \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \epsilon(\ell) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \geq \frac{1}{2}\epsilon(\ell)$. □

Theorem 6.2 *Our attribute-based data transfer with filtering scheme is $(T, q, \epsilon(\ell))$ secure in the filtration security model if the $(T', \epsilon'(\ell))$ computational bilinear Diffie-Hellman assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$ and the hash function $\mathcal{H}$ is collusion resistant, where*

$$T' = T + \Theta(T) \quad \textit{and} \quad \epsilon(\ell) = \epsilon'(\ell).$$

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the filtration security in our scheme, we can construct an algorithm $\mathcal{B}$ that can uses $\mathcal{A}$ to break the CBDH assumption as follows.

The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$ and a hash function $\mathcal{H} : \mathbb{G}_\tau \times \mathbb{G}_\tau \rightarrow \{0,1\}^\lambda$. Let g be a generator of the group $\mathbb{G}$. $\mathcal{C}$ sends $(A, B, C) = (g^a, g^b, g^c)$ to $\mathcal{B}$. $\mathcal{B}$ will outputs $Z = e(g,g)^{abc}$.

Initialization. $\mathcal{A}$ submits a set of attributes $\mathbf{R}^*$ with which he wants to be challenged.

Setup. $\mathcal{B}$ sets $Y = e(g,g)^{ab}$ and $h = Ag^{\eta}$, where $\eta \overset{R}{\leftarrow} \mathbb{Z}_p$. If $a_i \in R^*$, he selects $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = g^{t_i}$. Otherwise, he selects $t_i \overset{R}{\leftarrow} \mathbb{Z}_p$ and computes $T_i = h^{t_i} = g^{t_i(a+\eta)}$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$, while the master secret key is $(ab, \{t_i\}_{a_i \in R^*}, \{t_i(a+\eta)\}_{a_i \notin R^*})$. $\mathcal{B}$ responds $\mathcal{A}$ with $(e, p, \mathbb{G}, \mathbb{G}_\tau, Y, T_1, T_2, \cdots, T_n)$.

Phase 1. For an authentication information query on an access structure $\mathbb{A}_{\mathcal{R}} = (W, \{a_{i_j}, E_{i_j}\}_{j=1}^{\rho})$ where $|\mathbf{R}^* \bigcap \mathbf{Q}| < k$ and $\mathbf{Q} = \{a_{i_1}, a_{i_2}, \cdots, a_{i_\rho}\}$, $\mathcal{B}$ can create a secret key $(D_S, \{D_{S,v_j}\}_{a_{v_j} \in \mathbf{Q}'})$ using the techniques in (6.1), (6.2) and (6.3), where $\mathbf{Q}' \subseteq \mathbf{Q}$ and $|\mathbf{Q}'| = k$. $\mathcal{B}$ computes $F_S = e(D_S, W)$, $\{F_{v_j} = e(D_{S,v_j}, E_{v_j})\}_{a_{v_j} \in \mathbf{Q}'}$ and $F = \frac{F_S}{\prod_{a_{v_j} \in \mathbf{Q}'} F_{v_j}}$. Then, $\mathcal{B}$ selects $s \overset{R}{\leftarrow} \mathbb{Z}_p$, $M \overset{R}{\leftarrow} \mathbb{G}_\tau$ and a set of attributes $\mathbf{A}_C = \{a_{j_1}, a_{j_2}, \cdots, a_{j_k}\} \subseteq \mathbb{U}$, and computes $CT = (C_0, C_1, \{C_{j_x}\}_{x=1}^{k})$, where $C_0 = e(g,g)^{\alpha s}$, $C_1 = g^s$ and $\{C_{j_x} = T_{j_x}^s\}_{a_{j_x} \in \mathbf{A}_C}$. $\mathcal{B}$ computes $\Gamma = \mathcal{H}(F||CT)$ and responds $\mathcal{A}$ with the authentication information $AI = \Gamma$ and the ciphertext CT.

Challenge. $\mathcal{B}$ sets $W^* = C$, and computes $\{E_{j_c}^* = C^{t_{j_c}}\}_{a_{j_c} \in \mathbf{R}^*}$. $\mathcal{B}$ responds $\mathcal{A}$ with the challenged access structure $\mathbb{A}_{\mathcal{R}^*} = (W^*, \{E_{j_c}^*\}_{a_{j_c} \in \mathbf{R}^*})$.

Phase 2. Phase 1 is repeated.

Outputs. $\mathcal{A}$ outputs an authentication information $AI = \Gamma^*$.

As shown above, the public parameters and the authentication information are identical to those in the real protocol. Therefore, we can compute the advantage with which $\mathcal{B}$ can use $\mathcal{A}$ to break the CBDH assumption as follows.

When $W = C = g^c$, the authentication key is $K^* = e(g,g)^{abc}$. So, if $\mathcal{A}$ can output a valid authentication information $AI = \Gamma^*$ for the access structure $\mathbb{A}_{\mathcal{R}^*}$ with the advantage at least $\epsilon(\ell)$, he can computes $F^* = e(g,g)^{abc}$ with the same advantage as $\mathcal{H}$ is a one-way hash function. Therefore, the advantage with which $\mathcal{B}$ can break the CBDH assumption is at least $\epsilon(\ell)$.

□

In our ABDTF scheme, we can implement a (k,n)-threshold access structure. In order to express a complex access structure, we can use the *access tree* technique introduced in Section 5.3.1.

6.4 Chapter Summary

The confidentiality of the sensitive data and the defence of DoS attacks attract lots of attention in data transfer research community. Although data transfer scheme have been extensively studied, there is no scheme where how to transfer and filter data according the required attributes is discussed.

We proposed the formal definition and security model for ABDTF schemes, which provides a formal treatment for the research of ABDTF schemes. Subsequently, we constructed an ABDTF scheme and proved its security in the proposed security model. In our scheme, both the authentication key and the authentication information are short, and can be computed off-line by the receiver and the sender, respectively. Notably, the authentication key can be updated dynamically without re-initializing the system and re-issuing secret keys to users.

Table 6.1: The Comparison of Computation cost

Scheme	Computation Cost				
	Setup	KeyGen	Receiver Policy	Encryption	Decryption
SW[SW05]	$(n+1)T_e + T_p$	$\|\mathbf{A}_\mathcal{U}\|T_e$	$--$	$(\|\mathbf{A}_C\|+1)T_e$	$kT_e + 2kT_p$
GPSW[GPSW06]	$(n+1)T_e + T_p$	$\|\mathbf{A}_\mathcal{U}\|T_e$	$--$	$(\|\mathbf{A}_C\|+1)T_e$	$k(T_e + T_p)$
PTMW[PTMW06]	T_e	$3\|\mathbf{A}_\mathcal{U}\|T_e$	$--$	$(\|\mathbf{A}_C + 2\|)T_e$	$2kT_e + (k+1)T_p$
Our Scheme	$(n+2)T_e + T_p$	$(\|\mathbf{A}_\mathcal{U}\|+1)T_e$	$(\rho+2)T_e$	$(\|\mathbf{A}_C\| + k + 2)T_e + T_p$	$kT_e + (k+1)T_p$

Table 6.2: The Comparison of Communication cost

Scheme	Communication Cost			
	Setup	KeyGen	Receiver Policy	Encryption
SW[SW05]	$nE_\mathbb{G} + E_{\mathbb{G}_\tau}$	$\|\mathbf{A}_\mathcal{U}\|E_\mathbb{G}$	$--$	$\|\mathbf{A}_C\|E_\mathbb{G} + E_{\mathbb{G}_\tau}$
GPSW[GPSW06]	$nE_\mathbb{G} + E_{\mathbb{G}_\tau}$	$\|\mathbf{A}_\mathcal{U}\|E_\mathbb{G}$	$--$	$\|\mathbf{A}_C\|E_\mathbb{G} + E_{\mathbb{G}_\tau}$
PTMW[PTMW06]	$(n+1)E_\mathbb{G} + E_{\mathbb{G}_\tau}$	$2\|\mathbf{A}_\mathcal{U}\|E_\mathbb{G}$	$--$	$(\|\mathbf{A}_C\|+1)E_\mathbb{G} + E_{\mathbb{G}_\tau}$
Our Scheme	$nE_\mathbb{G} + E_{\mathbb{G}_\tau}$	$(\|\mathbf{A}_\mathcal{U}\|+1)E_\mathbb{G}$	$(\rho+1)E_\mathbb{G} + E_{\mathbb{G}_\tau}$	$(\|\mathbf{A}_C\|+1)E_\mathbb{G} + 2E_{\mathbb{G}_\tau}$

Chapter 7

A Generic Construction of Dynamic Single Sign-on

In this chapter, we propose the formal definitions and security models for single sign-on (SSO) and dynamic single sign-on (DSSO). Furthermore, we give a generic construction of DSSO and prove its security. Parts of this work appeared in [HMSY10].

7.1 Introduction

With the increasing use of personalized/protected services, users need to create and maintain more and more usernames and corresponding passwords so that they can access the entitled services. However, this imposes a burden on users. SSO provides a good remedy to this problem as it allows a user to access multiple services using only one password. A traditional SSO system comprises three entities: an identity provider, a group of requesters and a group of service providers. The identity provider manages requesters' personally identifiable information (PII), authenticates them and issues credentials to them. Service providers provide services to the requesters authorized by the identity provider. SSO is a scheme where a requester authenticates himself to the identity provider and can access the services managed by the designated service providers without the need for further authentication [PM03b]. Hence, SSO can shift the great administrative burden of the numerous requesters' profiles from service providers to the identity provider. Meanwhile, SSO plays a core role in the federated identity management (FIdM) where the exchange of users' identity-related information can be optimized [CP07].

In practice, users change their services frequently, so an SSO scheme where the service change can be provided is desirable. DSSO is a flexible SSO scheme where a requester can change his/her choices dynamically without initializing the system and re-issuing credentials to other users.

Unfortunately, there are some limitations in the existing SSO schemes. For example, they are subject to the single point of failure [KR00]. The main reason for this is that the identity provider must always be online, otherwise requesters cannot be granted services from service providers. Furthermore, a credential could be used by an illegitimate user to obtain the services which should not be accessed by him. Finally, these schemes are fragile to the impersonation attacks, namely attackers can impersonate a requester and log in his account when his password is compromised. This can be mainly ascribed to the missing of individual participation principle [OEC80] and the user control and consent principle [Cam05]. All these flaws stem from the lack of active/dynamic control over the process by the requester, after he has entered the correct password.

7.1.1 Related Work

Microsoft .NET Passport is one of the most widely deployed SSO systems, where a passport server works as the identity provider [Opp03]. It uses cookies to store and convey user's PII. When accessing to a service provider, the requester is redirected to the passport server for authentication. After authentication, the passport server creates three cookies: ticket cookie, profile cookie and visited sites cookie. The ticket cookie consist of the unique identifier and a timestamp. The profile cookie comprises the requester's personal information. The visited sites cookie contains the list of the sites which the requester can access. All cookies created by the passport server are encrypted using the triple DES encryption algorithm under the key shared between the passport server and all service providers. Then, the passport server sends these cookies to the requester. In order to access a service, the requester submits these cookies to the service provider. The service provider decrypts them and obtains the authentication information of the requester. .Net Passport incurs some attacks, such as single point of failure, key management failure, misuse of cookies, *etc.* [KR00, Opp03].

The Liberty Alliance project was launched in 2001 [All01]. This project aimed to create an open, federated and SSO solution for the digital economy via any device connected to the Internet. Being different from the .Net Passport, the Liberty Alliance project uses HTTP redirects and URL encodings to transfer requesters' information. In this project, an SSO Service (SSOS) provides requesters an identity

web services framework (ID-WSF)-based method [All06] to obtain liberty authentication assertions which enable them to interact with service providers. In this project, a requester shows his credentials to the service providers without proving the ownership. Hence, it cannot prevent credentials sharing, namely a requester can share his credentials with other illegal users.

OpenID worked as an open and decentralized standard for authenticating requesters [Ope05]. In OpenId, a requester can access different services with the same digital identity and all service providers trust the identity providers. OpenID solves the problem that the digital identities of requesters must be confirmed by the central identity provider. There are more than one identity provider in the OpenID system and users can get their OpenID from any one of them. OpenID is an effective primitive for cross company authentication as well as for SSO. There are two major operation modes in OpenID: Dumb mode and Smart mode [Reh08]. In the first mode, a service provider needs to compare the authentication assertion received from the requester with the initial one stored in the identity provider to prevent the malicious attackers. While in the second mode, an identity provider encrypts the authentication assertions under the key shared between the service provider and him. Therefore, in both of these modes, the identity provider must always be online to enable the authentications of requesters.

Pashalidis and Mitchell [PM03b] revisited SSO systems and divided them into four types: local pseudo-SSO systems, proxy-based pseudo-SSO systems, local true SSO systems and proxy-based true SSO systems. They designed some SSO systems based on trusted platforms [PM03a], GSM/UMTS [PM03c] and EMV card [PM06], respectively.

In order to resolve the single point of failure, two distributed SSO systems were proposed: Cornell SSO (CorSSO) [JSS04] and Threshold Passport (ThresPassport) [CZLC05]. In these systems, an authentication key is split into n different shares and each share is sent to an authentication server. A user can get services from a service provider if he is authenticated by at least t servers.

Suriadi, Foo and Jϕsang [SFJ09] proposed a user-centric federated SSO system (UFed SSO) based on private credential mechanisms, where a requester can minimize the release of his PII.

Although the systems mentioned above have lots of merits, they were not formally proven.

Bhargav-Spantzel, Camenisch, Gross and Sommer [BSCGS06] proposed a taxonomy and raised some open issues on user centric FIdM systems. They classified the existing systems into two predominant types: credential-focused systems and relation-focused systems. In credential-focused systems, the identity providers may be off-line and issue long-term credentials. While in relationship-focused systems, users need to maintain the relationship with the online identity providers who create short-term credentials for them during the transactions. They claimed an universal user centric FIM which should have long-term as well as short-term credentials, online and off-line identity providers.

7.1.2 Our Contribution

We formalize the definitions and the security models for SSO and DSSO. It is the *first* time that the formal definitions and security models for SSO and DSSO are formally defined. We give a generic construction of DSSO systems based on three building blocks: (1) broadcast encryption; (2) digital signature and (3) zero-knowledge proof. We provide a formal security proof for our generic construction.

7.1.3 Chapter Organization

The remainder of this chapter is organized as follows. In Section 7.2, the formal definitions and security models for SSO and DSSO are proposed. We give a generic construction of DSSO in Section 7.3. The formal proof of our generic construction is provided in Section 7.4. Finally, Section 7.5 summarizes this chapter.

7.2 Formal Definitions and Security Models

In this section, we propose formal definitions and security models for both SSO and DSSO.

7.2.1 Single Sign-on

In an SSO scheme, there are three entities: identity provider $\mathcal{IP}$, requester $\mathcal{R}$ and service provider $\mathcal{SP}$. $\mathcal{R}$ authenticates himself to the $\mathcal{IP}$ once and can access multiple $\mathcal{SP}$s. In order to protect $\mathcal{R}$'s PII, an ideal SSO scheme should satisfy the basic requirement that only the designated $\mathcal{SP}$s can check $\mathcal{R}$'s PII. Suppose that there

are N service providers $\mathcal{SP}_1, \mathcal{SP}_2, \cdots, \mathcal{SP}_N$. A circle of trust (CoT) consists of the identity provider and service providers, where each service provider trusts the identity provider.

An SSO scheme consists of the following algorithms.

$\mathsf{Setup}(1^\ell) \rightarrow (params, (ISK, IPK))$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$ and a secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (ISK, IPK)$ for $\mathcal{IP}$.

$\mathsf{Reg}(params, RI) \rightarrow (ID_{\mathcal{SP}}, SK_{\mathcal{SP}_i})/(ID_{\mathcal{R}}, \mathbb{A}_{\mathcal{R}})$. The registration algorithm takes as input the public parameters $params$, the registration information $RI_{\mathcal{SP}_i}$ of $\mathcal{SP}_i$ (or $RI_{\mathcal{R}}$ of $\mathcal{R}$), and outputs $(ID_{\mathcal{SP}_i}, SK_{\mathcal{SP}_i})$ (or $(ID_{\mathcal{R}}, \mathbb{A}_{\mathcal{R}})$), where $ID_{\mathcal{SP}_i}$ and $ID_{\mathcal{R}}$ are the identifiers of $\mathcal{SP}_i$ and $\mathcal{R}$ in the circle of trust, respectively. $SK_{\mathcal{SP}_i}$ is $\mathcal{SP}_i$'s verification key and $\mathbb{A}_{\mathcal{R}}$ is $\mathcal{R}$'s access right which is a set consisting of the identifiers of the $\mathcal{SP}$s that $\mathcal{R}$ can access. $\mathcal{R}$ generates a secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (SK_{\mathcal{R}}, PK_{\mathcal{R}})$.

$\mathsf{CreGen}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA_{\mathcal{R}}, T_{\mathcal{R}}, ISK) \rightarrow Cred_{\mathcal{R}}$. The credential generation algorithm takes as input the public parameters $params$, the identifier $ID_{\mathcal{R}}$, the public key $PK_{\mathcal{R}}$, an authentication assertion $AA_{\mathcal{R}}$, a timestap $T_{\mathcal{R}}$ and the secret key ISK, and outputs a credential $Cred_{\mathcal{R}}$ for $\mathcal{R}$.

$\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA_{\mathcal{R}}, T_{\mathcal{R}}, Cred_{\mathcal{R}}, SK_{\mathcal{SP}_i}) \rightarrow Ture/False$. The credential verification algorithm takes as input the public parameters $params$, the public key IPK, the identifier $ID_{\mathcal{R}}$, the public key $PK_{\mathcal{R}}$, the authentication assertion $AA_{\mathcal{R}}$, the timestamp $T_{\mathcal{R}}$, the credential $Cred_{\mathcal{R}}$ and the verification key $SK_{\mathcal{SP}_i}$, and outputs $Ture$ if and only if $ID_{\mathcal{SP}_i} \in \mathcal{A}_{\mathcal{R}}$ and $Cred_{\mathcal{R}}$ is correctly generated by $\mathcal{IP}$; otherwise it outputs $False$.

$\mathsf{OProof}(\mathcal{R}(params, SK_{\mathcal{R}}) \leftrightarrow \mathcal{SP}_i(params, PK_{\mathcal{R}}, Resp)) \rightarrow (Resp, Accpt/Reject)$. The ownership proof algorithm is an interactive algorithm executed between $\mathcal{R}$ and $\mathcal{SP}_i$. $\mathcal{R}$ takes as input the public parameters $params$ and the secret key $SK_{\mathcal{R}}$, and outputs a response $Resp$ for the zero-knowledge proof that $SK_{\mathcal{R}}$ is the secret key corresponding to $PK_{\mathcal{R}}$. $\mathcal{SP}_i$ takes as input the public parameters $params$, the public key $PK_{\mathcal{R}}$ and the response $Resp$, and outputs $Accept$ if and only if the zero-knowledge proof is correct; otherwise it outputs $Reject$.

Definition 7.1 *We say that a single sign-on scheme is correct if*

$$\Pr\left[\begin{array}{l|l} & \mathsf{Setup}(1^{\ell}) \rightarrow (params, (ISK, IPK)); \\ & \mathsf{Reg}(params, RI) \rightarrow (ID_{\mathcal{SP}}, SK_{\mathcal{SP}_i})/ \\ & (ID_{\mathcal{R}}, \mathbb{A}_{\mathcal{R}}); \\ \mathsf{OProof}(\mathcal{R}(\boxtimes) \leftrightarrow \mathcal{SP}_i(\boxplus)) & \mathsf{CreGen}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA_{\mathcal{R}}, T_{\mathcal{R}}, \\ \rightarrow (Resp, Accpt) & ISK) \rightarrow Cred_{\mathcal{R}}; \\ & \mathsf{CreVer}(params, IPK, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA_{\mathcal{R}}, \\ & T_{\mathcal{R}}, Cre_{cR}, SK_{\mathcal{SP}_i}) \rightarrow Ture; \\ & ID_{\mathcal{SP}_i} \in \mathbb{A}_{\mathcal{R}} \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by the algorithms in the scheme, $\boxtimes = (params, SK_{\mathcal{R}})$ *and* $\boxdot = (params, PK_{\mathcal{R}}, Resp)$.

7.2.2 Security Model of Single Sign-on

SSO is a complicated system where multiple parties can co-exist. Considering the security of SSO schemes, three basic types of attacks which are based on relevant combinations of compromised parties should be addressed: *collusion credential forging attacks*, *collusion impersonation attacks* and *coalition credential forging attacks*. In the collusion credential forging attacks, malicious requesters can cooperatively forge a credential for the target requester. They can impersonate the target requester to get services from the service providers whose identifiers are included in the access right of the target requester. In the collusion impersonation attacks, since the malicious service providers have checked the credentials of a requester, they obtained the corresponding proof information on the requester. Hence, malicious service providers can cooperatively mimic the owner of the credentials. In the coalition credential forging attacks, malicious requesters and service providers can cooperatively forge a credential for the target requester, in which the identifiers of the malicious service providers are not listed.

The security model of SSO is formalized by the following games executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Game 1: Collusion Credential Forging Attacks.

Initialization. $\mathcal{A}$ submits a target requester $\mathcal{R}^*$ for whom it wants to forge a credential.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameter $params$ and a secret-public key pair (ISK, IPK). $\mathcal{C}$ responds $\mathcal{A}$ with $(params, IPK)$.

Registration Queries. $\mathcal{A}$ can adaptively query the registration oracle. $\mathcal{A}$ submits $\{RI_{\mathcal{R}_1}, RI_{\mathcal{R}_2}, \cdots, RI_{\mathcal{R}_{q_1}}\}$, where the only constraint is $RI_{\mathcal{R}_i} \neq RI_{\mathcal{R}^*}$. $\mathcal{C}$ runs $\mathsf{Reg}(params, RI_{\mathcal{R}_i})$ and responds $\mathcal{A}$ with $(ID_{\mathcal{R}_i}, \mathbb{A}_i)$ for $i = 1, 2, \cdots, q_1$.

Credential Generation Queries. $\mathcal{A}$ can adaptively query the credential generation oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_2}}, PK_{\mathcal{R}_{q_2}})\}$, where $(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}) \neq (ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*})$, $PK_{\mathcal{R}_i}$ is the public key of $\mathcal{R}_i$ and PK^* is the public key of $\mathcal{R}^*$. $\mathcal{C}$ runs $\mathsf{CreGen}(params, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_i, T_i, ISK)$ and responds $\mathcal{A}$ with $Cred_{\mathcal{R}_i}$ for $i = 1, 2, \cdots, q_2$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}, Cred_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}, Cred_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_3}}, PK_{\mathcal{R}_{q_3}}, Cred_{\mathcal{R}_{q_3}})\}$. $\mathcal{C}$ runs $\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_{\mathcal{R}_i}, T_{\mathcal{R}_i}, Cred_{\mathcal{R}_i}, SK_{\mathcal{SP}_j})$, and responds $\mathcal{A}$ with $True/False$, for $i = 1, 2, \cdots, q_3$.

Output. $\mathcal{A}$ outputs a credential $Cred_{\mathcal{R}^*}$ for $\mathcal{R}^*$. $\mathcal{A}$ wins the game if

$$\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}, SK_{\mathcal{SP}_j}) \rightarrow True.$$

Definition 7.2 *We say that a single Sign-on system is $(t, q_1, q_2, q_3, \epsilon(\ell))$-secure against the collusion credential forging attacks if no PPT adversary $\mathcal{A}$ making at most q_1 registration queries, q_2 credential generation queries and q_3 credential verification queries can win the game with the advantage*

$$\Pr\left[\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}, SK_{\mathcal{SP}_j}) \rightarrow True\right] \geq \epsilon(\ell)$$

in the above model.

Game 2. Collusion Impersonate Attacks.

Initialization. $\mathcal{A}$ submits a target requester $\mathcal{R}^*$ whom it wants to impersonate. $\mathcal{A}$ works as malicious service providers.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{C}$ responds $\mathcal{A}$ with $(params, IPK)$.

Ownership Proof Queries. $\mathcal{A}$ can adaptively query the ownership proof oracle. $\mathcal{A}$ submits $\{PK_{\mathcal{R}_1}, PK_{\mathcal{R}_2}, \cdots, PK_{\mathcal{R}_q}\}$, where $PK_{\mathcal{R}_i} \neq PK_{\mathcal{R}^*}$ for $i = 1, 2, \cdots, q$. $\mathcal{C}$ runs $\mathsf{OProof}(\mathcal{R}(params, SK_{\mathcal{R}_i}) \leftrightarrow \mathcal{SP}_j(params, PK_{\mathcal{R}_i}, \cdot))$, and responds $\mathcal{A}$ with $Resp$, for $i = 1, 2, \cdots, q$.

Output. $\mathcal{A}$ outputs a response $Resp^*$ for the challenged $PK_{\mathcal{R}^*}$ from $\mathcal{C}$. $\mathcal{A}$ wins the game if

$$\mathsf{OProof}(\mathcal{A}(params, \cdot) \leftrightarrow \mathcal{SP}_i(params, PK_{\mathcal{R}^*}, Resp^*)) \rightarrow (Resp^*, Accept).$$

Definition 7.3 *We say that a single sign-on system is $(t, q, \epsilon(\ell))$-secure against the collusion impersonation attacks if no PPT adversary $\mathcal{A}$ making at most q ownership proof queries can win the game with the advantage*

$$\Pr\left[\mathsf{OProof}(\mathcal{A}(params, \cdot) \leftrightarrow \mathcal{SP}_i(params, PK_{\mathcal{R}^*}, Resp^*)) \rightarrow (Resp^*, Accept)\right] \geq \epsilon(\ell)$$

in the above model.

Game 3: Coalition Credential Forging Attacks.

Initialization. $\mathcal{A}$ submits a target requester $\mathcal{R}^*$ whom it wants to impersonate and a target service provider $\mathcal{SP}^*$ whom it wants to attack. $\mathcal{A}$ works as malicious requesters and service providers.

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{C}$ responds $\mathcal{A}$ with $(params, IPK)$.

Registration Queries. $\mathcal{A}$ can adaptively query the registration oracle. $\mathcal{A}$ submits $\{RI_{\mathcal{R}_1}, RI_{\mathcal{R}_2}, \cdots, RI_{\mathcal{R}_{t_1}}\}$ with $RI_{\mathcal{R}_i} \neq RI_{\mathcal{R}^*}$ and $\{RI_{\mathcal{SP}_1}, RI_{\mathcal{SP}_2}, \cdots, RI_{\mathcal{SP}_{t_2}}\}$ with $RI_{\mathcal{SP}_j} \neq RI_{\mathcal{SP}^*}$ for $i = 1, 2, \cdots, t_1$ and $j = 1, 2, \cdots, t_2$. Let $q_1 = t_1 + t_2$. $\mathcal{C}$ runs $\mathsf{Reg}(params, RI_{\mathcal{R}_i})$ and $\mathsf{Reg}(params, RI_{\mathcal{SP}_j})$, and responds $\mathcal{A}$ with $(ID_{\mathcal{R}_i}, \mathbb{A}_i)$ and $(ID_{\mathcal{SP}_j}, SK_{\mathcal{SP}_j})$ for $i = 1, 2, \cdots, t_1$ and $j = 1, 2, \cdots, t_2$.

Credential Generation Queries. $\mathcal{A}$ can adaptively query the credential generation oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_2}}, PK_{\mathcal{R}_{q_2}})\}$, where $(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}) \neq (ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*})$, $PK_{\mathcal{R}_i}$ is the public key of $\mathcal{R}_i$ and PK^* is the public key of $\mathcal{R}^*$. $\mathcal{C}$ runs $\mathsf{CreGen}(params, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_i, T_i, ISK)$ and responds $\mathcal{A}$ with $Cred_{\mathcal{R}_i}$ for $i = 1, 2, \cdots, q_2$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}, Cred_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}, Cred_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_3}}, PK_{\mathcal{R}_{q_3}}, Cred_{\mathcal{R}_{q_3}})\}$. $\mathcal{C}$ runs $\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_{\mathcal{R}_i}, T_{\mathcal{R}_i}, Cred_{\mathcal{R}_i}, SK_{\mathcal{SP}_j})$, and responds $\mathcal{A}$ with $True/False$ for $i = 1, 2, \cdots, q_3$.

Output. $\mathcal{A}$ outputs a credential $Cred_{\mathcal{R}^*}$ for $\mathcal{R}^*$, which will be verified by $\mathcal{SP}^*$ with $\mathcal{SP}^* \notin \mathcal{A}$. $\mathcal{A}$ wins the game if

$$\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}, SK_{\mathcal{SP}^*}) \rightarrow True.$$

Definition 7.4 *We say that a single sign-on scheme is $(t, q_1, q_2, q_3, \epsilon(\ell))$-secure against coalition credential forging attacks if no PPT adversary $\mathcal{A}$ making at most q_1 registration queries, q_2 credential generation queries and q_3 credential verification queries can win the game with the advantage*

$$\Pr[\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}, SK_{\mathcal{SP}^*}) \rightarrow True] \geq \epsilon(\ell)$$

in the above model.

7.2.3 Dynamic Single Sign-on

The formal definition of DSSO is as follows.

A DSSO scheme consists of the following seven algorithms: $\mathsf{Setup}(\cdot)$, $\mathsf{Reg}(\cdot)$, $\mathsf{CreGen}(\cdot)$, $\mathsf{CreVer}(\cdot)$, $\mathsf{OProof}(\cdot)$, an addition algorithm $\mathsf{Add}(\cdot)$ and a deletion algorithm $\mathsf{Del}(\cdot)$, where $\mathsf{Setup}(\cdot)$, $\mathsf{Reg}(\cdot)$, $\mathsf{CreGen}(\cdot)$, $\mathsf{CreVer}(\cdot)$ and $\mathsf{OProof}(\cdot)$ are the same as in section 7.2.1. The addition algorithm and the delete algorithm work as follows.

Add $(params, ID_{\mathcal{SP}}) \rightarrow \mathbb{A}'_{\mathcal{R}}$. The addition algorithm takes as input the public parameters $params$ and an identifier $ID_{\mathcal{SP}}$ of a service provider $\mathcal{SP}$, and outputs a new access right $\mathbb{A}'_{\mathcal{R}} = \mathbb{A}_{\mathcal{R}} \bigcup \{ID_{\mathcal{SP}}\}$ for $\mathcal{R}$.

Del $(params, ID_{\mathcal{SP}}) \rightarrow \mathbb{A}'_{\mathcal{R}}$. The delete algorithm takes as input the public parameters $params$ and an identifier $ID_{\mathcal{SP}}$ of a service provider $\mathcal{SP}$, and outputs a new access right $\mathbb{A}'_{\mathcal{R}} = \mathbb{A}_{\mathcal{R}} - \{ID_{SP}\}$ for $\mathcal{R}$.

7.2.4 Security Model of Dynamic Single Sign-on

In multiple parties communication and dynamic schemes, two special attacks should be addressed: *forward security* and *backward security*. Because, in these systems,

participants can join or leave the systems frequently. Since a requester is permitted to add or revoke services dynamically, a secure DSSO system should resist these attacks. By forward security, we mean that the service provider $\mathcal{SP}$ cannot validate the credentials which were issued before his identifier $ID_{\mathcal{SP}}$ is added to $\mathcal{R}$'s access right $\mathbb{A}_{\mathcal{R}}$. While, by backward security, we mean that the service provider $\mathcal{SP}$ cannot validate the credentials which are issued after his identifier $ID_{\mathcal{SP}}$ has been removed from $\mathcal{R}$'s access right $\mathbb{A}_{\mathcal{R}}$. These models are defined by the following games executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Game 4: Forward Security.

Setup. Suppose that $\mathcal{A}$ is a malicious service provider. $\mathcal{C}$ runs $\mathsf{Setup}(1^{\ell})$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{C}$ sends $(params, IPK)$ to $\mathcal{A}$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^1_{\mathcal{R}}), (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^2_{\mathcal{R}}), \cdots, (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^q_{\mathcal{R}})\}$, where $Cred^i_{\mathcal{R}}$ is a credential issued after $ID_{\mathcal{A}}$ is added to $\mathbb{A}_{\mathcal{R}}$, for $i = 1, 2, \cdots, q$. $\mathcal{C}$ runs $\mathsf{CreVer}(params, IPK, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^i_{\mathcal{R}}, T^i_{\mathcal{R}}, Cred^i_{\mathcal{R}}, SK_{\mathcal{SP}'})$ where $\mathcal{A} \neq \mathcal{SP}'$ and $ID_{\mathcal{SP}'} \in \mathbb{A}_{\mathcal{R}}$, and responds $\mathcal{A}$ with $True/False$ for $i = 1, 2, \cdots, q$.

Challenge. $\mathcal{C}$ runs $\mathsf{CreGen}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^O_{\mathcal{R}}, T^O_{\mathcal{R}}, ISK)$ to generate an older credential $Cred^O_{\mathcal{R}}$ which was issued before $ID_{\mathcal{A}}$ is listed in $\mathbb{A}_{\mathcal{R}}$. $\mathcal{C}$ randomly generates $Cred^*$ which has the same distribution with $Cred^O_{\mathcal{R}}$. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ sets $Cred_b = Cred^O_{\mathcal{R}}$ and $Cred_{1-b} = Cred^*$. $\mathcal{C}$ responds $\mathcal{A}$ with $(Cred_b, Cred_{1-b})$.

Output. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 7.5 *We say that a dynamic single sign-on scheme is $(t, q, \epsilon(\ell))$-forward secure if no PPT adversary $\mathcal{A}$ making at most q credential verification queries can win the game with the advantage*

$$Adv^F_{\mathcal{A}-DSSO} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

Game 5: Backward Security.

Setup. Suppose that $\mathcal{A}$ is a malicious service provider. $\mathcal{C}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{C}$ sends $(params, IPK)$ to $\mathcal{A}$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query credential verification oracle. $\mathcal{A}$ submits $\{(ID_\mathcal{R}, PK_\mathcal{R}, Cred^1_\mathcal{R}), (ID_\mathcal{R}, PK_\mathcal{R}, Cred^2_\mathcal{R}), \cdots, (ID_\mathcal{R}, PK_\mathcal{R}, Cred^q_\mathcal{R})\}$, where $Cred^i_\mathcal{R}$ is a credential issued before $ID_\mathcal{A}$ is deleted from $\mathbb{A}_\mathcal{R}$ for $i = 1, 2, \cdots, q$. $\mathcal{C}$ runs $\mathsf{CreVer}(params, IPK, ID_\mathcal{R}, PK_\mathcal{R}, AA^i_\mathcal{R}, T^i_\mathcal{R}, Cred^i_\mathcal{R}, SK_{\mathcal{SP}'})$ where $\mathcal{A} \neq \mathcal{SP}'$ and $ID_{\mathcal{SP}'} \in \mathbb{A}_\mathcal{R}$, and responds $\mathcal{A}$ with $True/False$ for $i = 1, 2, \cdots, q$.

Challenge. $\mathcal{C}$ runs $\mathsf{CreGen}(params, ID_\mathcal{R}, PK_\mathcal{R}, AA^N_\mathcal{R}, T^N_\mathcal{R}, ISK)$ to generate a new credential $Cred^N_\mathcal{R}$ which is issued after $ID_\mathcal{A}$ has been deleted from $\mathbb{A}_\mathcal{R}$. $\mathcal{C}$ randomly generates $Cred^*$ which has the same distribution with $Cred^N_\mathcal{R}$. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. $\mathcal{C}$ sets $Cred_b = Cred^N_\mathcal{R}$ and $Cred_{1-b} = Cred^*$. $\mathcal{C}$ responds $\mathcal{A}$ with $(Cred_b, Cred_{1-b})$.

Output. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 7.6 *We say that a dynamic single sign-on scheme is $(t, q, \epsilon(\ell))$-backward secure if no PPT adversary $\mathcal{A}$ making at most q credential verification queries can win the game with the advantage*

$$Adv^B_{\mathcal{A}-DSSO} = \left| \Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above model.

7.3 Generic Construction for Dynamic Single Sign-on

In this section, we give a generic construction of DSSO based on a broadcast encryption scheme $\mathsf{BroEnc}(\cdot)$, a signature scheme $\mathsf{Sign}(\cdot)$ and a zero knowledge proof scheme $(\mathcal{P} \leftrightarrow \mathcal{V})[\cdot]$.

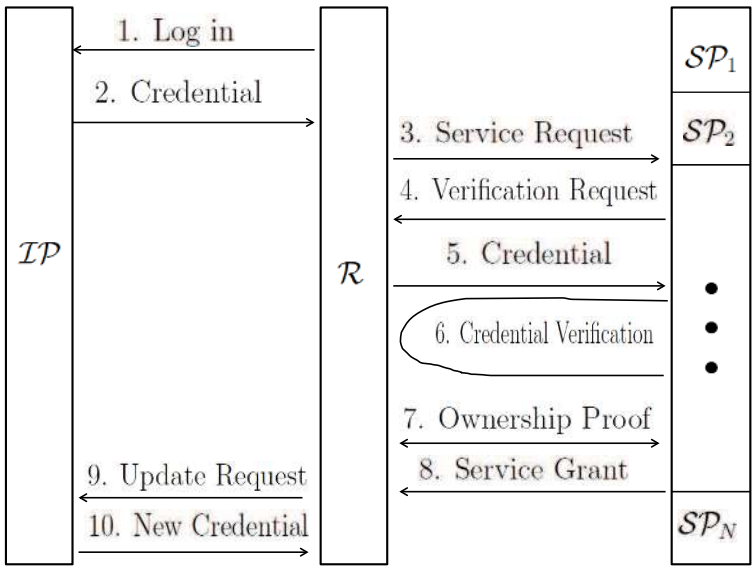

Figure 7.1: The Model of Dynamic Single Sign-on

Overview. In our construction, a requester can change his choices dynamically, while other participants (requester and service provider) in the system do not need to change their credentials or secret keys. When a requester $\mathcal{R}$ logs in, the identity provider ($\mathcal{IP}$) creates a credential for him. Then, the requester can use this credential to access all the services managed by the designated service providers ($\mathcal{SP}$s), instead of submitting different credentials to different $\mathcal{SP}$s. For each logging request, the $\mathcal{IP}$ creates a new credential for $\mathcal{R}$. At this point of time, $\mathcal{R}$ can also be revoked due to the expiry of his membership, for instance. Our construction can prevent illegal credential sharing, which is named as *all-or-nothing non-transferability*. By all-or-nothing non-transferability, we mean that all the credentials of a requester are shared, once he shares one of them with others [CL01]. The model of our construction is explained in Figure 7.1.

We describe our generic construction of DSSO in Figure 7.2.

7.4 Security Analysis

In this section, we prove that our generic construction of DSSO is secure against the collusion credential forging attacks, collusion impersonate attacks and coalition credential forging attacks, and provides forward security and backward security.

Theorem 7.1 *Our generic construction of DSSO is* $(T, q_1, q_2, q_3, \epsilon(\ell))$*-secure against*

System Set-up. This algorithm takes as input 1^ℓ, and outputs the public parameters $params$ which includes all public parameters required in the three underlying building blocks, and a secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (ISK, IPK)$ for the identity provider $\mathcal{IP}$.

Registration.

1. Service Provider Registration. This algorithm takes as input the necessary registration information $RI_{\mathcal{SP}_i}$ of a service provider $\mathcal{SP}_i$, and outputs an identifier $ID_{\mathcal{SP}_i}$ and a verification key $\mathsf{BKeyGen}(params, N) \rightarrow SK_{\mathcal{SP}_i}$, where $\mathsf{BkeyGen}(\cdot)$ is the key generation algorithm in the broadcast encryption scheme. It stores $(\mathcal{SP}_i, ID_{\mathcal{SP}_i}, SK_{\mathcal{SP}_i})$ for $\mathcal{SP}_i$.
2. Requester Registration. A requester $\mathcal{R}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (SK_\mathcal{R}, PK_\mathcal{R})$. This algorithm takes as input the necessary registration information $RI_\mathcal{U}$, and outputs an identifier $ID_\mathcal{U}$ and an access right $\mathbb{A}_\mathcal{R}$ which is a set consisting of the identifiers of the service providers that $\mathcal{R}$ can access. It stores $(ID_\mathcal{R}, PK_\mathcal{R}, \mathbb{A}_\mathcal{R})$ for $\mathcal{R}$. Notably, $\mathbb{A}_\mathcal{R}$ is regarded as the receiver set $\mathbf{S}$ in the broadcast encryption scheme.

Single Sign-on.

1. Log In. $\mathcal{R}$ logs in the system by his username and corresponding password.
2. Credential Generation. $\mathcal{IP}$ runs $\mathsf{BEnc}(params, |\mathbb{A}_\mathcal{R}|) \rightarrow (Hdr_\mathcal{R}, K_\mathcal{R})$, where $\mathsf{BEnc}(\cdot)$ is the encryption algorithm in the broadcast encryption, $Hdr_\mathcal{R}$ is the header and $K_\mathcal{R}$ is the message encryption key which can only be obtained by $\mathcal{SP}_j$ with $ID_{\mathcal{SP}_j} \in \mathbb{A}_\mathcal{R}$.

   $\mathcal{IP}$ generates a signature $\delta_\mathcal{R} = \mathsf{Sign}(params, ID_\mathcal{R}, PK_\mathcal{R}, AA_\mathcal{R}, T_\mathcal{R}, ISK)$, where $ID_\mathcal{R}$ is the identifier of $\mathcal{R}$, $PK_\mathcal{R}$ is the public key of $\mathcal{R}$, $AA_\mathcal{R}$ is an authentication assertion, $T_\mathcal{R}$ is tiemstamp and ISK is the secret key of $\mathcal{IP}$. Then, $\mathcal{IP}$ computes $CT_\mathcal{R} = \mathcal{E}(params, K_\mathcal{R}, \delta_\mathcal{R})$ where $\mathcal{E}(\cdot)$ is a symmetric encryption algorithm. The credential for $\mathcal{R}$ is $Cred_\mathcal{R} = (\mathbb{A}_\mathcal{R}, Hdr_\mathcal{R}, CT_\mathcal{R})$.
3. Service Request. $\mathcal{R}$ sends a service request to $\mathcal{SP}_i$ with $ID_{\mathcal{SP}_i} \in \mathbb{A}_\mathcal{R}$.
4. Verification Request. $\mathcal{R}$ is asked to show his credential to $\mathcal{SP}_i$. $\mathcal{R}$ submits his credential $Cred_\mathcal{R}$ to $\mathcal{SP}_i$
5. Credential Verification. $\mathcal{SP}_i$ runs $\mathsf{BDec}(param, SK_{\mathcal{SP}_i}, Hdr) \rightarrow K_\mathcal{R}$, decrypts $CT_\mathcal{R}$ and obtains $\delta_\mathcal{R}$. $\mathcal{SP}_i$ runs $\mathsf{Ver}(params, ID_\mathcal{R}, PK_\mathcal{R}, AA_\mathcal{R}, T_\mathcal{R}, IPK, \delta_\mathcal{R}) \rightarrow True/False$. If $\mathsf{Ver}(params, ID_\mathcal{R}, PK_\mathcal{R}, AA_\mathcal{R}, T_\mathcal{R}, IPK, \delta_\mathcal{R}) \rightarrow True$, $\mathcal{SP}_i$ goes to the next step; otherwise he aborts.

6. Ownership Proof. $\mathcal{R}$ and $\mathcal{SP}_i$ execute a zero-knowledge proof protocol $(\mathcal{R} \leftrightarrow \mathcal{SP}_i)[SK_{\mathcal{R}}]$ to prove that $\mathcal{R}$ knows the secret key $SK_{\mathcal{R}}$ corresponding the public key $PK_{\mathcal{R}}$ included in $Cred_{\mathcal{R}}$. If the zero-knowledge proof is successful, $\mathcal{SP}_i$ goes to the next step; otherwise he aborts.

7. Service Grant. $\mathcal{SP}_i$ grants the service requested by $\mathcal{R}$ to him.

If $\mathcal{R}$ wants to access other service providers $\mathcal{SP}_j$ with $ID_{\mathcal{SP}_j} \in \mathbb{A}_{\mathcal{R}}$, he can send $Cred_{\mathcal{R}}$ to him directly, without the need to obtain a new credential.

Dynamic Change. If $\mathcal{R}$ needs to change his access right, when logging in, he must submit a request to the $\mathcal{IP}$. After checking the request, $\mathcal{IP}$ creates a new credential for $\mathcal{R}$ according to his current status. $\mathcal{IP}$ works as follows.

1. Addition. $\mathcal{IP}$ adds an identifier $ID_{\mathcal{SP}_c}$ of the service provider $\mathcal{SP}_c$ to $\mathbb{A}_{\mathcal{R}}$ by setting $\mathbb{A}'_{\mathcal{R}} = \mathbb{A}_{\mathcal{R}} \bigcup \{ID_{\mathcal{SP}_j}\}$, and runs $\mathsf{BEnc}(params, |\mathbb{A}_{\mathcal{R}}|') \rightarrow (Hdr'_{\mathcal{R}}, K'_{\mathcal{R}})$.
2. Delete. $\mathcal{IP}$ deletes an identifier $ID_{\mathcal{SP}_c}$ of the service provider $\mathcal{SP}_c$ from $\mathbb{A}_{\mathcal{R}}$ by setting $\mathbb{A}'_{\mathcal{R}} = \mathbb{A}_{\mathcal{R}} - \{ID_{\mathcal{SP}_j}\}$, and runs $\mathsf{BEnc}(params, |\mathbb{A}'_{\mathcal{R}}|) \rightarrow (Hdr'_{\mathcal{R}}, K'_{\mathcal{R}})$.
3. New Credential Generation. $\mathcal{IP}$ generates a signature $\delta'_{\mathcal{R}} = \mathsf{Sign}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA'_{\mathcal{R}}, T'_{\mathcal{R}}, ISK)$, and computes $CT'_{\mathcal{R}} = \mathcal{E}(params, K'_{\mathcal{R}}, \delta'_{\mathcal{R}})$. The new credential for $\mathcal{R}$ is $Cred'_{\mathcal{R}} = (\mathbb{A}'_{\mathcal{R}}, Hdr'_{\mathcal{R}}, CT'_{\mathcal{R}})$

Figure 7.2: A Generic Construction of Dynamic Single Sign-on Schemes

the collusion credential forging attacks if the broadcast encryption scheme is $(T', N, q_2, \epsilon_1(\ell))$*-secure against the chosen ciphertext attacks (or IND-CCA2) and the signature scheme is* $(T'', q_3, \epsilon'(\ell))$*-strongly existentially unforgeable (or SEU-CMA), where* $T'' = T + T' + \Theta(T + T')$ *and* $\epsilon'(\ell) = \epsilon(\ell) \cdot (1 - \epsilon_1(\ell))^{q_3}$.

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q_1, q_2, q_3, \epsilon(\ell))$ break the collusion credential unforgeability of our generic construction of DSSO, we will show that there exists an algorithm $\mathcal{B}$ that can $(T'', q_3, \epsilon'(\ell))$ breaks the strongly existential unforgeability of the signature scheme.

Initialization. $\mathcal{A}$ submits a target requester $\mathcal{R}^*$ for whom $\mathcal{A}$ wants to forge a credential. $\mathcal{A}$ works as malicious requesters.

Setup. $\mathcal{B}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{B}$ responds $\mathcal{A}$ with $(params, IPK)$.

Registration Queries. $\mathcal{A}$ can adaptively query the registration oracle. $\mathcal{A}$ submits $\{RI_{\mathcal{R}_1}, RI_{\mathcal{R}_2}, \cdots, RI_{\mathcal{R}_{q_1}}\}$. $\mathcal{B}$ randomly assigns an identifier $ID_{\mathcal{R}_i}$ and selects a set $\mathbb{A}_{\mathcal{R}_i}$ which consists of the identifiers of service providers, for $i = 1, 2, \cdots, q_1$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{(ID_{\mathcal{R}_i}, \mathbb{A}_{\mathcal{R}_i})\}_{i=1}^{q_1}$ and stores $\{(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, \mathbb{A}_{\mathcal{R}_i})\}_{i=1}^{q_1}$, where $PK_{\mathcal{R}_i}$ is the public key of $\mathcal{R}_i$ for $i = 1, 2, \cdots, q_1$.

Credential Generation Queries. $\mathcal{A}$ can adaptively query the credential generation oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_2}}, PK_{\mathcal{R}_{q_2}})\}$, where the only constraint is $(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}) \neq (ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*})$. $\mathcal{B}$ runs $\mathsf{BEnc}(params, |\mathbb{A}_{\mathcal{R}_i}|) \rightarrow (Hdr_{\mathcal{R}_i}, K_{\mathcal{R}_i})$, $\mathsf{Sign}(params, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_{\mathcal{R}_i}, T_{\mathcal{R}_i}, ISK) \rightarrow \delta_{\mathcal{R}_i}$ and $\mathcal{E}(params, K_{\mathcal{R}_i}, \delta_{\mathcal{R}_i}) \rightarrow CT_{\mathcal{R}_i}$, for $i = 1, 2, \cdots, q_2$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{Cred_{\mathcal{R}_i} = (\mathbb{A}_{\mathcal{R}_i}, Hdr_{\mathcal{R}_i}, CT_{\mathcal{R}_i})\}_{i=1}^{q_2}$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query the credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}, Cred_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}, Cred_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_3}}, PK_{\mathcal{R}_{q_3}}, Cred_{\mathcal{R}_{q_3}})\}$. $\mathcal{B}$ runs $\mathsf{BDec}(params, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}_i}) \rightarrow K_{\mathcal{R}_i}$ where $ID_{\mathcal{SP}_j} \in \mathbb{A}_{\mathcal{R}_i}$, decrypts $CT_{\mathcal{R}_i}$ and obtains $\delta_{\mathcal{R}_i}$. $\mathcal{B}$ runs $\mathsf{Ver}(params, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_{\mathcal{R}_i}, T_{\mathcal{R}_i}, IPK, \delta_{\mathcal{R}_i}) \rightarrow True/False$, for $i = 1, 2, \cdots, q_2$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{(True/False)\}$.

Output. $\mathcal{A}$ outputs a credential $Cred_{R^*} = (\mathbb{A}_{\mathcal{R}^*}, Hdr_{\mathcal{R}^*}, CT_{\mathcal{R}^*})$, where $(ID_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}) \notin \{(ID_{\mathcal{R}_1}, Cred_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, Cred_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_2}}, Cred_{\mathcal{R}_{q_2}})\}$.

$\mathcal{B}$ runs $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*})$ where $ID_{\mathcal{SP}_j} \in \mathbb{A}_{\mathcal{R}^*}$.

1. If $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*}) \rightarrow \perp$, $\mathcal{B}$ aborts. The simulation fails.

2. If $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*}) \rightarrow K_{\mathcal{R}^*}$ and $\mathcal{B}$ can decrypt $CT_{\mathcal{R}^*}$ to obtain $\delta_{\mathcal{R}^*}$, $\mathcal{B}$ aborts. $\mathcal{B}$ can use $\mathcal{A}$ to break IND-CCA2 security of the broadcast encryption scheme. Due to the broadcast encryption is $(T', N, q_2, \epsilon_1(\ell))$-secure against the chose ciphertext attacks, the advantage that $\delta_{\mathcal{R}^*}$ can be obtained is at least $\epsilon_1(\ell)$.

3. If $\mathcal{B}$ does not abort, he can obtain a valid signature $\delta_{\mathcal{R}_*}$ on $(params, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*})$ with the advantage at least $\epsilon(\ell)$.

Now we compute the advantage with which $\mathcal{B}$ does not abort. If the broadcast encryption scheme is $(T', N, q_2, \epsilon_1(\ell))$-secure against chosen ciphertext attacks, $\mathcal{B}$

can abort with the probability at most $\epsilon_1(\ell)$. Therefore, the advantage with which $\mathcal{B}$ does not abort during the q_3 credential verification queries is at least $(1-\epsilon_1(\ell))^{q_3}$. Thus, the advantage with which $\mathcal{B}$ can break the strongly existential unforgeability of the signature scheme is at least $\epsilon(\ell)\cdot(1-\epsilon_1(\ell))^{q_3}$. This contradicts the assumption that the signature scheme is $(T'', q_3, \epsilon'(\ell))$-strongly existentially unforgeable.

□

Theorem 7.2 *Our generic construction of DSSO is secure against collusion impersonation attacks if the zero-knowledge proof scheme is secure.*

Proof: If there exists a PPT adversary $\mathcal{A}$ to whom a requester $\mathcal{R}$ has showed credentials and proved the ownership of them can impersonate $\mathcal{R}$ to prove the ownership on the received credentials, we can construct an algorithm $\mathcal{B}$ (knowledge extractor) that can use $\mathcal{A}$ to break the security of the underlying zero-knowledge proof scheme.

After receiving a response *Resp* for the proof that $SK_{\mathcal{R}}$ is the corresponding secret key of the public key $PK_{\mathcal{R}}$ listed in the credential $Cred_{\mathcal{R}}$, if $\mathcal{A}$ can impersonate $\mathcal{R}$ to prove the ownership on $Cred_{\mathcal{R}}$, he can execute the zero-knowledge proof scheme with a service provider $\mathcal{SP}$ to prove that he knows $SK_{\mathcal{R}}$. If $\mathcal{A}$ can output a new response *Resp*$'$ which correctly answers the challenge from $\mathcal{SP}$, $\mathcal{B}$ (knowledge extractor) aborts. $\mathcal{B}$ can compute $SK_{\mathcal{R}}$ from the two responses *Resp* and *Resp*$'$ using the rewinding technique. So, $\mathcal{B}$ can use $\mathcal{A}$ to break the zero-knowledge property of the underlying zero-knowledge proof scheme.

□

In our generic construction, a requester $\mathcal{R}$ can not share his credentials with others. Because, if he wants to share one credential with others, he must reveal his secret key $SK_{\mathcal{R}}$ to them. If it is, all credentials of $\mathcal{R}$ will be shared with others. This is the so-called *all-or-nothing non-transferability property* [CL01].

Theorem 7.3 *Our generic construction of DSSO is* $(T, q_1, q_2, q_3, \epsilon(\ell))$*-secure against the coalition credential forging attacks if the broadcast encryption scheme is* $(T', N, q_2, \epsilon_1(\ell))$*-secure against the chosen ciphertext attacks (or IND-CCA2) and the signature scheme is* $(T'', q_3, \epsilon'(\ell))$*-strongly existentially unforgeable (or SEU-CMA), where* $T'' = T + T' + \Theta(T + T')$ *and* $\epsilon'(\ell) = \epsilon(\ell)\cdot(1-\epsilon_1(\ell))^{q_3}$.

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q_1, q_2, q_3, \epsilon(\ell))$ break the coalition credential unforgeability, we will show that there exists an algorithm $\mathcal{B}$ that can $(T'', q_3, \epsilon'(\ell))$ break the strongly existential unforgeability of the signature scheme. $\mathcal{A}$ works as the coalition of malicious requesters and service providers.

Initialization. $\mathcal{A}$ submits a target requester $\mathcal{R}^*$ for whom he wants to forge a credential and a target service provider $\mathcal{SP}^*$ whom he wants to access, where $\mathcal{R}^* \notin \mathcal{A}$ and $\mathcal{SP}^* \notin \mathcal{A}$.

Setup. $\mathcal{B}$ runs $\mathsf{Setup}(1^\ell)$ to generates the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{B}$ responds $\mathcal{A}$ with $(params, IPK)$.

Registration Queries. $\mathcal{A}$ can adaptively query the registration oracle. $\mathcal{A}$ submits $\{RI_{\mathcal{R}_1}, RI_{\mathcal{R}_2}, \cdots, RI_{\mathcal{R}_{t_1}}\}$ and $\{RI_{\mathcal{SP}_1}, RI_{\mathcal{SP}_2}, \cdots, RI_{\mathcal{SP}_{t_2}}\}$, where $q_1 = t_1 + t_2$, $RI_{\mathcal{SP}^*} \neq RI_{\mathcal{SP}_j}$ and $j = 1, 2, \cdots, t_2$. $\mathcal{B}$ randomly assigns an identifier $ID_{\mathcal{R}_i}$ and selects a set $\mathbb{A}_{\mathcal{R}_i}$ which consists of the identifiers of service providers, for $i = 1, 2, \cdots, t_1$. $\mathcal{B}$ randomly assigns an identifier $ID_{\mathcal{SP}_j}$ and runs $\mathsf{BKeyGen}(params, N) \rightarrow SK_{\mathcal{SP}_j}$, for $j = 1, 2, \cdots, t_2$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{(ID_{\mathcal{R}_i}, \mathbb{A}_{\mathcal{R}_i})\}_{i=1}^{t_1}$ and $\{(ID_{\mathcal{SP}_j}, SK_{\mathcal{SP}_j})\}_{j=1}^{t_2}$ and stores $\{(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, \mathbb{A}_{\mathcal{R}_i})\}_{i=1}^{q_1}$ and $\{(\mathcal{SP}_j, ID_{\mathcal{SP}_j}, SK_{\mathcal{SP}_j})\}_{j=1}^{t_2}$, where $PK_{\mathcal{R}_i}$ is the public key of $\mathcal{R}_i$ for $i = 1, 2, \cdots, t_1$.

Credential Generation Queries. $\mathcal{A}$ can adaptively query the credential generation oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1}, PK_{\mathcal{R}_1}), (ID_{\mathcal{R}_2}, PK_{\mathcal{R}_2}), \cdots, (ID_{\mathcal{R}_{q_2}}, PK_{\mathcal{R}_{q_2}})\}$, where the constraints are $(ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}) \neq (ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*})$ and $ID_{\mathcal{SP}^*} \notin \mathbb{A}_{\mathcal{R}_i}$. $\mathcal{B}$ runs $\mathsf{BEnc}(params, |\mathbb{A}_{\mathcal{R}_i}|) \rightarrow (Hdr_{\mathcal{R}_i}, K_{\mathcal{R}_i})$, $\mathsf{Sign}(params, ID_{\mathcal{R}_i}, PK_{\mathcal{R}_i}, AA_{\mathcal{R}_i}, T_{\mathcal{R}_i}, ISK) \rightarrow \delta_{\mathcal{R}_i}$ and $\mathcal{E}(params, K_{\mathcal{R}_i}, \delta_{\mathcal{R}_i}) \rightarrow CT_{\mathcal{R}_i}$, for $i = 1, 2, \cdots, q_2$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{Cred_{\mathcal{R}_i} = (\mathbb{A}_{\mathcal{R}_i}, Hdr_{\mathcal{R}_i}, CT_{\mathcal{R}_i})\}_{i=1}^{q_2}$.

Credential verification queries. $\mathcal{A}$ can adaptively query the credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}_1},\ PK_{\mathcal{R}_1},\ Cred_{\mathcal{R}_1}),\ (ID_{\mathcal{R}_2},\ PK_{\mathcal{R}_2},\ Cred_{\mathcal{R}_2}),\ \cdots,\ (ID_{\mathcal{R}_{q_3}}, PK_{\mathcal{R}_{q_3}}, Cred_{\mathcal{R}_{q_3}})\}$, where the only constraint is $ID_{\mathcal{SP}^*} \notin \mathbb{A}_{\mathcal{R}_i}$. $\mathcal{B}$ runs $\mathsf{BDec}(params, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}_i}) \rightarrow K_{\mathcal{R}_i}$ with $ID_{\mathcal{SP}_j} \in \mathbb{A}_{\mathcal{R}_i}$, decrypts $CT_{\mathcal{R}_i}$, obtains $\delta_{\mathcal{R}_i}$ and runs $\mathsf{Ver}(params,\ ID_{\mathcal{R}_i},\ PK_{\mathcal{R}_i},\ AA_{\mathcal{R}_i},\ T_{\mathcal{R}_i},\ IPK,\ \delta_{\mathcal{R}_i}) \rightarrow True/False$, for $i = 1, 2, \cdots, q_3$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{True/False\}$.

Output. $\mathcal{A}$ outputs a credential $Cred_{R^*} = (\mathbb{A}_{\mathcal{R}^*},\ Hdr_{\mathcal{R}^*},\ CT_{\mathcal{R}^*})$, where $(ID_{\mathcal{R}^*}, Cred_{\mathcal{R}^*}) \notin \{(ID_{\mathcal{R}_1}, Cred_{\mathcal{R}_1}),\ (ID_{\mathcal{R}_2}, Cred_{\mathcal{R}_2}),\ \cdots,\ (ID_{\mathcal{R}_{q_2}}, Cred_{\mathcal{R}_{q_2}})\}$ and $ID_{\mathcal{SP}^*} \in \mathbb{A}_{\mathcal{R}^*}$.

$\mathcal{B}$ runs $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*})$ with $ID_{\mathcal{SP}} \in \mathbb{A}_{\mathcal{R}^*}$.

1. If $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*}) \rightarrow \perp$, $\mathcal{B}$ aborts. The simulation fails.

2. If $\mathsf{BDec}(param, SK_{\mathcal{SP}_j}, Hdr_{\mathcal{R}^*}) \rightarrow K_{\mathcal{R}^*}$ and $\mathcal{B}$ can decrypt $CT_{\mathcal{R}^*}$ to obtain $\delta_{\mathcal{R}^*}$, $\mathcal{B}$ aborts. $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CCA2 security of the broadcast encryption scheme. Due to the broadcast encryption scheme is $(T', N, q_2, \epsilon_1(\ell))$-secure against chosen ciphertext attacks, the advantage that $\delta_{\mathcal{R}^*}$ can be obtained is at least ϵ_1.

3. If $\mathcal{B}$ does not abort, he can obtain a valid signature $\delta_{\mathcal{R}^*}$ on $(params, ID_{\mathcal{R}^*}, PK_{\mathcal{R}^*}, AA_{\mathcal{R}^*}, T_{\mathcal{R}^*})$ with the advantage at least $\epsilon(\ell)$.

Now, we compute the advantage with which $\mathcal{B}$ does not abort. If the broadcast encryption scheme is $(T', N, q_2, \epsilon_1(\ell))$-secure against the chosen ciphertext attacks, $\mathcal{B}$ can abort with the probability at most $\epsilon_1(\ell)$. Therefore the advantage with which $\mathcal{B}$ dose not abort at the q_3 credential verification queries is at least $(1 - \epsilon_1(\ell))^{q_3}$. So, the advantage with which $\mathcal{B}$ can break the strongly existential unforgeability of the underlying signature scheme is at least $\epsilon(\ell) \cdot (1 - \epsilon_1(\ell))^{q_3}$. This contradicts the assumption that the signature scheme is $(T'', q_3, \epsilon'(\ell))$-strongly existentially unforgeable.

□

Theorem 7.4 *Our generic construction of DSSO is $(T, q, \epsilon(\ell))$-forward secure if the broadcast encryption scheme is $(T', N, q, \epsilon(\ell))$-secure against the chosen ciphertext attacks (or IND-CCA2), where $T' = T + \Theta(T)$.*

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the forward security of our generic construction of DSSO, we will show that there exists an algorithm $\mathcal{B}$ that can $(T', N, q, \epsilon(\ell))$ break the IND-CCA2 security of the broadcast encryption scheme. $\mathcal{A}$ works as a malicious service provider. By $\mathbb{A}^O_{\mathcal{R}}$, we denote the access right of $\mathcal{R}$ before $\mathcal{A}$'s identifier $ID_{\mathcal{A}}$ is listed in $\mathbb{A}_{\mathcal{R}}$.

Setup. $\mathcal{B}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{B}$ sends $(params, IPK)$ to $\mathcal{A}$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query the credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^1_{\mathcal{R}}), (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^2_{\mathcal{R}}), \cdots, (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^q_{\mathcal{R}})\}$, where $Cred^i_{\mathcal{R}}$ is issued after $ID_{\mathcal{A}}$ has been added to $\mathbb{A}_{\mathcal{R}}$. $\mathcal{B}$ runs $\mathsf{BDec}(params, SK_{\mathcal{SP}_j}, Hdr^i_{\mathcal{R}}) \rightarrow K^i_{\mathcal{R}}$ where $ID_{\mathcal{SP}_j} \in \mathbb{A}^O_{\mathcal{R}} \bigcup \{ID_{\mathcal{A}}\}$, decrypts $CT^i_{\mathcal{R}}$ and obtains $\delta^i_{\mathcal{R}}$. $\mathcal{B}$ runs $\mathsf{Ver}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^i_{\mathcal{R}}, T^i_{\mathcal{R}}, IPK, \delta^i_{\mathcal{R}}) \rightarrow True/False$, for $i = 1, 2, \cdots, q$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{True/False\}$.

Challenge. $\mathcal{B}$ runs $\mathsf{BEnc}(params, |\mathbb{A}^O_{\mathcal{R}}|) \rightarrow (Hdr^O_{\mathcal{R}}, K^O_{\mathcal{R}})$, $\mathsf{Sign}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^O_{\mathcal{R}}, T^O_{\mathcal{R}}, ISK) \rightarrow \delta^O_{\mathcal{R}}$ and $\mathcal{E}(params, K^O_{\mathcal{R}}, \delta^O_{\mathcal{R}}) \rightarrow CT^O_{\mathcal{R}}$. Let $Cred^O_{\mathcal{R}} = (\mathbb{A}^O_{\mathcal{R}}, Hdr^O_{\mathcal{R}}, CT^O_{\mathcal{R}})$. $\mathcal{B}$ randomly creates a $Cred^*_{\mathcal{R}}$ which has the same distribution with $Cred^O_{\mathcal{R}}$. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$ and obtains one bit $b \in \{0,1\}$. $\mathcal{B}$ sets $Cred_b = Cred^O_{\mathcal{R}}$ and $Cred_{1-b} = Cred^*_{\mathcal{R}}$. $\mathcal{B}$ responds $\mathcal{A}$ with $(Cred_b, Cred_{1-b})$.

Output. $\mathcal{A}$ outputs his guess b' on b.

If $\left|\Pr[b' = b] - \frac{1}{2}\right| \geq \epsilon(\ell)$, $\mathcal{A}$ can distinguish $\delta^O_{\mathcal{R}}$ with the same advantage. Namely, $\mathcal{A}$ is not a receiver in the broadcast encryption scheme, but can distinguish the message encryption key $K^O_{\mathcal{R}}$ from $Hdr^O_{\mathcal{R}}$ with the advantage at least $\epsilon(\ell)$. So, $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CCA2 security of the broadcast encryption scheme with the advantage at least $\epsilon(\ell)$.

□

Theorem 7.5 *Our generic construction of DSSO is $(T, q, \epsilon(\ell))$-backward secure if the broadcast encryption scheme is $(T', N, q, \epsilon(\ell))$-secure against the chosen ciphertext attacks (or IND-CCA2), where $T' = T + \Theta(T)$.*

Proof: If there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the backward security of our generic construction of DSSO, we will show that there exists an algorithm $\mathcal{B}$ that can $(T', N, q, \epsilon(\ell))$ break the IND-CCA2 security of the broadcast encryption scheme. $\mathcal{A}$ works as a malicious service provider. By $\mathbb{A}^N_{\mathcal{R}}$, we denote the access right of $\mathcal{R}$ after $\mathcal{A}$'s identifier $ID_{\mathcal{A}}$ has been deleted from $\mathbb{A}_{\mathcal{R}}$.

Setup. $\mathcal{B}$ runs $\mathsf{Setup}(1^\ell)$ to generate the public parameters $params$ and a secret-public key pair (ISK, IPK). $\mathcal{B}$ sends $(params, IPK)$ to $\mathcal{A}$.

Credential Verification Queries. $\mathcal{A}$ can adaptively query the credential verification oracle. $\mathcal{A}$ submits $\{(ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^1_{\mathcal{R}}), (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^2_{\mathcal{R}}), \cdots, (ID_{\mathcal{R}}, PK_{\mathcal{R}}, Cred^q_{\mathcal{R}})\}$, where $Cred^i_{\mathcal{R}}$ was issued before $ID_{\mathcal{A}}$ was deleted from $\mathbb{A}_{\mathcal{R}}$. $\mathcal{B}$ runs $\mathsf{BDec}(params, SK_{\mathcal{SP}_j}, Hdr^i_{\mathcal{R}}) \rightarrow K^i_{\mathcal{R}}$ where $ID_{\mathcal{SP}_j} \in \mathbb{A}^N_{\mathcal{R}} \bigcup \{ID_{\mathcal{A}}\}$, decrypts $CT^i_{\mathcal{R}}$ and obtains $\delta^i_{\mathcal{R}}$. $\mathcal{B}$ runs $\mathsf{Ver}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^i_{\mathcal{R}}, T^i_{\mathcal{R}}, IPK, \delta^i_{\mathcal{R}}) \rightarrow True/False$, for $i = 1, 2, \cdots, q$. $\mathcal{B}$ responds $\mathcal{A}$ with $\{True/False\}$.

Challenge. $\mathcal{B}$ runs $\mathsf{BEnc}(params, |\mathbb{A}^N_{\mathcal{R}}|) \rightarrow (Hdr^N_{\mathcal{R}}, K^N_{\mathcal{R}})$, $\mathsf{Sign}(params, ID_{\mathcal{R}}, PK_{\mathcal{R}}, AA^N_{\mathcal{R}}, T^N_{\mathcal{R}}, ISK) \rightarrow \delta^N_{\mathcal{R}}$ and $\mathcal{E}(params, K^N_{\mathcal{R}}, \delta^N_{\mathcal{R}}) \rightarrow CT^N_{\mathcal{R}}$. Let $Cred^N_{\mathcal{R}} = (\mathbb{A}^N_{\mathcal{R}},$

$Hdr^N_{\mathcal{R}}, CT^N_{\mathcal{R}}$). $\mathcal{B}$ randomly creates a $Cred^*_{\mathcal{R}}$ which has the same distribution with $Cred^N_{\mathcal{R}}$. $\mathcal{B}$ flips an unbiased coin with $\{0,1\}$ and obtains one bit $b \in \{0,1\}$. $\mathcal{B}$ sets $Cred_b = Cred^N_{\mathcal{R}}$ and $Cred_{1-b} = Cred^*_{\mathcal{R}}$. $\mathcal{B}$ responds $\mathcal{A}$ with $(Cred_b, Cred_{1-b})$.

Output. $\mathcal{A}$ outputs his guess b' on b.

If $\left|\Pr[b' = b] - \frac{1}{2}\right| \geq \epsilon(\ell)$, $\mathcal{A}$ can distinguish $\delta^N_{\mathcal{R}}$ with the same advantage. Namely, $\mathcal{A}$ is not a receiver in the broadcast encryption scheme, but can distinguish the message encryption key $K^N_{\mathcal{R}}$ from $Hdr^N_{\mathcal{R}}$ with the advantage at least $\epsilon(\ell)$. So, $\mathcal{B}$ can use $\mathcal{A}$ to break the IND-CCA2 security of the broadcast encryption scheme with the advantage at least $\epsilon(\ell)$.

□

7.5 Chapter Summary

The existing SSO schemes suffer from various security issues such as illegally sharing credentials and difficulties in user revocation. In this chapter, we proposed the formal definitions and security models for SSO and DSSO, We proposed a generic construction of DSSO and proved its security. Our generic construction provides a sound solution to avoid the mentioned attacks.

Part III

Protection of Accessed Contents & Personal Information

Chapter 8

Attribute-based Oblivious Access Control

In this chapter, we first construct a CP-ABE scheme with constant communication and computation cost, then propose an attribute-based oblivious access control (ABOAC) scheme. Parts of this work appeared in [HSMY12a].

8.1 Introduction

In an attribute-based access control (ABAC) system, users are identified by their distinct attributes. An access request is accepted or denied depending on whether a requester's attributes satisfy the specified access policies. The *magic* of an ABAC system not only lies in its high flexibility and strong expressibility, but also its anonymity. For example, Charlie associates a service with a set of attributes $\mathbf{S} = \{American, Student, Adult\}$ such that only the users whose attributes include $\mathbf{S}$ can access the service. Suppose that Alice and Bob hold sets of attribute $\mathbf{S}_A = \{American, Married, Adult, Student\}$ and $\mathbf{S}_B = \{American, Adult, Vegetarian, Student\}$, respectively. In the case that the service is accessed, Charlie cannot determine who accessed the service as both of Alice and Bob are authorized to access the service. What Charlie knows is $\mathbf{S} \subseteq \mathbf{S}_A$ and $\mathbf{S} \subseteq \mathbf{S}_B$. Therefore, Charlie cannot identify the real identity of the requester from the required attributes. This implies that an ABAC system can provide anonymity to users.

Although attribute-based systems can provide some sound properties, both the computation cost and communication cost are linear in the number of the required attributes. Hence, existing ABAC schemes are not suitable to the systems with limited communication and computation ability, such as wireless sensor and actor networks (WSANs) [AK04] and mobile ad hoc networks (MANETs) [NN08]. In a WSANs system, the sensors are lower price and lower power devices with limited computation, communication and sensing ability [AK04]. Similarly, the nodes in a

MANETs system have limited power, computation ability and small memory space [NN08]. Thereafter, it is an interesting and challenging work to design an ABAC scheme with constant computation and communication cost.

8.1.1 Related Work

The literature about ABE is referred to Section 5.1.1. In this section, we mainly review ABE schemes with constant computation and communication cost.

One intrinsic flaw of ABE schemes is that the length of the ciphertexts is linear in the number of the required attributes. Solutions towards reducing the length of the ciphertexts have been proposed. In some of these schemes, the computation cost of the encryption algorithms is constant, but the exponential and pairing operations executed in the decryption algorithm are linear in the number of the required attributes. Therefore, these schemes are not an ideal primitive for the systems with limited computing ability, such as WSANs, MANETS, *etc.* To name a few, Emura *et al.* [EMN+09] proposed a CP-ABE scheme with constant sized ciphertext, where for a set of attributes, only one secret key is generated. In this scheme, a user can only decrypt the ciphertext which requires the exact attributes which he holds. Especially, a user cannot decrypt the ciphertext where the required attributes are included in his attributes as he cannot use his attributes separately. So far as this property is concerned, this scheme is more like an IBE scheme, where all the attributes held by a user can be mapped into his sole identity. Herranz, Laguillaumie and Ràfols [HLR10] proposed a threshold ABE scheme with constant sized ciphertext, which was derived from the threshold public-key encryption proposed by Delerablée and Pointcheval [DP08]. Prior to executing the decryption algorithm, a user must aggregate the required secret keys to one value using the Aggregate algorithm proposed in [DPP07]. As pointed in [DPP07], the running time of the Aggregate algorithm is about $\frac{\gamma(\gamma-1)}{2}T_e$, where γ denotes the number of the required attributes. Zhou and Huang [ZH10] proposed a CP-ABE scheme with constant sized ciphertext based on the q-DBDHE assumption. The computational cost of the encryption algorithm is constant, but the number of the pairing operations executed in the decryption algorithm are linear in the number of the required attributes. Attrapadung, Libert and Panafieu [ALP11] proposed a KP-ABE scheme with constant sized ciphertext, where the number of the exponential and the pairing operations executed in the decryption algorithm are linear in the number of

the required attributes. Subsequently, Chen, Zhang and Feng [CZF11] proposed a non-monotonic CP-ABE scheme with constant sized ciphertext and computation cost. Both [ALP11] and [CZF11] are based on non-standard assumption (q-DBDHE assumption).

In an ABE scheme, a user must obtain a secret key for each of his attributes from the central authority. Therefore, it is hard to guarantee that the received secret keys are generated correctly and not tampered if they cannot be verified. In the scenario that a user cannot decrypt a ciphertext when his attributes satisfy the specified access policies, he cannot determine which secret keys caused this. He also cannot detect whether his secret keys or the cihpertext are not created correctly. Especially, if the issuer and the encryptor are different entities, the user cannot detect who is malicious. This will risk the user's access right. While, most of the previous schemes did not consider the verification of the issued secret keys.

To protect users' privacy, anonymity should be addressed. We note that it is unfortunately insufficient [IKOS06]. This is because anonymity can hide who the user is, but it cannot hide what actions the user performed. For instance, suppose that a user can access resources anonymously. Although we cannot identify who he/she is, we can guess that it is Alice with high probability if it can access Alice's medical data, financial condition and insurance records. Hence, in terms of *privacy*, we need a system where both the identity of the user and the actions performed by him can be hidden.

Proposed by Rabin [Rab81], k-out-of-m oblivious transfer (OT_k^m) is a cryptosystem where a sender and a receiver have a set of messages $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$ and a set of choices $\mathcal{C} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\} \subseteq \{1, 2, \cdots, m\}$, respectively. After an interaction, the receiver can obtain the intended messages $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$, while the sender knows nothing about the receiver's choices. Adaptive k-out-of-m oblivious transfer ($OT_{k\times 1}^m$) is a strongly secure OT scheme, where a receiver can obtain messages from the sender adaptively [NP99a, NP99b]. OT schemes have been used as an efficient primitive to hide users' actions [AIR01, CGH09, CDN09].

Friken, Atallah and Li [FAL06] proposed three ABAC schemes, where both the access policies and the receivers' attributes can be hidden. In the first scheme, a receiver knows a superset of the attributes required by the access policy. In the second scheme, a receiver knows the number of the attributes which he satisfies. While, in the third scheme, a receiver only knows the upper bound of the attributes which he can use to access the system. The sender only knows the number of attributes which

a receiver must use to access the system. Their schemes were based on homomorphic encryption [RAD77], 1-out-of-2 oblivious transfer [Rab81] and set intersection [KS92]. These schemes provided sound solutions to protect users' privacy. One disadvantage of these schemes is its efficiency. The communication complexity in these three schemes are $\Theta(n)$, $\Theta(\gamma n)$ and $\Theta(\gamma n)$, respectively, where n is the number of the attributes required by the access policies, and γ is the number of the attributes held by a user. For each required attribute in the access policy, the encryption operations executed in these schemes are $\Theta(1)$, $\Theta(\gamma)$ and $\Theta(\gamma)$, respectively. The interactions for each required attribute are 3 rounds, 5 rounds and 5 rounds, respectively. Furthermore, the computation cost depends on the exploited encryption scheme and OT scheme.

Coull, Green and Hohenberger [CGH09] proposed an oblivious transfer with access control (AC-OT) scheme by introducing an anonymous credential scheme to an OT scheme, where the access policy is determined by a state graph. Each node in the graph represents a state, and each edge represents a transaction from one state to another. In order to access the database, a user must prove that he has obtained the required credentials (attributes) in zero-knowledge. Camenisch, Dubovitskaya and Neven [CDN09] proposed another AC-OT scheme which improved the scheme [CGH09]. This scheme avoids to re-issue credentials to users at each transfer by the following two approaches. In the first approach, they assigned a state to a subset of attributes which a user holds, with a self-loop which can be accessed using this subset of attributes. In the second approach, they assigned a state to a subset of attributes which are published as the access policy, with a self-loop for each data which is associated with this subset of attributes. Let $|\mathbf{A}_{C_i}|$ denotes the number of the attributes required by the ith record (data), for $i = 1, 2, \cdots, m$. For a set of choices $\mathbf{C} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}$, the computation cost and communication cost in these two schemes are $\Theta(\sum_{i=1}^{k} |\mathbf{A}_{C_{\sigma_i}}|)$ and $\Theta(m)$, respectively.

Zhang *et al.* [ZAW+10] proposed a new AC-OT scheme which is based on the CP-ABE scheme [LOS+10] and $OT^m_{k\times 1}$ scheme [CNS07]. As mentioned in [Wat11], the CP-ABE scheme [LOS+10] designed in the composite order ($N = p_1p_2p_3$) bilinear groups is not efficient, where p_1, p_2 and p_3 are different prime numbers. The length of the ciphertexts in this scheme is linear in the number of the required attributes. Additionally, both the exponential and the pairing operations executed in the decryption algorithm are linear in the number of the required attributes. Furthermore, in order to introduce the CP-ABE scheme [LOS+10] to $OT^m_{k\times 1}$ scheme

[CNS07], a data encapsulation mechanism (DEM) must be exploited to encrypt the messages from different message spaces. Consequently, the computational cost and communication cost in this scheme are $\Theta(\sum_{i=1}^{k} |\mathbf{A}_{C_{\sigma_i}}|)$ and $\Theta(\sum_{i=1}^{m} |\mathbf{A}_{C_i}|)$, respectively.

Rial and Preneel [RP10] proposed a blind ABE and an AC-OT scheme by providing a blind key extract protocol for the CP-ABE scheme [BSW07]. While, the CP-ABE scheme [BSW07] was proven to be secure in the generic group model, instead of being reduced to a complexity assumption. The length of the ciphertexts and the computation cost of the decryption algorithm in [BSW07] are linear in the number of the required attributes. Furthermore, the computational cost and communication cost in this scheme are $\Theta(\sum_{i=1}^{k} |\mathbf{A}_{C_{\sigma_i}}|)$ and $\Theta(\sum_{i=1}^{m} |\mathbf{A}_{C_i}|)$, respectively.

8.1.2 Our Contribution

In ABE schemes, complex and fine-gained access structures can be expressed. Meanwhile, OT schemes have been used to hide the actions performed by a user. Hence, the combination of ABE and OT schemes provide an elegant solution to protect users' privacy in privacy-sensitive systems, such as medical records, patent searches, *etc.* However, the computational and communication costs in existing schemes are linear in the number of the required attributes. It is an interesting and challenging work to design an attribute-based oblivious access control (ABOAC) scheme with constant computation and communication costs. This is necessary in the systems with limited computing and communication ability, such as WSANs, MANETs, *etc.*

In this chapter, we first propose an ABE scheme with constant sized ciphertext. We observe that both the encryption and decryption algorithms in our scheme are efficient. For an encryption and decryption procedure, only 3 exponentiations and 2 pairing operations are executed, respectively. This is in contrast to the previous ABE schemes where the numbers of exponentiation and pairing operations executed in the encryption and decryption algorithms are linear in the number of the required attributes. Furthermore, the secret key for each attribute can be efficiently verified.

Then, we propose an ABOAC scheme by introducing the proposed ABE scheme to an OT scheme. In our ABOAC scheme, a requester can obtain services obliviously if his attributes satisfy the specified access policies. As a result, the requester does not release anything about his attributes and the selected services to the service provider. The service provider only knows the number of the services accessed by

the requester. Hence, both the attributes of the requester and the actions performed by him can be hidden. Notably, in our ABOAC scheme, for each service encrypted under the required attributes, only one-round interaction is executed between the service provider and the requester. The service provider needs to execute 3 exponential operations, and the requester needs to execute 2 pairing and 2 exponential operations.

8.1.3 Chapter Organization

This chapter is organized as follows. In Section 8.2, we propose the formal definitions and security models of the CP-ABE and ABOAC schemes. A new ABE with constant computation and communication cost is proposed, and proven in Section 8.3. In Section 8.4, an ABOAC scheme is proposed and proven. Finally, Section 8.5 summarizes this chapter.

8.2 Formal Definitions and Security Models

In this section, we introduce the formal definitions and security models of CP-ABE and ABOAC schemes.

8.2.1 Cipher-Policy Attribute-based Encryption

A CP-ABE scheme consists of the following algorithms [BSW07]:

$\mathsf{Setup}(1^\ell) \rightarrow (params, MSK)$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$ and a master secret key MSK.

$\mathsf{KeyGen}(params, \mathbf{A}_\mathcal{U}, MSK) \rightarrow SK_\mathcal{U}$. The key generation algorithm takes as input the public parameters $params$, a set of attributes $\mathbf{A}_\mathcal{U}$ and the master secret key MSK, and outputs a secret key $SK_\mathcal{U}$ for a user $\mathcal{U}$ with a set of attributes $\mathbf{A}_\mathcal{U}$.

$\mathsf{Enc}(params, \mathbb{A}_C, M) \rightarrow CT$. The encryption algorithm takes as input the the public parameters $params$, an access structure $\mathbb{A}_C$ and a message M, and outputs a ciphertext CT which can be decrypted by the user who holds a set of attributes $\mathbf{A}_\mathcal{U}$ if $\mathbf{A}_\mathcal{U} \in \mathbb{A}_C$.

$\mathsf{Dec}(params, SK_{\mathcal{U}}, CT) \rightarrow M$. The decryption algorithm takes as input the public parameters $params$, the secret key $SK_{\mathcal{U}}$ and the ciphertext CT, and outputs the message M.

Definition 8.1 *We say that a cipher-policy attribute-based encryption scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Dec}(params, SK_{\mathcal{U}}, CT) & \mathsf{Setup}(1^{\ell}) \rightarrow (params, MSK); \\ \rightarrow M & \mathsf{KeyGen}(params, \mathbf{A}_{\mathcal{U}}, MSK) \rightarrow SK_{\mathcal{U}}; \\ & \mathsf{Enc}(params, \mathbb{A}_C, M) \rightarrow CT; \\ & \mathbf{A}_{\mathcal{U}} \in \mathbb{A}_C \end{array}\right] = 1$$

where the probability is taken over the random coins which are consumed by the algorithms in the scheme.

8.2.2 Selective-Attributes Model

We propose a selective-attribute model which is slightly stronger than the selective-set model introduced in [GPSW06]. This model is analogous to the selective-ID model in the IBE scheme [BF01]. This model is defined by the following game executed between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

Initiation. $\mathcal{A}$ submits a set of attributes $\mathbf{A}^* = \{a^*\}$ which she wants to be challenged with.

Setup. $\mathcal{C}$ runs the $\mathsf{Setup}(1^{\ell})$ algorithm to generate the public parameters $params$ and a master secret key MSK. $\mathcal{C}$ responds $\mathcal{A}$ with $params$.

Phase 1. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}q_1$, where the only restrict is $\mathbf{A}^* \not\subseteq \mathbf{A}_i$ for $i = 1, 2, \cdots, q_1$. $\mathcal{C}$ responds $\mathcal{A}$ with $\mathsf{KeyGen}(params, \mathbf{A}_i, MSK)$ for $i = 1, 2, \cdots, q_1$.

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{C}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $b \in \{0, 1\}$. It selects an access structure $\mathbb{A}$ and computes $CT^* = \mathsf{Enc}(params, \mathbb{A}, M_b)$, where $\mathbf{A}^* \in \mathbb{A}$. $\mathcal{C}$ responds $\mathcal{A}$ with CT^*.

Phase 2. $\mathcal{A}$ can adaptively query secret keys for sets of attributes $\mathbf{A}_{q_1+1}, \mathbf{A}_{q_1+2}, \cdots, \mathbf{A}_q$, where the only constraint is $\mathbf{A}^* \not\subseteq \mathbf{A}_j$, for $j = q_1 + 1, q_1 + 2, \cdots, q$. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 8.2 *An cipher-policy attribute-based encryption is (T, q, ϵ)-secure against chosen plaintext attacks (or IND-CPA) if no PPT adversary $\mathcal{A}$ making at most q secret key queries can win the game with the advantage*

$$Adv_{\mathcal{A}-ABE}^{IND-sa-CPA}(\ell) = \left| Pr[b' = b] - \frac{1}{2} \right| \geq \epsilon(\ell)$$

in the above selective-attribute model.

8.2.3 Attribute-based Oblivious Access Control

An ABOAC scheme consists of the following four algorithms:

Setup$(1^\ell) \rightarrow (params, MSK, (SSK, SPK))$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$, a master key MSK and a secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (SSK, SPK)$ for the service provider $\mathcal{SP}$.

KeyGen$(params, \mathbf{A}_\mathcal{R}, MSK) \rightarrow SK_\mathcal{R}$. The key generation algorithm takes as inputs the public parameters $params$, a set of attributes $\mathbf{A}_\mathcal{R}$ and the master key MSK, and outputs a secret key $SK_\mathcal{R}$ for a requester $\mathcal{R}$ with a set of attributes $\mathbf{A}_\mathcal{R}$.

Commit$(params, SSK, \mathbb{A}_i, M_i) \rightarrow CT_i$. Suppose that $\mathcal{SP}$ manages m messages $M_1, M_2, \cdots, M_m$. To commit a message M_i, the commitment algorithm takes as input the public parameters $params$, the secret key SSK, an access structure $\mathbb{A}_i$ and the message M_i, and outputs a ciphertext CT_i which can be decrypted by a requester $\mathcal{R}$ who holds a set of attributes $\mathbf{A}_\mathcal{R}$ with $\mathbf{A}_\mathcal{R} \in \mathbb{A}_i$, for $i = 1, 2, \cdots, m$.

Transf$(\mathcal{SP}(params, SSK) \leftrightarrow \mathcal{R}(params, SPK, \mathbf{C}, SK_\mathcal{R})) \rightarrow (\perp, \mathbf{M})$. The transfer algorithm is an interactive algorithm executed between $\mathcal{SP}$ and $\mathcal{R}$. $\mathcal{SP}$ takes as input the public parameters $params$ and his secret key SSK, and outputs nothing. $\mathcal{R}$ takes as input the public parameters $params$, a set of choices $\mathbf{C} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\} \subseteq \{1, 2, \cdots, m\}$ and his secret key $SK_\mathcal{R}$, and outputs messages $\mathbf{M} = \{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$.

Definition 8.3 *We say that an attribute-based oblivious access control scheme is correct if*

$$\Pr\left[\begin{array}{l|l} \mathsf{Transf}(\mathcal{SP}(\boxtimes) \leftrightarrow \mathcal{R}(\boxplus)) & \mathsf{Setup}(1^{\ell}) \rightarrow (params, MSK, (SSK, SPK)); \\ \rightarrow (\perp, \mathbf{M}) & \mathsf{KeyGen}(params, \mathbf{A}_{\mathcal{R}}, MSK) \rightarrow SK_{\mathcal{R}} \\ & \mathsf{Commit}(params, SSK, \mathbb{A}_i, M_i) \rightarrow CT_i; \\ & \mathbf{A}_{\mathcal{R}} \in \mathbb{A}_{\sigma_j} \textit{ for } j = 1, 2, \cdots, k \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by the algorithms in the scheme, $\boxtimes = (params, SSK)$, $\boxplus = (params, SPK, \mathbf{C}, SK_{\mathcal{R}})$, $\mathbf{C} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ *and* $\mathbf{M} = \{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$.

8.2.4 Security Model for Attribute-based Oblivious Access Control

The security model of ABOAC schemes is defined as follows. For the privacy of the requester $\mathcal{R}$, his choices should be unconditionally secure and his attributes are not released to the service providers $\mathcal{SP}$s, even the number and a superset of his attributes. For the security of $\mathcal{SP}$, a real world paradigm and an ideal world paradigm are exploited. If there exists an adversary in the real world, there will exist an adversary in the ideal world such that the outputs of these two adversaries are indistinguishable. We name this model as half-simulation model, which is similar to the models in [NP99a, NP99b].

Privacy of Requester. An ABOAC scheme can protect the privacy of $\mathcal{R}$ if it can provide the following properties:

1. $\mathcal{R}$ releases nothing about his attributes to $\mathcal{SP}$s.

2. For any two different choice sets $\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ and $\{\sigma'_1, \sigma'_2, \cdots, \sigma'_k\}$, the transcripts received by $\mathcal{SP}$ corresponding to $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$ and $\{M_{\sigma'_1}, M_{\sigma'_2}, \cdots, M_{\sigma'_k}\}$ are indistinguishable. Especially, the choices of $\mathcal{R}$ are unconditionally requester-secure if $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$ and $\{M_{\sigma'_1}, M_{\sigma'_2}, \cdots, M_{\sigma'_k}\}$ are identically distributed.

Security of Service Provider. Suppose that $\mathcal{R}$ has obtained the required secret keys. To define the security of $\mathcal{SP}$, we compare a real world experiment and an ideal world experiment. In the real world experiment, $\mathcal{R}$ and $\mathcal{SP}$ execute

the protocol. Meanwhile, in the ideal world expriment, the functionality of the protocol is replaced by a trusted third party (TTP). $\mathcal{SP}$ sends all his messages $\{M_1, M_2, \cdots, M_m\}$ to the TTP. $\mathcal{R}$ adaptively submits his choices $\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ to the TTP. If $\sigma_1, \sigma_2, \cdots, \sigma_k \in \{1, 2, \cdots, m\}$, the TTP responds $\mathcal{R}$ with $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$. An ABOAC scheme is service provider secure, if for any malicious requester $\mathcal{R}^*$ in the real world, there exists a requester $\hat{\mathcal{R}}^*$ in the ideal world such that the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

Semantic security. Let $\mathbf{A}^*$ be a set of the attributes which an adversary holds. If $\mathbf{A}^* \notin \mathbb{A}_i$, the adversary cannot obtain anything about the protected message M_i, for $i = 1, 2, \cdots, m$.

Definition 8.4 *We say that an attribute-based oblivious access control scheme is secure if it can protect the requester's privacy, and is service provider secure and semantically secure.*

8.3 Efficient Attribute-Based Encryption with Constant Cost

In this section, we propose a new ABE scheme where the length of the ciphertexts is constant. Furthermore, the encryption and decryption algorithms in our scheme is very efficient. For each encryption requiring t attributes, only 3 exponentiation operations and 2 pairing operations are executed in the encryption algorithm and decryption algorithm, respectively. Our idea is derived from the schemes [Wat05, Wat11, BW06]. We describe our ABE scheme in Figure 8.1.

Correctness. The correctness of the scheme is shown as follows.

$$\begin{aligned} Y &= D_{\mathcal{U},2} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} Y_{c_j} \\ &= g_2^{\alpha} h^{r_u} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} T_{c_j}^{r_u}, \end{aligned} \tag{8.1}$$

$$\begin{aligned} e(D_{\mathcal{U},1}, C_2) &= e(g^{r_u}, (h \cdot \prod_{a_{c_j} \in \mathbf{A}_C} T_{c_j})^s) \\ &= e(g, h)^{r_u s} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} e(g, T_{c_j})^{r_u s}, \end{aligned} \tag{8.2}$$

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let g, g_2, h be the generators of $\mathbb{G}$. Suppose that the set of universal attributes $\mathbb{U} = \{a_1, a_2, \cdots, a_n\} \subseteq \{0,1\}^n$. For each $a_j \in \mathbb{U}$, it chooses $T_j \stackrel{R}{\leftarrow} \mathbb{G}$. It generates a master secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (\alpha, g_1)$, where $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $g_1 = g^\alpha$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, g_1, g_2, h, T_1, T_2, \cdots, T_n)$.

KeyGen. To generate a secret key for a user $\mathcal{U}$ with a set of attributes $\mathbf{A}_\mathcal{U}$, this algorithm chooses $r_u \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$D_{\mathcal{U},1} = g^{r_U},\ D_{\mathcal{U},2} = g_2^\alpha h^{r_u} \text{ and } \{Y_{i_j} = T_{i_j}^{r_u}\}_{a_{i_j} \in \mathbf{A}_\mathcal{U}}.$$

The secret key for $\mathcal{U}$ is $SK_\mathcal{U} = (D_{\mathcal{U},1}, D_{\mathcal{U},2}, \{Y_{i_j}\}_{a_{i_j} \in \mathbf{A}_\mathcal{U}})$. It can be verified as follows:

$$e(g, D_{\mathcal{U},2}) \stackrel{?}{=} e(g_1, g_2) \cdot e(D_{\mathcal{U},1}, h) \text{ and } e(g, Y_{i_j}) \stackrel{?}{=} e(D_{\mathcal{U},1}, T_{i_j}) \text{ for } a_{i_j \in \mathbf{A}_\mathcal{U}}.$$

Encryption. Let $\mathbb{A}$ be a monotonic access structure and $\mathbf{A}_C \in \mathbb{A}$ be the minimal set in $\mathbb{A}$ [DT07][a]. To encrypt a message $M \in \mathbb{G}_\tau$ under $\mathbf{A}_C$, this algorithm chooses $s \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$C_0 = e(g_1, g_2)^s \cdot M,\ C_1 = g^s \text{ and } C_2 = (h \cdot \prod_{a_{c_j} \in \mathbf{A}_C} T_{c_j})^s.$$

The ciphertext is $CT = (C_0, C_1, C_2)$.

Decryption. To decrypt the ciphertext $CT = (C_0, C_1, C_2)$, $\mathcal{U}$ performs as follows.

1. Compute $Y = D_{\mathcal{U},2} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} Y_{c_j}$, where $Y_{c_j} \in SK_\mathcal{U}$ and $\mathbf{A}_C \subseteq \mathbf{A}_\mathcal{U} \in \mathbb{A}$.
2. Compute
$$C_0 \cdot \frac{e(D_{\mathcal{U},1}, C_2)}{e(C_1, Y)} = M$$

[a] By $\mathbf{A}_C \in \mathbb{A}$ is the minimal set of $\mathbb{A}$, we mean that $\mathbf{A}_C \subseteq \mathbf{B}$ if $\mathbf{B} \in \mathbb{A}$.

Figure 8.1: Attribute-based Encryption with Constant Cost

$$\begin{aligned} e(C_1, Y) =& e(g^s, g_2^{\alpha} h^{r_u} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} T_{c_j}^{r_u}) \\ =& e(g^{\alpha}, g_2)^s \cdot e(g, h)^{r_u s} \cdot \prod_{a_{c_j} \in \mathbf{A}_C} e(g, T_{c_j})^{r_u s} \\ =& e(g_1, g_2)^s \cdot e(D_{\mathcal{U},1}, C_2) \end{aligned} \tag{8.3}$$

and

$$\begin{aligned} C_0 \cdot \frac{e(D_{\mathcal{U},1}, C_2)}{e(C_1, Y)} =& M \cdot e(g_1, g_2)^s \cdot \frac{e(D_{\mathcal{U},1}, C_2)}{e(g_1, g_2)^s e(D_{\mathcal{U},1}, C_2)} \\ =& M \cdot e(g_1, g_2)^s \cdot \frac{1}{e(g_1, g_2)^s} \\ =& M \end{aligned} \tag{8.4}$$

Theorem 8.1 *Our attribute-based encryption scheme is $(T, q, \epsilon(\ell))$-secure against chosen plaintext attacks (or IND-CPA) in the selective-attribute model if the $(T', \epsilon'(\ell))$ decisional bilinear Diffie-Hellman assumption holds in $(e, p, \mathbb{G}, \mathbb{G}_\tau)$, where*

$$T' = T + (n + 4(q+1) + 3(|\mathbf{A}_1| + |\mathbf{A}_2| + \cdots + |\mathbf{A}_q|))T_e, \ \epsilon'(\ell) = \frac{\epsilon(\ell)}{2}$$

and $\mathbf{A}_j$ is the set of attributes queried by an adversary, for $j = 1, 2, \cdots, q$.

Proof: Suppose that there exists a PPT adversary $\mathcal{A}$ who can $(T, q, \epsilon(\ell))$ break the IND-CPA security of our CP-ABE scheme in the selective-attribute model, we can construct an algorithm $\mathcal{B}$ that can $(T', \epsilon'(\ell))$ break the DBDH assumption as follows.

The challenger $\mathcal{C}$ generates a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$. Let g be a generator of the group $\mathbb{G}$. He flips an unbiased coin with $\{0, 1\}$ and obtains one bit $\mu \in \{0, 1\}$. If $\mu = 0$, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ to $\mathcal{B}$; otherwise, he sends $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ to $\mathcal{B}$, where $z \overset{R}{\leftarrow} \mathbb{Z}_p$. $\mathcal{B}$ will output his guess μ' on μ.

Initialization. $\mathcal{A}$ submits an attributes $\mathbf{A}^* = \{a_i\}$.

Setup. $\mathcal{B}$ sets $g_1 = g^a$ and $g_2 = g^b$. It selects n random integers $e_1, e_2, \cdots, e_n \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $T_i = g_1^{e_i}$ and $T_j = g^{e_j}$, for $j \in \{1, 2, \cdots, n\} - \{i\}$. $\mathcal{B}$ chooses $\gamma \overset{R}{\leftarrow} \mathbb{Z}_p$, and sets $h = g_1^{-e_i} g^{\gamma}$. The public parameters are $(e, \mathbb{G}, \mathbb{G}_\tau, g, g_1, g_2, h, T_1, T_2, \cdots, T_n)$, while the master secret key is ab. $\mathcal{B}$ responds $\mathcal{A}$ with $(e, \mathbb{G}, \mathbb{G}_\tau, g, g_1, g_2, h, T_1, T_2, \cdots, T_n)$.

Phase 1. For a secret key query on a set of attributes $\mathbf{A}$ where the only restrict is $\mathbf{A}^* \not\subseteq \mathbf{A}$, $\mathcal{B}$ chooses $r \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$D_1 = g^{-r} g_2^{\frac{1}{e_i}},\ D_2 = g_2^{\frac{\gamma}{e_i}} h^{-r} \text{ and } \{Y_{v_j} = (g^{-r} g_2^{\frac{1}{e_i}})^{e_{i_j}}\}_{a_{v_j} \in \mathbf{A}}.$$

We claim that $(D_1, D_2, \{Y_{v_j}\}_{a_{v_j} \in \mathbf{A}})$ are correctly computed. Because, we have

$$\begin{aligned} g_2^{\frac{\gamma}{e_i}} h^{-r} &= g_1^{\frac{b\gamma}{ae_i}} (g_1^{-e_i + \frac{\gamma}{a}})^{-r} \\ &= (g_1^{-e_i + \frac{\gamma}{a}})^{\frac{b}{e_i}} g_1^b (g_1^{-e_i + \frac{\gamma}{a}})^{-r} \\ &= g_1^b (g_1^{-e_i + \frac{\gamma}{a}})^{-r + \frac{b}{e_i}} \\ &= g_2^a (g_1^{-e_i} g^\gamma)^{-r + \frac{b}{e_i}} \\ &= g_2^a h^{-r + \frac{b}{e_i}}. \end{aligned}$$

Let $r' = -r + \frac{b}{e_i}$, we have

$$g_2^{\frac{\gamma}{e_i}} h^{-r} = g_2^a h^{r'},$$

$$g^{-r} g_2^{\frac{1}{e_i}} = g^{-r + \frac{b}{e_i}} = g^{r'}$$

and

$$(g^{-r} g_2^{\frac{1}{e_i}})^{e_{v_j}} = (g^{-r + \frac{b}{e_i}})^{e_{v_j}} = (g^{r'})^{e_{v_j}} = T_{v_j}^{r'}.$$

Challenge. $\mathcal{A}$ submits two messages M_0 and M_1 with equal length. $\mathcal{B}$ flips an unbiased coin with $\{0, 1\}$ and obtains one bit $\omega \in \{0, 1\}$. $\mathcal{B}$ chooses a set of attributes $\mathbf{A}_C$ with $\mathbf{A}^* \subseteq \mathbf{A}_C$, and computes

$$C_0 = Z \cdot M_\omega,\ C_1 = C \text{ and } C_2 = C^\gamma \cdot C^{\sum_{a_{\lambda_j} \in \mathbf{A}_C - \mathbf{A}^*} e_{\lambda_j}}.$$

$\mathcal{B}$ responds $\mathcal{A}$ with the challenged ciphertext $CT^* = (C_0, C_1, C_2)$. So, whenever $Z = e(g, g)^{abc}$, $CT^* = (Z \cdot M_\omega,\ g^c,\ (hT_i \prod_{j=1}^{\pi} T_{\lambda_j})^c)$ is a valid ciphertext of M_ω.

Phase 2. Phase 1 is repeated.

Guess. $\mathcal{A}$ outputs his guess ω' on ω. If $\omega' = \omega$, $\mathcal{B}$ outputs $\mu' = 0$. Otherwise, $\mathcal{B}$ outputs $\mu' = 1$.

As shown above, the public parameters and secret keys generated in the simulation paradigm are identical to those in the real protocol. Now, we compute the advantage with which $\mathcal{B}$ can break the DBDH assumption.

If $\mu = 0$, $CT^* = (C_0, C_1, C_2)$ is a valid ciphertext of M_ω. Therefore, $\mathcal{A}$ can output $\omega' = \omega$ with advantage at least $\epsilon(\ell)$, namely $\Pr[\omega' = \omega | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$. Since $\mathcal{B}$ guesses $\mu' = 0$ when $\omega' = \omega$, we have $\Pr[\mu' = \mu | \mu = 0] \geq \frac{1}{2} + \epsilon(\ell)$.

If $\mu = 1$, $\mathcal{A}$ cannot obtain any information about ω'. Therefore, $\mathcal{A}$ can output $\omega' \neq \omega$ with no advantage, namely $\Pr[\omega' \neq \omega | \mu = 1] = \frac{1}{2}$. Since $\mathcal{B}$ guesses $\mu' = 1$ when $\omega' \neq \omega$, we have $\Pr[\mu' = \mu | \mu = 1] = \frac{1}{2}$.

Thereafter, the advantage with which $\mathcal{B}$ can break the DBDH assumption is $\left|\frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2}\right| \geq \frac{1}{2} \times (\frac{1}{2} + \epsilon(\ell)) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} = \frac{\epsilon(\ell)}{2}$.

□

Comparison. We compare the computation cost of our scheme with that of previous schemes in Table 8.3. By $|\mathbb{U}|$, $|\mathbf{A}_\mathcal{U}|$ and $\mathbf{A}_C$, we denote the number of the universal attributes, the number of the attributes held by a user $\mathcal{U}$ and the number listed in the ciphertext, respectively. In ABE schemes, some important properties should be considered, including types, access structure, security model and the length of ciphertext. In a CP-ABE scheme, the encryptor can determine the access policy; while, in a KP-ABE scheme, the access policy is determined by the CA. Generally, non-monotonic access structures can express more complex access policy than monotonic access structures. ABE scheme which can be proven in the full security model are more secure than those which can be prove in selective-set model. ABE schemes with constant sized ciphertext are efficient than those with ciphertext size linear in the number of required attributes. We compare these properties in our scheme with those in previous schemes in Table 8.4.

8.4 Attribute-Based Oblivious Access Control

In this section, we propose an ABOAC scheme based on the CP-ABE with constant cost in Figure 8.1. In our ABOAC scheme, both the actions performed by a requester and the attributes of him can be protected, namely the requester does not release anything about the content of the selected services and his attributes to the service

provider, even the number and supersets of his attributes. The service provider only knows the number of services accessed by an authorized requester. We describe our ABOAC scheme in Figure 8.2.

Overview. Our idea is that we introduce the proposed ABAC scheme to an OT scheme. At the beginning, the requester authenticates himself to the issuer, and obtains the secret keys for his attributes. Then, the service provider commits all messages under different attributes using the OT technique. Finally, the requester interacts with the service provider adaptively and obtains the intended services. We claim that the requester does not release anything about the selected services and his attributes to the service provider, even the number and a superset of his attributes. This is because all services are encrypted under different attributes by the service provider, but he cannot know which services the requester selected. So, he cannot conclude anything about the requester's attributes from the selected services.

Correctness. From equations (8.1), (8.2) and (8.3) in section 8.3, we have

$$\begin{aligned}\Gamma_z =& \frac{e(C_{\sigma_{z_1}}, Y_z)}{e(D_{\mathcal{R},1}, C_{\sigma_{z_2}})} \\ =& \frac{e(g_1, g_2)^{\omega_{\sigma_z}} \cdot e(g^{\omega_{\sigma_z}}, (h\prod_{a_{c_i}\in \mathbf{A}_{C_{\sigma_z}}})^{s_r})}{e(g^{s_r}, (h\prod_{a_{c_i}\in \mathbf{A}_{C_{\sigma_z}}})^{\omega_{\sigma_z}})} \\ =& e(g_1, g_2)^{\omega_{\sigma_z}},\end{aligned} \tag{8.5}$$

$$\Upsilon_z = \Gamma_z^{x_z} = e(g_1, g_2)^{\omega_{\sigma_z} x_z}, \tag{8.6}$$

$$\Phi_z = \Upsilon_z^{\vartheta} = e(g_1, g_2)^{\vartheta\omega_{\sigma_z} x_z}, \tag{8.7}$$

$$\Psi_z = \Phi_z^{x_z^{-1}} = e(g_1, g_2)^{\vartheta\omega_{\sigma_z}}, \tag{8.8}$$

and

$$\frac{C_{\sigma_{z_0}}}{\Psi_z} = \frac{e(g_1, g_2)^{\vartheta\omega_{\sigma_z}} \cdot M_{\sigma_z}}{e(g_1, g_2)^{\vartheta\omega_{\sigma_z}}} = M_{\sigma_z} \tag{8.9}$$

Theorem 8.2 *Our attribute-based oblivious access control scheme is unconditionally requester-secure.*

Proof: For any Υ_j received by $\mathcal{SP}$ from $\mathcal{R}$, there exists an x_i such that

$$\Upsilon_j = e(g_1, g_2)^{\omega_{\sigma_j} x_j} = e(g_1, g_2)^{\omega_{\sigma_i} x_i} = \Upsilon_i,$$

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}, \mathbb{G}_\tau)$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let g, g_2, h be the generators of the group $\mathbb{G}$. Suppose that the set of universal attributes is $\mathbb{U} = \{a_1, a_2, \cdots, a_n\} \subseteq \{0,1\}^n$.

The issuer $\mathcal{I}$ generates his master secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (\alpha, g_1)$, where $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$ and $g_1 = g^\alpha$. For each $a_j \in \mathbb{U}$, $\mathcal{I}$ chooses $T_j \overset{R}{\leftarrow} \mathbb{G}$. The public parameters are $(e, p, \mathbb{G}, \mathbb{G}_\tau, g, g_1, g_2, h, T_1, T_2, \cdots, T_n)$.

The service provider $\mathcal{SP}$ generates his secret-public pair $\mathcal{KG}(1^\ell) \rightarrow (\vartheta, \mathfrak{g})$, where $\vartheta \overset{R}{\leftarrow} \mathbb{Z}_p$ and $\mathfrak{g} = e(g_1, g_2)^\vartheta$.

KeyGen. To generate a secret key for a requester $\mathcal{R}$ with a set of attributes $\mathbf{A}_\mathcal{R}$, $\mathcal{I}$ chooses $s_r \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$D_{\mathcal{R},1} = g^{s_r},\ D_{\mathcal{R},2} = g_2^\alpha h^{s_r} \text{ and } \{Y_{i_j} = T_{i_j}^{s_r}\}_{a_{ij} \in \mathbf{A}_\mathcal{U}}.$$

The secret key for $\mathcal{R}$ is $SK_\mathcal{R} = (D_{\mathcal{R},1},\ D_{\mathcal{R},2},\ \{Y_{i_j}\}_{a_{i_j} \in \mathbf{A}_\mathcal{R}})$. It can be verified as follows:

$$e(g, D_{\mathcal{R},2}) \overset{?}{=} e(g_1, g_2) \cdot e(D_{\mathcal{R},1}, h) \text{ and } e(g, Y_{i_j}) = e(D_{\mathcal{R},1}, T_{i_j}) \text{ for } a_{i_j} \in \mathbf{A}_\mathcal{R}.$$

Commitment. Suppose that $\mathcal{SP}$ manages m messages $\mathbf{M} = \{M_1, M_2, \cdots, M_m\} \in \mathbb{G}_\tau^m$. Let $\mathbb{A}_j$ be a monotonic access structure and $\mathbf{A}_{C_j} \in \mathbb{A}_j$ be the minimal set in $\mathbb{A}_j$ [DT07][a]. To commit a message M_j under the set of attributes $\mathbf{A}_{C_j}$, $\mathcal{SP}$ chooses $\omega_j \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$C_{j0} = e(g_1, g_2)^{\vartheta\omega_j} \cdot M_j,\ C_{j1} = g^{\omega_j} \text{ and } C_{j2} = (h \prod_{a_{c_t} \in \mathbf{A}_{C_j}} T_{c_t})^{\omega_j}.$$

$\mathcal{SP}$ publishes the ciphertext $\{CT_1, CT_2, \cdots, CT_m\}$, where $CT_j = (C_{j0}, C_{j_1}, C_{j_2})$ for $j = 1, 2, \cdots, m$.

Transfer. A requester $\mathcal{R}$ adaptively chooses $\sigma_z \in \{1, 2, \cdots, m\}$, and computes $Y_z = D_{\mathcal{R},2} \cdot \prod_{a_{c_i} \in \mathbf{A}_{C_{\sigma_z}}} Y_{c_i}$, where $Y_{c_i} \in SK_\mathcal{R}$ and $\mathbf{A}_{C_{\sigma_z}} \subseteq \mathbf{A}_\mathcal{R} \in \mathbb{A}_{\sigma_z}$. He computes $\Gamma_z = \frac{e(C_{\sigma_{z_1}}, Y_z)}{e(D_{\mathcal{R},1}, C_{\sigma_{z_2}})}$. $\mathcal{R}$ chooses $x_z \overset{R}{\leftarrow} \mathbb{Z}_p$, and computes $\Upsilon_z = \Gamma_z^{x_z}$, for $z = 1, 2, \cdots, k$.

1. $\mathcal{R} \xrightarrow{\Upsilon_z} \mathcal{SP}$. $\mathcal{R}$ sends Υ_z to $\mathcal{S}$.
2. $\mathcal{R} \xleftarrow{\Phi_z} \mathcal{SP}$. $\mathcal{SP}$ computes $\Phi_z = \Upsilon_z^\vartheta$, and responds $\mathcal{R}$ with Φ_z.
3. $\mathcal{R}$ computes $\Psi_z = \Phi_z^{x_z^{-1}}$ and $M_{\sigma_z} = \frac{C_{\sigma_{z0}}}{\Psi_z}$, for $z = 1, 2, \cdots, k$.

[a] By $\mathbf{A}_{C_j} \in \mathbb{A}_j$ be the minimal set of $\mathbb{A}_j$, we mean that $\mathbf{A}_{C_j} \subseteq \mathbf{S}$ if $\mathbf{S} \in \mathbb{A}_j$.

Figure 8.2: ABOAC: Attribute-based Oblivious Access Control

namely $x_i = \frac{\omega_{\sigma_j} x_j}{\omega_{\sigma_i}} \pmod p$. Therefore, from the view of $\mathcal{SP}$, whether Υ_j is computed from C_{σ_j} or C_{σ_i} is identically distributed. So, our ABOAC scheme is unconditionally requester-secure.

□

Theorem 8.3 *Our attribute-based oblivious access control scheme is service provider secure if the extended chosen-target computational Diffie-Hellman assumption holds in the group* $\mathbb{G}_\tau$.

Proof: For any PPT adversary $\mathcal{R}^*$ in the real world, we will show that there exists a PPT adversary $\hat{\mathcal{R}}^*$ in the ideal world such that the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable. The real world and ideal world paradigms are processed as follows:

1. $\mathcal{SP}$ sends all his messages $\{M_1, M_2, \cdots, M_m\}$ to a trusted third party TTP.
2. $\hat{\mathcal{R}}^*$ sends $\{CT_1^*, CT_2^*, \cdots, CT_m^*\}$ to TTP, where $CT_i^* \stackrel{R}{\leftarrow} \mathbb{G}_\tau \times \mathbb{G}^2$.
3. $\hat{\mathcal{R}}^*$ monitors the outputs of $\mathcal{R}^*$. If $\mathcal{R}^*$ can output $(\Gamma_1, \Upsilon_1), (\Gamma_2, \Upsilon_2), \cdots, (\Gamma_k, \Upsilon_k)$, $\hat{\mathcal{R}}^*$ outputs $(\Gamma_1^*, \Upsilon_1^*), (\Gamma_2^*, \Upsilon_2^*), \cdots, (\Gamma_k^*, \Upsilon_k^*)$, where $(\Gamma_j^*, \Upsilon_j^*) \stackrel{R}{\leftarrow} \mathbb{G}^2$, for $j = 1, 2, \cdots, k$.
4. When $\mathcal{R}^*$ submits $\{\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_k\}$ to obtain $\{\Phi_1, \Phi_2, \cdots, \Phi_k\}$, $\hat{\mathcal{R}}^*$ queries the help oracle $H_{\mathbb{G}_\tau}(\cdot)$ on $\{\Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*\}$, and gets back with $\{\Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*\}$, where $\Phi_j^* = (\Upsilon_j^*)^{\vartheta^*}$, for $j = 1, 2, \cdots, k$.
5. If $\mathcal{R}^*$ can output Ψ_j, $\hat{\mathcal{R}}^*$ sends σ_j to the TTP. TTP responds $\hat{\mathcal{R}}^*$ with $\frac{C_{\sigma_{j_0}}^*}{M_{\sigma_j}}$.
6. $\hat{\mathcal{R}}^*$ outputs $\{\Gamma_1^*, \Gamma_2^*, \cdots, \Gamma_k^*, \Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*, \Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*, CT_1^*, CT_2^*, \cdots, CT_m^*\}$.

If $\mathcal{R}$ obtains $k+1$ messages and $\hat{\mathcal{R}}^*$ does not know which k indices have been selected by $\mathcal{R}^*$, the simulation fails. Otherwise, we will show that $\mathcal{R}^*$ can obtain no more than k messages under the XCT-CDH assumption. If $\mathcal{R}$ can obtain $k+1$ messages, he can compute Ψ_j, for $j = 1, 2, \cdots, k+1$. Therefore, after receiving $(e(g_1, g)^{\omega_{\sigma_1}})^{\vartheta}, (e(g_1, g)^{\omega_{\sigma_2}})^{\vartheta}, \cdots, (e(g_1, g)^{\omega_{\sigma_k}})^{\vartheta}$, $\mathcal{R}$ can compute $(e(g_1, g)^{\omega_{\sigma_{k+1}}})^{\vartheta}$. This contradicts the XCT-CDH assumption. So, $\mathcal{R}$ can obtain at most k messages.

$\{\Gamma_1, \Gamma_2, \cdots, \Gamma_k\}$ and $\{\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_k\}$ are random elements in $\mathbb{G}_\tau$. $\{CT_1, CT_2, \cdots, CT_m\}$ are random elements in $\mathbb{G}_\tau \times \mathbb{G}^2$. $\{\Phi_1, \Phi_2, \cdots, \Phi_k\}$ and $\{\Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*\}$ are identically distributed.

Hence, the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

□

Theorem 8.4 *Our attribute-based oblivious access control scheme is semantically secure if the extended chosen-target Diffie-Hellman assumption holds in the group* $\mathbb{G}_\tau$.

Proof: There are two kinds of adversaries:

- Type-I. The adversary can compute Γ_j from (C_{j_1}, C_{j_2}), then acts as a legal requester to interact with the service provider.
- Type-II. The adversary can compute M_j from $CT_j = (C_{j_0}, C_{j_1}, C_{j_2})$.

We will show that Type-I adversary can be used to break the IND-CPA security of the CP-ABE scheme proposed in Figure 8.1, and Type-II adversary can be used to break the XCT-CDH assumption.

Type-I. Suppose that $\mathcal{A}$ is a Type-I adversary who can compute Γ_j from (C_{j_1}, C_{j_2}). We can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the IND-CPA security of the CP-ABE in Figure 8.1. Suppose that M_j is encrypted under the same set of attributes $\mathbf{A}_{C_j}$ in the proposed CP-ABE scheme and the ciphertext is $CT'_j = (C'_{j_0}, C'_{j_1}, C'_{j_2})$, where $C'_{j_i} = C_{j_i}$, for $i = 1, 2$. $\mathcal{B}$ sends (C'_{j_1}, C'_{j_2}) to $\mathcal{A}$. If $\mathcal{A}$ can compute Γ_j, $\mathcal{B}$ can compute $M_j = \frac{C'_{j_0}}{\Gamma_j}$. This contradicts Theorem 8.1.

Type-II. Suppose that $\mathcal{A}$ is a Type-II adversary who can compute the M_j from the commitment $CT_j = (C_{j_0}, C_{j_1}, C_{j_2})$. We can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the XCT-CDH assumption as follows. Given $e(g_1, g_2)^\vartheta$, $e(g_1, g)^{\omega_j}$, $e(g_1, h \cdot \prod_{a_{c_i} \in \mathbf{A}_{C_j}} T_{c_i})^{\omega_j}$, the aim of $\mathcal{B}$ is to compute $(e(g_1, g_2)^\vartheta)^{\omega_j}$. $\mathcal{B}$ sends $CT_j = (C_{j_0}, C_{j_1}, C_{j_2})$ to $\mathcal{A}$. If $\mathcal{A}$ can output M_j, $\mathcal{B}$ can compute $(e(g_1, g_2)^\vartheta)^{\omega_j} = \frac{C_{j_0}}{M_j}$. So $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption.

□

Complexity. We list the computation cost and communication cost of our ABOAC scheme in Table 8.3 and Table 8.4, respectively. By $|\mathbf{A}_\mathcal{U}|$, we denote the number of attributes held by a user $\mathcal{U}$.

8.5 Chapter Summary

In this chapter, we first proposed a CP-ABE scheme where both the computation cost and the communication cost are constant. Both the encryption and the decryption algorithms in our ABE scheme are very efficient. Then, we proposed an ABOAC scheme by introducing the proposed CP-ABE scheme to an OT scheme. In our ABOAC scheme, both the attributes of the requester and the actions performed by him can be hidden. The requester does not release anything about the selected services and his attributes to the service provider, even the number and a superset of his attributes. The service provider only knows the number of the services accessed by an authorized requester. Hence, our ABOAC scheme provides an intuitive and novel solution to privacy-preserving ABAC schemes. Note that the computing cost and communication cost in our ABOAC scheme are constant and independent of the required attributes. So, our ABOAC can be exploited in the systems with limited computing and communication ability, such as WSANs, MANETS, *etc.*

Table 8.1: The Comparison of Computation Cost

Schemes	Setup	Key Generation	Encryption	Decryption
SW [SW05]	$(\|\mathbb{U}\|+1)T_e$	$\|\mathbf{A}_{\mathcal{U}}\|T_e$	$\|\mathbf{A}_C\|T_e$	$\|\mathbf{A}_C\|T_e+\|bA_C\|T_p$
GPSW [GPSW06]	$(\|\mathbb{U}\|+1)T_e$	$\|\mathbf{A}_{\mathcal{U}}\|T_e$	$\|\mathbf{A}_C\|T_e$	$\|\mathbf{A}_C\|T_e+\|bA_C\|T_p$
OSW [OSW07]	$2(\|\mathbb{U}\|+1)T_e$	$3\|\mathbf{A}_{\mathcal{U}}\|T_e$	$2(\|\mathbf{A}_C\|+1)T_e$	$\|\mathbf{A}_C\|T_e+\|\mathbf{A}_C\|T_p$
BSW [BSW07]	$3T_e$	$2(\|\mathbf{A}_{\mathcal{U}}+1\|)T_e$	$2(\|(\mathbf{A}_C\|+1)\|T_e$	$\|\mathbf{A}_C\|T_e+\|\mathbf{A}_C\|T_P$
CN [CN07]	$(3\|\mathbb{U}\|+1)T_e$	$(\|\mathbb{U}\|+\|\mathbf{A}_{\mathcal{U}}\|)T_e$	$(\|\mathbb{U}\|+2)T_e$	$\|\mathbb{U}\|T_p$
EMONS [EMN$^+$09]	$(\|\mathbb{U}\|+1)T_e$	$4T_e$	$3T_e$	$2T_p$
ZH [ZH10]	$6\|\mathbb{U}\|E$	$(\|\mathbf{A}_{\mathcal{U}}\|+1)T_e$	$3T_e$	$(2\|\mathbf{A}_C\|+1)T_p$
HLR [HLR10]	$2(\|\mathbb{U}+1\|)T_e$	$(\|\mathbb{U}\|+\|\mathbf{A}_{\mathcal{U}}\|)T_e$	$3T_e$	$(\frac{\|\mathbf{A}_C\|(\|\mathbf{A}_C\|-1)}{2}+2)T_e+3T_p$
LOSTW [LOS$^+$10]	$(\|\mathbb{U}\|+2)T_e$	$(\|\mathbf{A}_U\|+2)T_e$	$(3\|\mathbf{A}_C\|+2)T_e$	$\|\mathbf{A}_C\|T_e+(2\|\mathbf{A}_C\|+1)T_p$
Waters [Wat11]	$3T_e$	$(\|\mathbf{A}_{\mathcal{U}}\|+2)T_e$	$2(\|\mathbf{A}_C\|+1)T_e$	$\|\mathbf{A}_C\|T_e+(2\|\mathbf{A}_C\|+1)T_p$
ALP [ALP11]	$(2\|\mathbb{U}\|+1)T_e$	$(5\|\mathbf{A}_{\mathcal{U}}\|-2)T_e$	$4T_e$	$(2\|\mathbb{U}\|-1)T_e+2\|\mathbf{A}_C\|T_p$
CZF [CZF11]	$2\|\mathbb{U}\|(T_e+T_p)$	$\|\mathbf{A}_{\mathcal{U}}\|T_e$	$3T_e$	$2T_p$
Our scheme	$2T_e$	$(\|\mathbf{A}_{\mathcal{U}}\|+2)T_e$	$3T_e$	$2T_p$

Table 8.2: The Comparison of Type, Access Structure, security Model and The Length of Ciphertext

Schemes	KP/CP-ABE	Access Structure	Security Model	Length of Ciphertext
SW [SW05]	KP-ABE	monotonic	selective-set	$\|\mathbf{A}_C\|E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
GPSW [GPSW06]	KP-ABE	monotonic	selective-set	$\|\mathbf{A}_C\|E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
OSW [OSW07]	KP-ABE	non-monotonic	selective-set	$(\|\mathbf{A}_C\|+1)E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
BSW [BSW07]	CP-ABE	monotonic	full security	$(\|\mathbf{A}_C\|+2)E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
CN [CN07]	CP-ABE	non-monotonic	selective-set	$(\|\mathbf{A}_C\|+1)E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
EMONS [EMN$^+$09]	CP-ABE	monotonic	selective-set	$2E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
ZH [ZH10]	CP-ABE	non-monotonic	selective-set	$2E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
HLR [HLR10]	CP-ABE	monotonic	selective-set	$2E_{\mathbb{G}}+2E_{\mathbb{G}_\tau}$
LOSTW [LOS$^+$10]	CP-ABE	monotonic	full security	$(2\|\mathbf{A}_C\|+1)E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
Waters [Wat11]	CP-ABE	monotonic	selective-set	$(2\|\mathbf{A}_C\|+1)E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
ALP [ALP11]	KP-ABE	non-monotonic	selective-set	$3E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
CZF [CZF11]	KP-ABE	non-monotonic	selective-set	$2E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$
Our scheme	CP-ABE	monotonic	selective-attribute	$2E_{\mathbb{G}}+E_{\mathbb{G}_\tau}$

Table 8.3: The Computation Cost of Our ABOAC Scheme

Scheme	Computation Cost					
	Setup	KeyGen		Commitment	Transfer	
	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{SP}$	$\mathcal{R}$	$\mathcal{SP}$
ABOAC	$2T_e$	$(\lvert\mathbf{A}_{\mathcal{U}}\rvert + 2)T_e$	$(2(\lvert\mathbf{A}_{\mathcal{U}}\rvert + 1)T_p$	$3mT_e$	$2kT_e + 2 + kT_p$	kT_e

Table 8.4: The Communication Cost of Our ABOAC Scheme

Scheme	Communication cost			
	Key Generation	Commitment	Transfer	
	$\mathcal{I} \rightarrow \mathcal{R}$	$\mathcal{SP} \rightarrow \mathcal{R}$	$\mathcal{R} \rightarrow \mathcal{SP}$	$\mathcal{R} \leftarrow \mathcal{SP}$
ABOAC	$(\lvert\mathbf{A}_{\mathcal{U}}\rvert + 2)E_{\mathbb{G}}$	$2mE_{\mathbb{G}} + mE_{\mathbb{G}_\tau}$	$kE_{\mathbb{G}_\tau}$	$kE_{\mathbb{G}_\tau}$

Chapter 9

Efficient Oblivious Transfer with Access Control

In this chapter, we proposed two efficient oblivious transfer with access control (AC-OT) schemes by introducing the oblivious signature-based envelope (OSBE) technique to an oblivious transfer scheme. Parts of this work appeared in [HSMY12b].

9.1 Introduction

Security and privacy problems have been major concerns to Internet users. For example, users might be worried about whether their personally identifiable information (PII) are illegally collected, pilfered and disseminated. A small part of PII is insufficient for identifying the real identity of a user, but the malicious attackers can aggregate the collected partial PII, such as health condition, financial data and hobbies, to analyse and trace the real user. Lessons from identity fraud, identity theft, fictitious identity, *etc.* [KL06] suggest that PII should be released under the user's control.

Obviously, there is a trade-off between the accountability and privacy. Solution towards to balance the trade-off have been proposed, such as identity management [CGS06, CP07], user-centric system [BSCGS06], privacy-preserving systems [AGK03], anonymous credential [Cha85], hidden credentials [HBSO03], k-time anonymous authentication [TFS04, ASM06]. In these systems, the security of users' PII are considered. However, in practice, adversaries can trace and identify a user not only by his PII, but also by the actions performed by him, such as the websites he visited frequently, the goods he purchased online. Therefore, in order to protect users' privacy, a new system should be proposed to secure their PII and the actions performed by them. Suppose that there exists a trusted third party (TTP) called the issuer, who is trusted by all participants in the system. Prior to accessing the

services, a user needs to authenticate himself to the issuer and obtain the required credentials from him. Then, the user can use the received credentials to access the protected services, without revealing any information about his choices and PII to the service providers.

9.1.1 Related Work

The introduction of oblivious transfer (OT) and oblivious transfer with access control (AC-OT) can be found in Section 8.1.1.

Proposed by Li, Du and Boneh [LDB03], oblivious signature-based envelope (OSBE) is a cryptographic protocol, where a receiver can obtain the secret encapsulated in an envelope by the sender if and only if he has possessed a signature from the issuer on a public message, such as the address on the envelope. Furthermore, the receiver is not required to authenticate himself to the sender. As a result, the sender cannot distinguish a receiver who has possessed a credential from the receiver who has not possessed a credential. Therefore, the signature works as a hidden credential [HBSO03]. Notably, the sender in an OSBE scheme cannot control the interaction as he must encrypt the secret under the parameters obtained from the issuer, instead of using his secret key.

Nasserian and Tsudik [NT06] revisited OSBE schemes and pointed some applications.

9.1.2 Our Contribution

In this chapter, we propose two novel and efficient AC-OT schemes. In our schemes, only the authorized requester can obtain services from the service provider obliviously. The service provider knows the number of the services accessed by an authorized user and nothing about the content of the accessed services. Furthermore, a requester is not required to authenticate himself to the service provider. Therefore, the requester releases nothing about his PII to the service provider. Notably, our schemes *do not* require any zero knowledge proof, and hence, our scheme is more efficient than the previous schemes.

9.1.3 Chapter Organization

The remainder of this chapter is organized as follows. In Section 9.2, the formal definition and security model of AC-OT schemes are described. We propose two efficient AC-OT schemes in Section 9.3. Section 9.4 summarizes the chapter.

9.2 Formal Definition and Security Model

In this section, we introduce the formal definition and security model of AC-OT schemes.

9.2.1 Formal Definition

There are three entities in an AC-OT scheme: issuer $\mathcal{I}$, service provider $\mathcal{SP}$ and requester $\mathcal{R}$. $\mathcal{I}$ authenticates the requesters, and issues credentials to them. $\mathcal{SP}$ interacts with $\mathcal{R}$, and sends the selected services to him. $\mathcal{R}$ obtains credentials from $\mathcal{I}$, and interacts with $\mathcal{SP}$ to obtain the intended services. An AC-OT scheme consists of the following algorithms:

$\mathsf{Setup}(1^\ell) \rightarrow (params, (ISK, IPK), (SSK, SPK))$. The setup algorithm takes as input 1^ℓ, and outputs the public parameters $params$ and secret-public key pairs $\mathcal{KG}(1^\ell) \rightarrow (ISK, IPK)$ and $\mathcal{KG}(1^\ell) \rightarrow (SSK, SPK)$ for $\mathcal{I}$ and $\mathcal{SP}$, respectively.

$\mathsf{Issue}(params, SI, RI, ISK) \rightarrow \delta_{\mathcal{U}}$. The issue algorithm takes as input the public parameters $params$, $\mathcal{SP}$'s identifier SI, $\mathcal{R}$'s identifier RI and the secret key ISK, and outputs a credential $\sigma_{\mathcal{U}}$ for $\mathcal{U}$.

$\mathsf{Commit}(params, IPK, SSK, M_i) \rightarrow CT_i$. Suppose that $\mathcal{SP}$ manages m messages $M_1, M_2, \cdots, M_m$. To commit a message M_i, the commitment algorithm takes as input the public parameters $params$, the public key ISK, the secret key SSK and the message M_i, and outputs a ciphertext CT_i which can be decrypted by a requester $\mathcal{R}$ who obtains a signature $\delta_{\mathcal{R}}$ on the identifier of $\mathcal{SP}$, for $i = 1, 2, \cdots, m$.

$\mathsf{Transf}(\mathcal{SP}(params, SSK) \leftrightarrow \mathcal{R}(params, SPK, \mathbf{C}_{\mathcal{R}}, \delta_{\mathcal{R}})) \rightarrow (\perp, \mathbf{M}_{\mathcal{R}})$. The transfer algorithm is an interactive algorithm executed between $\mathcal{SP}$ and $\mathcal{R}$. $\mathcal{SP}$ takes as input the public parameters $params$ and his secret key SSK, and

outputs nothing. $\mathcal{R}$ takes as input the public parameters $params$, the public key SPK, a set of choices $\mathbf{C}_{\mathcal{R}} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ and his credential $\delta_{\mathcal{R}}$, and outputs messages $\mathbf{M}_{\mathcal{R}} = \{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$.

Definition 9.1 *We say that an oblivious transfer with access control scheme is correct if*

$$\Pr\left[\begin{array}{l|l} & \mathsf{Setup}(1^\ell) \rightarrow (params, (ISK, IPK), (SSK, SPK)); \\ \mathsf{Transf}(\mathcal{SP}(\boxtimes) \leftrightarrow & \mathsf{Issue}(params, SI, RI, ISK) \rightarrow \delta_{\mathcal{R}}; \\ \mathcal{R}(\boxplus)) \rightarrow (\perp, \mathbf{M}) & \mathsf{Commit}(params, IPK, SSK, M_i) \rightarrow CT_i; \\ & for\ j = 1, 2, \cdots, k \end{array}\right] = 1$$

where the probability is taken over the random coins consumed by the algorithms in the scheme, $\boxtimes = (params, SSK)$, $\boxplus = (params, SPK, \mathbf{C}_{\mathcal{R}}, \delta_{\mathcal{R}})$, $\mathbf{C}_{\mathcal{R}} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ *and* $\mathbf{M}_{\mathcal{R}} = \{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$.

9.2.2 Security Model

The security model of AC-OT schemes is similar to that for ABOAC schemes in Section 8.2.4. It is defined as follows.

For the privacy of $\mathcal{R}$, his choices should be unconditionally secure and his credentials are not exposed to $\mathcal{SP}$. For the security of $\mathcal{SP}$, a real world paradigm and an ideal world paradigm are exploited. If there exists an adversary in the real world, there will exist an adversary in the ideal world such that the outputs of these two adversaries are indistinguishable. We call this model as half-simulation model, which is similar to that in [NP99a, NP99b].

Privacy of Requester. An AC-OT scheme can protect the privacy of $\mathcal{R}$ if it can provide the following properties:

1. $\mathcal{R}$ does not release anything about his PII to $\mathcal{SP}$.

2. For any two different choice sets $\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ and $\{\sigma'_1, \sigma'_2, \cdots, \sigma'_k\}$, the transcripts received by $\mathcal{SP}$ corresponding to $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$ and $\{M_{\sigma'_1}, M_{\sigma'_2}, \cdots, M_{\sigma'_k}\}$ are indistinguishable. Especially, the choices of $\mathcal{R}$ are unconditionally requester secure if $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$ and $\{M_{\sigma'_1}, M_{\sigma'_2}, \cdots, M_{\sigma'_k}\}$ are identically distributed.

Security of Service Provider. Suppose that $\mathcal{R}$ has obtained the required credentials. To define the security of $\mathcal{SP}$, we compare a real world experiment and an ideal world experiment. In the real world experiment, $\mathcal{R}$ and $\mathcal{SP}$ execute the protocol. Meanwhile, in the ideal world experiment, the functionality of the protocol is replaced by a trusted third party (TTP). $\mathcal{SP}$ sends all his messages $\{M_1, M_2, \cdots, M_m\}$ to the TTP. $\mathcal{R}$ adaptively submits his choices $\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ to the TTP. If $\sigma_1, \sigma_2, \cdots, \sigma_k \in \{1, 2, \cdots, m\}$, the TTP responds $\mathcal{R}$ with $\{M_{\sigma_1}, M_{\sigma_2}, \cdots, M_{\sigma_k}\}$. An AC-OT scheme is service provider secure, if for any malicious requester $\mathcal{R}^*$ in the real world, there exists a requester $\hat{\mathcal{R}}^*$ in the ideal world such that the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

Semantic security. An adversary cannot obtain anything about the messages managed by the service provider if he has not obtained a credential on the identifier of the service provider from the issuer.

Definition 9.2 *We say an oblivious transfer with access control scheme is secure if it can protect the requester's privacy, and is service provider secure and semantically secure.*

9.3 Oblivious Transfer with Access Control

In this section, we propose two efficient AC-OT schemes. The first one is very simple, while the credentials of the requester are transferable. Being different from the first scheme, the second scheme sacrifices a little efficiency, while the credentials of the requester are *all-or-nothing nontransferable*, namely all the credentials of the requester are shared if he shares one with others [CL01].

Overview. Our idea is as follows. At first, the requester interacts with the issuer to obtain a credential which is a signature on a public message, for example the identifier of the service provider in the trusted circle[1]. Then, the service provider encrypts his messages using the OSBE technique under the public message and *his secret key*. Finally, the requester interacts with the service provider, decrypts the ciphertexts using his credential, and obtains the intended messages. In our schemes, only the authorized requester can obtain services from the service provider obliviously, while he is not required to authenticate (proof) himself to the service provider in zero-knowledge.

[1]Trusted circle is a domain where all participants trust the issuer.

9.3.1 Oblivious Transfer with Access Control I

In this section, we proposed an efficient AC-OT scheme (AC-OT$^m_{k\times 1}$-I) based on the short signature [BB04b] and the oblivious transfer [CT05]. In our AC-OT$^m_{k\times 1}$-I, only the requester who has possessed the required credential can obtain services from the service provider adaptively, without releasing anything about his PII and the content of the selected services to the service provider. Meanwhile, the service provider only knows the number of the services selected by the authorized requester. Our AC-OT$^n_{k\times 1}$-I is described in Figure 9.1.

Correctness. The correctness of our AC-OT$^m_{k\times 1}$-I is shown as follows. We have

$$\begin{aligned}\Gamma_\varrho &= e(\delta_{\mathcal{R}}, C_{\sigma_{\varrho_1}}) \\ &= e(g^{\frac{1}{x+r}}, (yh^r)^{t_{\sigma_\varrho}}) \\ &= e(g^{\frac{1}{x+r}}, h^{x+r})^{t_{\sigma_\varrho}} \\ &= e(g,h)^{t_{\sigma_\varrho}},\end{aligned}$$

and

$$\begin{aligned}\frac{C_{\sigma_{\varrho 0}}}{\Psi_\varrho} &= \frac{e(g,h)^{zt_{\sigma_\varrho}} \cdot M_{\sigma_\varrho}}{\Phi^{\upsilon_\varrho^{-1}}} \\ &= \frac{e(g,h)^{zt_{\sigma_\varrho}} \cdot M_{\sigma_\varrho}}{\Upsilon_\varrho^{z\upsilon_\varrho^{-1}}} \\ &= \frac{e(g,h)^{zt_{\sigma_\varrho}} \cdot M_{\sigma_\varrho}}{\Gamma_\varrho^{z}} \\ &= \frac{e(g,h)^{zt_{\sigma_\varrho}} \cdot M_{\sigma_\varrho}}{e(g,h)^{zt_{\sigma_\varrho}}} \\ &= M_{\sigma_\varrho}.\end{aligned}$$

Theorem 9.1 *Our oblivious transfer with access control I (AC-OT$^m_{k\times 1}$-I) is unconditionally requester-secure.*

Proof: For any Υ_ϱ received by $\mathcal{SP}$ from $\mathcal{R}$, there exists a $\upsilon_\varphi \in \mathbb{Z}_p$ ($\varphi \neq \varrho$) such that $\Upsilon_\varrho = e(g,h)^{t_{\sigma_\varrho}\upsilon_\varrho} = e(g,h)^{t_{\sigma_\varphi}\upsilon_\varphi} = \Upsilon_\varphi$, namely $\upsilon_\varphi = \frac{t_{\sigma_\varrho}\upsilon_\varrho}{t_{\sigma_\varphi}} \pmod p$.

Hence, from the view of $\mathcal{SP}$, Υ_ϱ is computed from $C_{\sigma_{\varrho_1}}$ or $C_{\sigma_{\upsilon_1}}$ is identically distributed. Therefore, AC-OT$^m_{k\times 1}$-I is unconditionally requester-secure.

□

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, where $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let g and h be the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

The issuer $\mathcal{I}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (x, y)$, where $x \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and $y = h^x$.

The service provider $\mathcal{SP}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (z, Z)$, where $z \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and $Z = e(g, h)^z$.

$\mathcal{I}$ selects $r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ with $z \not\equiv x + r \pmod p$ and $r + x \not\equiv 0, 1 \pmod p$, and assigns r as the identifier of $\mathcal{SP}$ in the trusted circle.

Issue. To issue a credential on the identifer of $\mathcal{SP}$ to a requester $\mathcal{R}$, $\mathcal{I}$ computes $\delta_\mathcal{R} = g^{\frac{1}{x+r}}$. The credential for $\mathcal{R}$ is $(\delta_\mathcal{R}, r)$. It can be verified by checking $e(\delta_\mathcal{R}, yh^r) \stackrel{?}{=} e(g, h)$.

Commitment. Suppose that $\mathcal{SP}$ manages messages $\mathbf{M} = \{M_1, M_2, \cdots, M_m\} \in \mathbb{G}_\tau^m$. To commit a message M_j, $\mathcal{SP}$ selects $t_j \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, and computes

$$C_{j_0} = e(g, h)^{zt_j} \cdot M_j \text{ and } C_{j_1} = (yh^r)^{t_j}.$$

$\mathcal{SP}$ publishes the ciphertexts $\{CT_1, CT_2, \cdots, CT_m\}$, where $CT_j = \{C_{j_1}, C_{j_2}\}$ for $j = 1, 2, \cdots, m$.

Transfer. $\mathcal{R}$ adaptively selects $\sigma_\varrho \in \{1, 2, \cdots, m\}$, and computes $\Gamma_\varrho = e(\delta_\mathcal{R}, C_{\sigma_{\varrho_1}})$. $\mathcal{R}$ chooses $\upsilon_\varrho \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $\Upsilon_\varrho = \Gamma_\varrho^{\upsilon_\varrho}$.

1. $\mathcal{R} \stackrel{\Upsilon_\varrho}{\longrightarrow} \mathcal{SP}$. $\mathcal{R}$ sends Υ_ϱ to $\mathcal{SP}$.
2. $\mathcal{R} \stackrel{\Phi_\varrho}{\longleftarrow} \mathcal{SP}$. $\mathcal{SP}$ computes $\Phi_\varrho = \Upsilon_\varrho^z$, and responds $\mathcal{R}$ with Φ_ϱ.
3. $\mathcal{R}$ computes $\Psi_\varrho = \Phi_\varrho^{\upsilon_\varrho^{-1}}$ and $M_{\sigma_\varrho} = \frac{C_{\sigma_{\rho_0}}}{\Psi_\varrho}$.

Figure 9.1: AC-OT$^m_{k\times 1}$-I: Oblivious Transfer with Access Control I

Theorem 9.2 *Our oblivious transfer with access control I (AC-OT$^M_{k\times 1}$-I) is service provider secure if the extended chosen-target computational Diffie-Hellman assumption holds in* $\mathbb{G}_\tau$.

Proof: For any PPT adversary $\mathcal{R}^*$ in the real world, we will show that there exists a PPT adversary $\hat{\mathcal{R}}^*$ in the ideal world such that the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable. The real world and the ideal world are processed as follows:

1. $\mathcal{SP}$ sends all his messages $\{M_1, M_2, \cdots, M_m\}$ to a trusted third party TTP.
2. $\hat{\mathcal{R}}^*$ sends $\{CT_1^*, CT_2^*, \cdots, CT_m^*\}$ to the TTP, where $CT_i^* = (C_{i_0}^*, C_{i_1}^*) \stackrel{R}{\leftarrow} \mathbb{G}_\tau \times \mathbb{G}_2$, for $i = 1, 2, \cdots, m$.
3. $\hat{\mathcal{R}}^*$ monitors the outputs of $\mathcal{R}^*$. If $\mathcal{R}^*$ can output $(\Gamma_1, \Upsilon_1), (\Gamma_2, \Upsilon_2), \cdots, (\Gamma_k, \Upsilon_k)$, $\hat{\mathcal{R}}^*$ outputs $(\Gamma_1^*, \Upsilon_1^*), (\Gamma_2^*, \Upsilon_2^*), \cdots, (\Gamma_k^*, \Upsilon_k^*)$, where $(\Gamma_i, \Upsilon_i) \stackrel{R}{\leftarrow} \mathbb{G}_\tau^2$, for $i = 1, 2, \cdots, k$.
4. When $\mathcal{R}^*$ submits $(\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_k)$ to obtain $(\Phi_1, \Phi_2, \cdots, \Phi_k)$, $\hat{\mathcal{R}}^*$ queries the help oracle $H_{\mathbb{G}_\tau}(\cdot)$ on $(\Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*)$, and gets back with $(\Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*)$, where $\Phi_i = (\Upsilon_i^*)^{z^*}$, for $i = 1, 2, \cdots, k$.
5. If $\mathcal{R}$ can compute Ψ_ϱ, $\hat{\mathcal{R}}^*$ sends σ_ϱ to the TTP. TTP responds $\hat{\mathcal{R}}^*$ with $\frac{C_{\sigma_{\varrho 0}}^*}{M_{\sigma_\varrho}}$.
6. $\hat{\mathcal{R}}^*$ outputs $(\Gamma_1^*, \Gamma_2^*, \cdots, \Gamma_k^*, \Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*, \Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*, CT_1^*, CT_2^*, \cdots, CT_m^*)$.

If $\mathcal{R}^*$ gets $k+1$ messages and $\hat{\mathcal{R}}^*$ does not know which k indices are really selected by $\mathcal{R}^*$, the simulation fails. Otherwise, we will show that $\mathcal{R}^*$ can obtain at most k messages under the XCT-CDH assumption. If $\mathcal{R}^*$ can get $k+1$ messages, he can compute Ψ_j, for $j = 1, 2, \cdots, k+1$. Namely, after receiving $(e(g,h)^{t_{\sigma_1}})^z, (e(g,h)^{t_{\sigma_2}})^z, \cdots, (e(g,h)^{t_{\sigma_k}})^z$, $\mathcal{R}^*$ can compute $(e(g,h)^{t_{\sigma_{k+1}}})^z$. This contradicts to the XCT-CDH assumption. Hence, $\mathcal{R}^*$ can obtain at most k messages from $\mathcal{SP}$.

$\{\Gamma_1, \Gamma_2, \cdots, \Gamma_k\}$ and $\{\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_k\}$ are random elements in $\mathbb{G}_\tau$. $CT_1, CT_2, \cdots, CT_m$ are random elements in $\mathbb{G}_\tau \times \mathbb{G}_2$. $\{\Phi_1, \Phi_2, \cdots, \Phi_k\}$ and $\{\Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*\}$ are identically distributed. Therefore, the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

□

Theorem 9.3 *Our oblivious transfer with access control I (AC-OT$^m_{k\times 1}$-I) is semantically secure under the q-strong Diffie-Hellman assumption and extended chosen-target computational Deffie-Hellman assumption.*

Proof: There are two types adversaries:

- Type-I: The adversary can compute Γ_ϱ from $C_{\sigma_{\varrho_1}}$, then acts as an authorized requester to interact with the service provider.
- Type-II: The adversary can compute M_{σ_ϱ} from $CT_{\sigma_\varrho} = (C_{\sigma_{\varrho 0}}, C_{\sigma_{\varrho 1}})$.

We will show that a Type-I adversary can be used to break the q-SDH assumption or XCT-CDH assumption and a Type-II adversary can be used to break the XCT-CDH assumption.

Type-I: Suppose that $\mathcal{A}$ is a Type-I adversary who can compute Γ_ϱ from $C_{\sigma_{\varrho_1}}$. We can construct an algorithm $\mathcal{B}$ that can use $\mathcal{A}$ to break the q-SDH assumption or XCT-CDH assumption as follows.

1. If $\mathcal{A}$ can compute the signature (δ, r) with $\delta = g^{\frac{1}{x+r}}$, then obtain $\Gamma_\varrho, \Upsilon_\varrho, \Phi_\varrho$ and Ψ_ϱ, $\mathcal{B}$ can use $\mathcal{A}$ to break the q-SDH assumption[2].
2. If $\mathcal{A}$ can not compute the signature (δ, r), he can compute Γ_ϱ from $C_{\sigma_{\varrho_1}} = (yh^r)^{t_{\sigma_\varrho}}$. If it is, $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption as follows: given $e(g, C_{\sigma_{\varrho_1}}) = (e(g,h)^{x+r})^{t_{\sigma_\varrho}}$ and $e(g,h)$, the aim of $\mathcal{B}$ is to compute $e(g,h)^{t_{\sigma_\varrho}}$. $\mathcal{B}$ sends $C_{\sigma_{\varrho_1}}$ to $\mathcal{A}$, if $\mathcal{A}$ can outputs Γ_j, $\mathcal{B}$ aborts. $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption as $\Gamma_\varrho = e(g,h)^{t_{\sigma_\varrho}}$.

Type-II: Suppose that $\mathcal{A}$ is a Type-II adversary who can compute M_{σ_ϱ} from the ciphertext $CT_{\sigma_\varrho} = (C_{\sigma_{\varrho 0}}, C_{\sigma_{\varrho 1}})$. If it is, $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption as follows: given $(e(g,h)^{x+r})^{t_{\sigma_\varrho}}$ and $e(g,h)^z$, the aim of $\mathcal{B}$ is to compute $(e(g,h)^z)^{t_{\sigma_\varrho}}$. $\mathcal{B}$ sends $CT_{\sigma_\varrho} = (C_{\sigma_{\varrho 0}}, C_{\sigma_{\varrho 1}})$ to $\mathcal{A}$. If $\mathcal{A}$ can output M_{σ_ϱ}, $\mathcal{B}$ aborts. $\mathcal{B}$ can compute $e(g,h)^{zt_{\sigma_\varrho}} = \frac{C_{\sigma_{\varrho 0}}}{M_{\sigma_\varrho}}$. Hence $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption.

Therefore, AC-OT$^m_{k\times 1}$-I is semantically secure.

□

[2]The short signature is existentially unforgeable against the weakly chosen message attacks under the q-SDH assumption [BB04b].

9.3.2 Oblivious Transfer with Access Control II

In this section, we propose an AC-OT scheme (AC-OT$^m_{k\times 1}$-II) based on the signature scheme [ASM06][3] and the oblivious transfer scheme [CT05]. As a result, our AC-OT$^m_{k\times 1}$-II scheme captures the following properties:

1. Zero-knowledge proof is not required.
2. The requester is not required to authenticate himself to the service provider.
3. The service provider knows the number of the services selected by an authorized requester, and nothing about the contents of the selected services.
4. The requester cannot share his credentials with others.

Our AC-OT$^m_{k\times 1}$-II is described in Figure 9.2.

Correctness. The correctness of AC-OT$^m_{k\times 1}$-II is shown as follows. We have

$$\begin{aligned}
\Gamma_\varrho &= e(\delta_{\mathcal{R}}, C_{\sigma_{\varrho_1}}) \\
&= e((g_0 g_1^s g_2^{x_r})^{\frac{1}{x+r}}, (yh^r)^{t_{\sigma_\varrho}}) \\
&= e((g_0 g_1^s g_2^{x_r})^{\frac{1}{x+r}}, h^{x+r})^{t_{\sigma_\varrho}} \\
&= e(g_0 g_1^s g_2^{x_r}, h)^{t_{\sigma_\varrho}} \\
&= e(g_0, h)^{t_{\sigma_\varrho}} \cdot e(g_1, h)^{st_{\sigma_\varrho}} \cdot e(g_2, h)^{x_r t_{\sigma_\varrho}},
\end{aligned}$$

$$\begin{aligned}
\Upsilon_\varrho &= (\frac{\Gamma_\varrho}{C^s_{\sigma_{\varrho_2}} C^{x_r}_{\sigma_{\varrho_3}}})^{\upsilon_\varrho} \\
&= (\frac{e(g_0, h)^{t_{\sigma_\varrho}} \cdot e(g_1, h)^{st_{\sigma_\varrho}} \cdot e(g_2, h)^{x_r t_{\sigma_\varrho}}}{e(g_1, h)^{st_{\sigma_\varrho}} \cdot e(g_2, h)^{x_r t_{\sigma_\varrho}}})^{\upsilon_\varrho} \\
&= e(g_0, h)^{t_{\sigma_\varrho} \upsilon_\varrho},
\end{aligned}$$

$$\Psi_\varrho = \Phi_\varrho^{\upsilon_\varrho^{-1}} = \Upsilon_\varrho^{z\upsilon_\varrho^{-1}} = e(g_0, h)^{zt_{\sigma_\varrho}},$$

and

$$\begin{aligned}
\frac{C_{\sigma_{\varrho_0}}}{\Psi_\varrho} &= \frac{e(g_0, h)^{zt_{\sigma_\varrho}} \cdot M_{\sigma_\varrho}}{e(g_0, h)^{zt_{\sigma_\varrho}}} \\
&= M_{\sigma_\varrho}.
\end{aligned}$$

[3] This signature scheme was proposed by Boneh, Boyen and Shacham [BBS04], and modified by Au, Susilo, and Mu [ASM06].

Setup. This algorithm takes as input 1^ℓ, and outputs a bilinear group $\mathcal{GG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, where $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\tau$ and p is a prime number. Let g_0, g_1, g_2, g_3 be the generators of $\mathbb{G}_1$, and h be the generator of $\mathbb{G}_2$, respectively.

The issuer $\mathcal{I}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (x, y)$, where $x \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and $y = h^x$.

The service provider $\mathcal{SP}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (z, Z)$, where $z \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and $Z = e(g_0, h)^z$.

The requester $\mathcal{R}$ generates his secret-public key pair $\mathcal{KG}(1^\ell) \rightarrow (x_r, y_r)$, where $x_r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and $y_r = g_2^{x_r}$.

$\mathcal{I}$ selects $r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ with $z \not\equiv x + r \pmod p$ and $x + r \not\equiv 0, 1 \pmod p$, and assigns r as the identifier of $\mathcal{SP}$ in the trusted circle.

Issue. To issue a credential on the identifer of $\mathcal{SP}$ to $\mathcal{R}$, $\mathcal{I}$ selects $s \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, and computes $\delta_\mathcal{R} = (g_0 g_1^s g_2^{x_r})^{\frac{1}{x+r}}$. The credential for $\mathcal{R}$ is $(\delta_\mathcal{R}, s, r)$. It can be verified by checking $e(\delta_\mathcal{R}, yh^r) \stackrel{?}{=} e(g_0 g_1^s g_2^{x_r}, h)$.

Commitment. Suppose that $\mathcal{SP}$ manages messages $\mathbf{M} = \{M_1, M_2, \cdots, M_m\} \in \mathbb{G}_\tau^m$. To commit a message M_j, $\mathcal{SP}$ selects $t_j \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, and computes

$$C_{j_0} = e(g_0, h)^{zt_j} \cdot M_j,\ C_{j_1} = (yh^r)^{t_j},\ C_{j_2} = e(g_1, h)^{t_j},\ C_{j_3} = e(g_2, h)^{t_j}.$$

$\mathcal{SP}$ publishes the ciphertexts $\{CT_1, CT_2, \cdots, CT_m\}$, where $CT_j = (C_{j_0}, C_{j_1}, C_{j_2}, C_{j_3})$ for $j = 1, 2, \cdots, m$.

Transfer. $\mathcal{R}$ adaptively selects $\sigma_\varrho \stackrel{R}{\leftarrow} \{1, 2, \cdots m\}$, and computes $\Gamma_\varrho = e(\delta_\mathcal{R}, C_{\sigma_{\varrho_1}})$. $\mathcal{R}$ chooses $\upsilon_\varrho \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $\Upsilon_\varrho = (\frac{\Gamma_\varrho}{C_{\sigma_{\varrho 2}}^s C_{\sigma_{\varrho 3}}^{x_r}})^{\upsilon_\varrho}$ where $i_j \in \{1, 2, \cdots, n\}$ and $j \in \{1, 2, \cdots, k\}$.

1. $\mathcal{R} \xrightarrow{\Upsilon_\varrho} \mathcal{SP}$. $\mathcal{R}$ sends Υ_ϱ to $\mathcal{SP}$.
2. $R \xleftarrow{\Phi_\varrho} S$. $\mathcal{SP}$ computes $\Phi_\varrho = (\Upsilon_\varrho)^z$, and sends Φ_ϱ to $\mathcal{R}$.
3. $\mathcal{R}$ computes $\Psi_\varrho = \Phi_\varrho^{\upsilon_\varrho^{-1}}$ and $M_{\sigma_\varrho} = \frac{C_{\sigma_{\varrho 0}}}{\Psi_\varrho}$.

Figure 9.2: AC-OT$_{k\times 1}^m$-II: Oblivious Transfer with Access Control II

Theorem 9.4 *Our oblivious transfer with access control II (AC-OT$^m_{k\times 1}$-II) is unconditionally requester-secure.*

Proof: For any Υ_ϱ received by $\mathcal{SP}$ from $\mathcal{R}$, there exists an $\upsilon_\varphi \in \mathbb{Z}_p$ ($\varphi \neq \varrho$) such that $\Upsilon_\varrho = e(g,h)^{t_{\sigma_\varrho}\upsilon_\varrho} = e(g,h)^{t_{\sigma_\varphi}\upsilon_\varphi} = \Upsilon_\varphi$, namely $\upsilon_\varphi = \frac{t_{\sigma_\varrho}\upsilon_\varrho}{t_{\sigma_\varphi}} \pmod p$.

Hence, from the view of $\mathcal{SP}$, Υ_ϱ is computed from $C_{\sigma_{\varrho_1}}$ or $C_{\sigma_{\varphi_1}}$ is identically distributed. Therefore, AC-OT$^m_{k\times 1}$-II is unconditionally requester-secure.

□

Theorem 9.5 *Our oblivious transfer with access control II (AC-OT$^m_{k\times 1}$-II) is service provider secure if the extended chosen-target computational Deffie-Hellman assumption holds in $\mathbb{G}_\tau$.*

Proof: For any PPT adversary $\mathcal{R}^*$ in the real world, we will show that there exists a PPT adversary $\hat{\mathcal{R}}^*$ in the ideal world such that the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

1. $\mathcal{SP}$ sends all his messages $\{M_1, M_2, \cdots, M_m\}$ to a trusted third party TTP.
2. $\hat{\mathcal{R}}^*$ sends $\{CT_1^*, CT_2^*, \cdots, CT_m^*\}$ to the TTP, where $CT_j^* = (C_{j_0}^*, C_{j_1}^*, C_{j_2}^*, C_{j_3}^*) \stackrel{R}{\leftarrow} \mathbb{G}_\tau \times \mathbb{G}_2 \times \mathbb{G}_\tau^2$, for $i = 1, 2, \cdots, m$.
3. $\hat{\mathcal{R}}^*$ monitors the outputs of $\mathcal{R}^*$. If $\mathcal{R}^*$ can compute $(\Gamma_1, \Upsilon_1), (\Gamma_2, \Upsilon_2), \cdots, (\Gamma_k, \Upsilon_k)$, $\hat{\mathcal{R}}^*$ chooses $(\Gamma_1^*, \Upsilon_1^*), (\Gamma_2^*, \Upsilon_2^*), \cdots, (\Gamma_k^*, \Upsilon_k^*)$, where $(\Gamma_j^*, \Upsilon_j^*) \stackrel{R}{\leftarrow} \mathbb{G}_\tau^2$, for $j = 1, 2, \cdots, k$.
4. When $\mathcal{R}^*$ submits $(\Upsilon_1, \Upsilon_2, \cdots, \Upsilon_k)$ to obtain $(\Phi_1, \Phi_2, \cdots, \Phi_k)$, $\hat{\mathcal{R}}^*$ queries the help oracle $H_{\mathbb{G}_\tau}(\cdot)$ on $(\Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*)$, and gets back with $(\Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*)$, where $\Phi_j^* = \Upsilon_j^{z^*}$, for $j = 1, 2, \cdots, k$.
5. If $\mathcal{R}^*$ can compute Ψ_ϱ, $\hat{\mathcal{R}}^*$ sends σ_ϱ to the TTP. TTP responds $\mathcal{R}$ with $\frac{C_{\sigma_{\varrho_0}}^*}{M_{\sigma_\varrho}}$.
6. $\hat{\mathcal{R}}^*$ outputs $(\Gamma_1^*, \Gamma_2^*, \cdots, \Gamma_k^*, \Upsilon_1^*, \Upsilon_2^*, \cdots, \Upsilon_k^*, \Phi_1^*, \Phi_2^*, \cdots, \Phi_k^*, CT_1^*, CT_2^*, \cdots, CT_m^*)$.

If $\mathcal{R}^*$ obtains $k+1$ messages and $\hat{\mathcal{R}}^*$ does not know which k indices are really selected by $\mathcal{R}^*$, the simulation fails. Otherwise, we will show that $\mathcal{R}^*$ can get at most k messages under the XCT-CDH assumption. If $\mathcal{R}^*$ can get $k+1$ messages, he can compute Ψ_j, for $j = 1, 2, \cdots, k+1$. Namely, after obtaining $(e(g_0,h)^{t_{\sigma_1}})^z, (e(g_0,h)^{t_{\sigma_2}})^z, \cdots,$

$(e(g_0,h)^{t_{\sigma_k}})^z$, $\mathcal{R}^*$ can compute $(e(g_0,h)^{t_{\sigma_{k+1}}})^z$. This contradicts to the XCT-CDH assumption. Hence, $\mathcal{R}^*$ can obtain at most k messages.

$\{\Gamma_1,\Gamma_2,\cdots,\Gamma_k\}$ and $\{\Upsilon_1,\Upsilon_2,\cdots,\Upsilon_k\}$ are random elements in $\mathbb{G}_\tau$. $\{CT_1,CT_2,\cdots,CT_m\}$ are random elements in $\mathbb{G}_\tau\times\mathbb{G}_2\times\mathbb{G}^2$. $\{\Phi_1,\Phi_2,\cdots,\Phi_k\}$ and $\{\Phi_1^*,\Phi_2^*,\cdots,\Phi_k^*\}$ are identically distributed.

Therefore, the outputs of $\mathcal{R}^*$ and $\hat{\mathcal{R}}^*$ are indistinguishable.

□

Theorem 9.6 *Our oblivious transfer with access control II (AC-OT$^m_{k\times1}$-II) is semantically secure under the q-strong Diffie-Hellman assumption and extended chosen-target computational Deffie-Hellman assumption.*

Proof: There are two types adversaries:

- Type-I: The adversary can compute Γ_ϱ from $(C_{\sigma_{\varrho_1}},C_{\sigma_{\varrho_2}},C_{\sigma_{\varrho_3}})$, then he can act as an authorized requester to interact with the service provider.
- Type-II: The adversary can compute M_{σ_ϱ} from the ciphertext $CT_{\sigma_\varrho}=(C_{\sigma_{\varrho_0}},C_{\sigma_{\varrho_1}},C_{\sigma_{\varrho_2}},C_{\sigma_{\varrho_3}})$.

We will show that a Type-I adversary can be used to break the q-SDH assumption or XCT-CDH assumption and a Type-II adversary can be used to break the XCT-CDH assumption.

Type-I: Suppose that $\mathcal{A}$ is a Type-I adversary.

1. If $\mathcal{A}$ can forge a signature (σ^*,s^*,r) for a requester with secure-public key pair (x_r^*,y_r^*), then obtain $\Gamma_\varrho,\Upsilon_\varrho,\Phi_\varrho$ and Ψ_ϱ, $\mathcal{B}$ can use $\mathcal{A}$ to break the q-SDH assumption[4].
2. If $\mathcal{A}$ cannot compute (σ^*,s^*,r), he can compute Γ_ϱ from $(C_{\sigma_{\varrho_1}},C_{\sigma_{\varrho_2}},C_{\sigma_{\varrho_3}})$. If it is, $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption as follows: given $e(g,C_{\sigma_{\varrho_1}})=(e(g,h)^{x+r})^{t_{\sigma_\varrho}}$, $e(g_1,h)^{t_{\sigma_\varrho}}$, $e(g_2,h)^{t_{\sigma_\varrho}}$ and $e(g_0g_1^sg_2^{x_r},h)$, the aim of $\mathcal{B}$ is to compute $e(g_0g_1^sg_2^{x_r},h)^{t_{\sigma_\varrho}}$. $\mathcal{B}$ sends $(C_{\sigma_{\varrho_1}},C_{\sigma_{\varrho_2}},C_{\sigma_{\varrho_3}})$ to $\mathcal{A}$, if $\mathcal{A}$ can compute Γ_ϱ, $\mathcal{B}$ aborts. $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption as $\Gamma_\varrho=e(g_0g_1^sg_2^{x_r},h)^{t_{\sigma_\varrho}}$.

[4]The signature is existentially unforgeable against the adaptively chosen messages attack under q-SDH assumption [ASM06].

Type-II: Suppose that $\mathcal{A}$ is a Type-II adversary. If $\mathcal{A}$ can compute M_{σ_ϱ} from $CT_{\sigma_\varrho} = (C_{\sigma_{\varrho 0}}, C_{\sigma_{\varrho 1}}, C_{\sigma_{\varrho 2}}, C_{\sigma_{\varrho 3}})$, $\mathcal{B}$ can uses $\mathcal{A}$ to break the XCT-CDH assumption as follows: given $e(g, C_{\sigma_{\varrho 1}}) = (e(g,h)^{x+r})^{t_{\sigma_\varrho}}$, $e(g_1,h)^{t_{\sigma_\varrho}}$, $e(g_2,h)^{t_{\sigma_\varrho}}$ and $Z = e(g_0,h)^z$, the aim of $\mathcal{B}$ is to compute $(e(g_0,h)^z)^{t_{\sigma_\varrho}}$. $\mathcal{B}$ sends $CT_{\sigma_\varrho} = (C_{\sigma_{\varrho 0}}, C_{\sigma_{\varrho 1}}, C_{\sigma_{\varrho 2}}, C_{\sigma_{\varrho 3}})$ to $\mathcal{A}$. If $\mathcal{A}$ can compute M_{σ_ϱ}, $\mathcal{B}$ aborts. $\mathcal{B}$ can compute $e(g_0,h)^{zt_{\sigma_\varrho}} = \frac{C_{\sigma_{\varrho 0}}}{M_{\sigma_\varrho}}$. So, $\mathcal{B}$ can use $\mathcal{A}$ to break the XCT-CDH assumption.

Therefore, our AC-OT$^m_{k\times 1}$-II is semantically secure.

□

Complexity. We compare the computation cost and communication cost of our schemes with those of [CDN09] in Table 9.1 and Table 9.2, respectively. By m and γ, we denote the number of the messages managed by the service provider and the number of the categories included in the access control lists in [CDN09].

9.4 Chapter Summary

One of the fundamental challenges in an open communication channel is to protect users' privacy, including both PII and the selected services. In this chapter, we proposed two efficient AC-OT schemes. In our schemes, a requester can obtain services from the service provider obliviously if he has obtained a credential from the issuer. The service provider knows the number of the services selected by an authorized requester, but nothing about the content of the selected services and the PII of the requester. The requester is required to obtained a credential from the issuer, and is not required to authenticate himself to the service provider. Furthermore, there is no need of zero-knowledge proof. Notably, the credentials in the first scheme are transferable, and *all-or-nothing non-transferable* in the second scheme.

Table 9.1: The Computation Cost of AC-OT$^m_{k\times 1}$-I and AC-OT$^m_{k\times 1}$-II Schemes

Schem	Computation Cost							
	Setup			Issue		Commitment Phase	Transfer Phase	
	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{SP}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{SP}$	$\mathcal{SP}$	$\mathcal{R}$
[CDN09]	T_e	0	$(\gamma+1)T_e+T_p$	$(\gamma+5)T_e$	$5\gamma T_e+2\gamma T_p$	$m(T_e+T_p)$	$(6+14\gamma)T_e$ $+(6+9\gamma)T_p$	$(10+24\gamma)T_e$ $+(6+9\gamma)T_p$
AC-OT$^m_{k\times 1}$-I	T_e	0	T_e	T_e	T_e+T_p	$2mT_e$	kT_e	$2kT_e+kT_p$
AC-OT$^m_{k\times 1}$-II	T_e	T_e	T_e	$2T_e$	$2(T_e+T_p)$	$4mT_e$	kT_e	$4kT_e+kT_p$

Table 9.2: The Communication Cost of AC-OT$^m_{k\times 1}$-I and AC-OT$^m_{k\times 1}$-II Schemes

Scheme	Communication Cost				
	Setup	Issue	Commitment Phase	Transfer Phase	
	$\mathcal{I}\rightarrow\mathcal{SP}$	$\mathcal{I}\rightarrow\mathcal{R}$	$\mathcal{SP}\rightarrow\mathcal{R}$	$\mathcal{SP}\rightarrow\mathcal{R}$	$\mathcal{R}\rightarrow\mathcal{SP}$
[CDN09]	0	$2\gamma E_{\mathbb{Z}_p}+\gamma E_{\mathbb{G}}$	$m(E_{\mathbb{G}}+E_{\mathbb{G}_\tau})$	$3E_{\mathbb{G}}+3E_{\mathbb{G}_\tau}+E_{\mathbb{Z}_p}$	$(2\gamma+1)E_{\mathbb{G}}+(3\gamma+1)E_{\mathbb{G}_\tau}+(11\gamma+4)E_{\mathbb{Z}_p}$
AC-OT$^m_{k\times 1}$-I	$E_{\mathbb{Z}_p}$	$E_{\mathbb{G}_1}+E_{\mathbb{Z}_p}$	$m(E_{\mathbb{G}_2}+E_{\mathbb{G}_\tau})$	$kE_{\mathbb{G}_\tau}$	$kE_{\mathbb{G}_\tau}$
AC-OT$^m_{k\times 1}$-II	$E_{\mathbb{Z}_p}$	$E_{\mathbb{G}_1}+2E_{\mathbb{Z}_p}$	$mE_{\mathbb{G}_2}+3mE_{\mathbb{G}_\tau}$	$kE_{\mathbb{G}_\tau}$	$kE_{\mathbb{G}_\tau}$

Part IV

Conclusion and Future Work

Chapter 10

Conclusion and Future Work

10.1 Conclusion

Preserving privacy has been a primary concern of users in open communication environments. In this thesis, we proposed some secure and provable privacy-preserving access control schemes which were derived from cryptographic primitives. Our contributions to access control schemes lie in not only the theoretical research but also practical applications. The contributions in this work can be summarized in the following three aspects: protection of accessed contents, protection of personal information and protection of both access contents and personal information.

10.1.1 Protection of Accessed Contents

Access control schemes with accessed contents protection allow users access the intended services without the service providers seeing the contents of the selected services. In this thesis, we proposed two identity-based data storage schemes where a requester can obtain services from proxy servers without releasing the contents of the selected services to them. The first scheme aimed to provide a file-based and distributed data storage scheme in the intra-domain, while the second scheme considered the inter-domain scenario. We formalized the definitions and security models of file-based data storage schemes.

10.1.2 Protection of Personal Information

Access control schemes with personal information protection can enable users to obtain the selected services without being identified. This is especially necessary

in the complex communication environments, such as cloud computing, distributed systems. We proposed a privacy-preserving decentralized KP-ABE scheme where a user can obtain secret keys from multiple authorities without being traced by his attributes. Furthermore, multiple authorities can work independently without any cooperation. Notably, authorities can join or leave the system dynamically without re-initializing the systems and re-issuing secret keys to users.

Considering distributed systems are subject to DoS attacks, we constructed an attribute-based data transfer with filtering scheme where both the sender and the receiver can specify an access structure such that only the qualified receivers can access the protected services and only the qualified senders can send messages to him, respectively. Especially, an efficient filtering algorithm was proposed to help receivers filter out false messages prior to executing the expensive decryption algorithm.

Although SSO schemes have been proposed to reduce the burden of managing numerous usernames and passwords, they were not formally proven. We proposed formal definitions and security models for SSO and DSSO, and gave a generic construction of DSSO. Additionally, we proved the security of our generic construction of DSSO in the proposed security model.

10.1.3 Protection of Access Contents and Personal Information

In the schemes with accessed contents protection, a user can be traced by his personal information, such as identity and identifier. Meanwhile, in the schemes with personal information protection, a user can be traced and identified by the actions performed by him. Therefore, a sound privacy-preserving scheme should provide protections of both the accessed contents and the personal information.

We proposed two oblivious access control schemes. In these schemes, a user can access services obliviously if he has been authorized by the authority. A service provider knows the number of the services accessed by an authorized user and does not know anything about the contents of the user selected services and the user's personal information, such as attributes and private credentials. The first scheme was constructed by introducing an ABE with constant communication and computation cost into an OT scheme. Whereas, the second scheme was designed by introducing a cryptographic primitive called OSBE to an OT scheme.

10.2 Future Work

Future work on this thesis may consider the following research topics.

1. *Accountable Privacy-Preserving Access Control.* Although there are many privacy-preserving access control schemes have been proposed, there is no scheme to discuss how to proven legal users abusing resources. However, this is an important issue in practice as legal user may potentially overuse part of the resource.

2. *Privacy-Preserving Decentralized Ciphertext-Policy Attribute-Based Encryption.* In this thesis, we proposed a privacy-preserving decentralized key-policy attribute-based encryption scheme. The encryptor in a ciphertext-policy attribute -based encryption scheme can determine the access policy, namely the encryptor has more control on the encrypted data. Thereafter, it is an interesting work to construct a privacy-preserving decentralized ciphertext-policy attribute-based encryption.

3. *Application.* It is an interesting work to apply provable privacy-preserving access control schemes into practical and privacy-sensitive systems, such as cloud computing, patent search system and DNA-database.

Bibliography

[ABC+07] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. Provable data possession at untrusted stores. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2007*, pages 598–610, Alexandria, Virginia, USA, October 28-31 2007. ACM. 33

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Proceedings: Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270, Santa Barbara, California, USA, August 20-24 2000. Springer. 3

[ADR02] Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In Lars R. Knudsen, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107, Amsterdam, The Netherlands, April 28 - May 2 2002. Springer. 28

[AFGH06] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, 2006. 33, 35, 37

[AGK03] John Argyrakis, Stefanos Gritzalis, and Chris Kioulafas. Privacy enhancing technologies: A review. In Roland Traunmüller, editor, *Proceedings: Electronic Government - EGOV 2003*, volume 2739 of

Lecture Notes in Computer Science, pages 282–287, Prague, Czech Republic, September 1-5 2003. Springer. 163

[AI09] Nuttapong Attrapadung and Hideki Imai. Dual-policy attribute based encryption. In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *Proceedings: Applied Cryptography and Network Security-ACNS 2009*, volume 5536 of *Lecture Notes in Computer Science*, pages 168–185, Paris-Rocquencourt, France, June 2-5 2009. Springer. 82

[AIR01] Bill Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135, Innsbruck, Austria, May 6-10 2001. Springer. 3, 144

[AK04] Ian F. Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351367, 2004. 105, 142

[All01] Liberty Alliance, 2001. http://www.projectliberty.org/. 122

[All06] Liberty Alliance. Liberty id-wsf authentication, single sign-on, and identity mapping services specification, 2006. http://www.projectliberty.org/liberty/content/download/871/6189/file/liberty-idwsf-authn-svc-v2.0.pdf. 123

[ALP11] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Proceedings: Public Key Cryptography - PKC 2011*, volume 6571, of *Lecture Notes in Computer Science*, pages 90–108, Taormina, Italy, March 6-9 2011. Springer. 143, 144, 161

[AO01] Masayuki Abe and Miyako Ohkubo. Provably secure fair blind signatures with tight revocation. In Colin Boyd, editor, *Proceedings: Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 583–602, Gold Coast, Australia, December 9-13 2001. Springer. 3

[ASM06] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-taa. In Roberto De Prisco and Moti Yung, editors, *Proceedings: Security and Cryptography for Networks - SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125, Maiori, Italy, September 6-8 2006. Springer. 163, 172, 175

[Au09] Man Ho Allen Au. *Contribution to Privacy-Preserving Cryptographic Techniques*. PhD thesis, University of Wollongong, Wollongong, NSW, Australia, May 2009. 3

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Proceedings: Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, Interlaken, Switzerland, May 2-6 2004. Springer. 66, 91

[BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Proceedings: Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, Interlaken, Switzerland, May 2-6 2004. Springer. 15, 168, 171

[BBH06] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In David Pointcheval, editor, *Proceedings: The Cryptographers' Track at the RSA Conference - CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 226–243, San Jose, CA, USA, February 13-17 2006. Springer. 69

[BBS98] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144, Espoo, Finland, May 31-June 4, 1998. Springer. 3, 34

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *Proceedings: Advances in*

Cryptology - CRYPTO 2004, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, California, USA, August 15-19 2004. Springer. 3, 172

[BCC88] Gilles Brassard, David Chaum, and Claude Crébpeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, October 1988. 19

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Shai Halevi, editor, *Proceedings: Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 108–125, Santa Barbara, CA, USA, August 16-20 2009. Springer. 3

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Proceedings: Advances in Cryptology - CRYPTO 1996*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15, Santa Barbara, California, USA, August 18-22 1996. Springer. 18

[BdM94] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In Tor Helleseth, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 1993*, volume 765 of *Lecture Notes in Computer Science*, pages 274–285, Lofthus, Norway, May 23-27 1994. Springer. 3

[Bei96] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. Phd thesis, Israel Institute of Technology, Technion, Haifa, Israel, June 1996. 10, 82

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Proceedings: Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, California, USA, August 1923 2001. Springer. 12, 14, 35, 81, 105, 148

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Proceedings: Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275, Santa Barbara, California, USA, August 14-18 2005. Springer. 24, 58

[Bla93] Matt Blaze. A cryptographic file system for unix. In *Proceedings: ACM Conference on Computer and Communications Security - CCS1993*, pages 9–16, Fairfax, Virginia, USA, November 3-5 1993. ACM. 34

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970. 106

[BNPS02] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of rsa inversion oracles and the security of chaums rsa-based blind signature scheme. In Paul F. Syverson, editor, *Proceedings: Financial Cryptography - FC 2001*, volume 2339 of *Lecture Notes in Computer Science*, pages 319–338, Grand Cayman, British West Indies, February 19-22 2002. Springer. 15

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Yvo Desmedt, editor, *Proceedings: Public Key Cryptography - PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46, Miami, FL, USA, January 6-8 2003. Springer. 15

[Bon98] Dan Boneh. The decision diflie-hellman problem. In Joe P. Buhler, editor, *Proceedings: Algorithmic Number Theory - ANT 1998*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63, Portland, Oregon, USA, June 21-25 1998. Springer. 14

[BP97] Niko Bari and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *Proceedings: Advances m Cryptology - EUROCRYPT 1997, LNCS 1233,*

pp. 480-494, 1997, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494, Konstanz, Germany, May 11-15 1997. Springer. 3

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *Proceedings: ACM conference on Computer and communications security - CCS 1993*, pages 62–73, Fairfax, VA, USA, November 3-5 1993. ACM. 18

[Bro10] Daniel R. L. Brown. *Standards for Efficient Cryptography SEC 2: Recommended Elliptic Curve Domain Parameters*. Certicom Research, 2.0 edition, January 2010. http://www.secg.org/download/aid-784/sec2-v2.pdf. 12

[BSCGS06] Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: A taxonomy and open issues. In Ari Juels, Marianne Winslett, and Atsuhiro Goto, editors, *Proceedings: ACM Workshop on Digital Identity Managemen - DIM 2006*, pages 1–10, Alexandria, VA, USA, November 3 2006. ACM. 124, 163

[BSS02] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to Ad-hoc groups. In Moti Yung, editor, *Proceedings: Advances in Cryptology - CRYPTO 2002*, volume 2242 of *Lecture Notes in Computer Science*, pages 465–480, Santa Barbara, California, USA, August 18-22 2002. Springer. 3

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings: IEEE Symposium on Security and Privacy - S & P 2007*, pages 321–334, Oakland, California, USA, May 20-23 2007. 3, 80, 81, 82, 83, 84, 87, 146, 147, 161

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444, St. Petersburg, Russia, May 28-June 1 2006. Springer. 151

[Cam05] Kim Cameron. The laws of identity. Whitepaper, May 2005. http://msdn.microsoft.com/en-us/library/ms996456.aspx. 122

[CC09] Melissa Chase and Sherman S.M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Proceedings: ACM Conference on Computer and Communications Security-CCS'09*, pages 121–130, Chicago, Illinois, USA, November 9-13 2009. ACM. 81, 83, 84, 85, 87, 91, 104

[CDN09] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access control. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromyti, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2009*, pages 131–140, Chicago, Illinois, USA, November 9-13 2009. 3, 4, 144, 145, 176, 177

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In Jeffrey Scott Vitter, editor, *Proceedings: ACM Symposium on the Theory of Computing - STOC 1998*, pages 209–218, Dallas, Texas, USA, May 23-26 1998. ACM. 19

[CGH09] Scott E. Coull, Matthew Green, and Susan Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *Proceedings: Public Key Cryptography - PKC 2009*, volume 5443 of *Lecture Notes in Computer Science*, pages 501–520, Irvine, CA, USA, March 18-20 2009. Springer. 3, 4, 144, 145

[CGS06] Jan Camenisch, Thomas Gross, and Dieter Sommer. Enhancing privacy of federated identity management protocols: anonymous credentials in ws-security. In Ari Juels and Marianne Winslett, editors, *Proceedings: ACM workshop on Privacy in electronic society - WPES 2006*, pages 67–72, Alexandria, VA, USA,, October 30 2006. ACM. 163

[CH91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, Brighton, UK, April 8-11 1991. Springer. 3

[CH07] Ran Canetti and Susan Hohenberger. Chosen-ciphertext secure proxy re-encryption. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverso, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2007*, pages 185–194, Alexandria, Virginia, USA, October 28-31 2007. ACM. 3

[Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Proceedings: Advances in Cryptology - CRYPTO 1982*, pages 199–203, Santa Barbara, California, USA, August 23-25 1983. 3

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communication of ACM*, 28(10):1030–1044, 1985. 3, 91, 163

[Cha07] Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *Proceedings: Theory of Cryptography Conference-TCC'07*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534, Amsterdam, The Netherlands, February 21-24 2007. Springer. 80, 81, 83, 84, 87, 91, 104

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Proceedings: Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, Interlaken, Switzerland, May 2-6 2004. Springer. 52

[CKRS09] Jan Camenisch, Markulf Kohlweiss, Alfredo Rial, and Caroline Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In Stanislaw Jarecki and Gene Tsudik, editors, *Proceedings: Public Key Cryptography - PKC 2009*, volume 5443 of *Lecture Notes in Computer Scienc*, pages 196–214, Irvine, CA, USA, March 18-20 2009. Springer. 88, 89

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118, Innsbruck, Austria, May 6-10 2001. Springer. 3, 132, 136, 167

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Proceedings: Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76, Santa Barbara, California, USA, August 18-22 2002. Springer. 3, 91

[CN07] Ling Cheung and Calvin Newport. Provably secure ciphertext policy abe. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings: ACM Conference on Computer and Communications Security-CCS 2007*, pages 456–465, Alexandria, Virginia, USA, October 28-31 2007. ACM. 81, 82, 83, 161

[CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 573–590, Barcelona, Spain, May 20-24 2007. Springer. 3, 145, 146

[CP02] Winnie Chung and John Paynter. Privacy issues on the internet. In Jr Ralph H. Sprague, editor, *Proceedings: Hawaii International Conference on System Sciences - HICSS-35 2002*, pages 193: 1–9, Big Island, Hawaii, USA, January 7-10 2002. IEEE. 3

[CP07] Jan Camenisch and Birgit Pfitzmann. *Federated Identity Management*, security, privacy, and trust in modern data management Part III Privacy Enhancing, pages 213–238. Data-Centric Systems and Applications. Springer, New York, 2007. 121, 163

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In Burton S. Kaliski Jr., editor, *Proceedings: Advances in Cryptology - CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424, Santa Barbara, California, USA, August 17-21 1997. Springer. 30

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Proceedings: Advances in Cryptology - CRYPTO 19998*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Santa Barbara, California, USA, August 23-27 1998. Springer. 2, 23

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng:. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In Serge Vaudenay, editor, *Proceedings: Public Key Cryptography - PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 172–183, Les Diablerets, Switzerland, January 23-26 2005. Springer. 168, 172

[CT07] Cheng-Kang Chu and Wen-Guey Tzeng. Identity-based proxy re-encryption without random oracles. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *Proceedings: Information Security Conference - ISC 2007*, volume 4779 of *Lecture Notes in Computer Science*, pages 189–202, Valparaso, Chile, October 9-12 2007. Springer. 35, 37, 60, 62

[CW79] J.Lawrence Carter and Mark N. Wegman. Universalclasses of hash-functions. *Journal of Computer and System Sciences*, 18(2):143–154, April 1979. 18

[CZF11] Cheng Chen, Zhenfeng Zhang, and Dengguo Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In Xavier Boyen and Xiaofeng Chen, editors, *Proceedings: International Conference on Provable Security - ProvSec 2011*, volume 6980 of *Lecture Notes in Computer Science*, pages 84–101, Xi'an, China, October 16-18 2011. Springer. 144, 161

[CZLC05] Tierui Chen, Bin B. Zhu, Shipeng Li, and Xueqi Cheng. Threspassporta distributed single sign-on service. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Proceedings: International Conference on Intelligent Computing - ICIC 2005*, volume 3645 of *Lecture Notes in Computer Science*, pages 771–780, Hefei, China, August 23-26 2005. Springer. 123

[DAC87] *A Guide to Understanding Discretionary Access Control in Trusted Systems*. National Computer Security Center, USA, version-1 edition, September 1987. 1

[Dam99] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In Ivan Bjerre Damgård, editor, *Lectures on Data Security, Modern Cryptology in Theory and Practice, Summer School 1998*, volume 1561 of *Lecture Notes in Computer Science*, pages 63–86, Aarhus, Denmark, July 1999. Springer. 19

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976. 2, 13, 21, 26

[DP08] Cécile Delerablée and David Pointcheval. Dynamic threshold public-key encryption. In David Wagner, editor, *Proceedings: Advances in Cryptology-CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 317–334, Santa Barbara,California, USA,, August 17-21 2008. Springer. 143

[DPP07] Cécile Delerablée, Pascal Paillier, , and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Proceedings: Pairing-Based Cryptography-Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 39–59, Tokyo, Japan, July 2-4 2007. Springer. 143

[DSS94] *Digital Singnature Stands (DSS)*. Federal Information Processing Standards Publication 186, U.S. Department of Commerce, National

Institute of Standards and Technology (NIST), Computer Systems Laboratory (CSL), May 1994. 2

[DT07] Ivan Damgård and Rune Thorbek. Non-interactive proofs for integer multiplication. In Moni Naor, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 412–429, Barcelona, Spain, May 20-24 2007. Springer. 152, 157

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, July 1985. 2, 23

[EMN+09] Keita Emura, Atsuko Miyaji, Akito Nomura, Kazumasa Omote, and Masakazu Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In Feng Bao, Hui Li, and Guilin Wang, editors, *Proceedings: Information Security Practice and Experience-ISPEC 2009*, volume 5451 of *Lecture Notes in Computer Science*, pages 13–23, Xi'an, China, April 13-15 2009. Springer. 143, 161

[FAL06] Keith Frikken, Mikhail Atallah, and Jiangtao Li. Attribute-based access control with hidden policies and hidden credentials. *IEEE Transactions on Computers*, 55(10):1259–1270, 2006. 144

[Far75] Nabil H. Farhat. Nonlinear optical data processing and filtering: A feasibility study. *IEEE Transactions on Computers*, C-24(4):443–448, April 1975. 106

[FHS96] Stephanie Forrest, Steven A. Hofmeyr, and Anil Somayaji. Sense of self for unix processes. In *Proceedings: IEEE Symposium on Security and Privacy- S&P 1996*, pages 120–128, Oakland, CA, USA, May 6-8 1996. IEEE. 34

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Proceedings: Advances in Cryptology - CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491, Santa Barbara, California, USA, August 22-26 1994. Springer. 23

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. Rsa-oaep is secure under the rsa assumption. In Joe Kilian, editor, *Proccedings: Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274, Santa Barbara, California, USA, August 19-23 2001. Springer. 18, 23

[GA07] Matthew Green and Giuseppe Ateniese. Identity-based proxy re-encryption. In Jonathan Katz and Moti Yung, editors, *Proceedings: Applied Cryptography and Network Security - ACNS 2007*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306, Zhuhai, China, June 5-8 2007. Springer. 35, 37, 60, 62

[GH07] Matthew Green and Susan Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In Kaoru Kurosawa, editor, *Proceedings: Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 265–282, Kuching, Malaysia, December 2-6 2007. Springer. 88, 89

[GJKR01] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Robust threshold DSS signatures. *Information and Computation*, 164(1):54–84, 2001. 83

[GlJK+99] Rosario Gennaro, Stanis law Jarecki, Hugo Krawczyk, , and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In Jacques Stern, editor, *Proceedings: Advances in Cryptology-EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 295–310, Prague, Czech Republic, May 2-6 1999. Springer. 83

[GMR86] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Robert Sedgewick, editor, *Proceedings: ACM Symposium on Theory of Computing - STOC 1985*, pages 291–304, Providence, Rhode Island, USA, May 6-8 1986. ACM. 29

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988. 26, 27

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proceedings: Symposium on Foundations of Computer Science - FOCS 1986*, pages 174–187, Toronto, Canada, October 27-29 1986. IEEE. 29

[Gol90] Oded Goldr. A note on computationalindistinguishability. *Information Processing Letters*, 34(6):277–281, May 1990. 9

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, September 2008. 12

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings: 13th ACM Conference on Computer and Communications Security - CCS 2006*, pages 89–98, Alexandria, VA, USA, October 30 - November 3 2006. ACM. 3, 81, 82, 87, 98, 120, 148, 161

[HBSO03] Jason E. Holt, Robert W. Bradshaw, Kent E. Seamons, and Hilarie K. Orman:. Hidden credentials. In Paul F. Syverson Sushil Jajodia, Pierangela Samarati, editor, *Proceedings: ACM Workshop on Privacy in the Electronic Society - WPES 2003*, pages 1–8, Washington, DC, USA, October 30 2003. ACM. 163, 164

[HIM02] Hakan Hacigümü, Bala Iyer, and Sharad Mehrotra. Providing database as a service. In Rakesh Agrawal and Klaus R. Dittrich, editors, *Proceedings: International Conference on Data Engineering - ICDE 2002*, San Jose, CA, USA, February 26 - March 1 2002. IEEE. 32

[HLR10] Javier Herranz, Fabien Laguillaumie, and Carla Ráfols. Constant size ciphertexts in threshold attribute-based encryption. In Phong Q. Nguyen and David Pointcheval, editors, *Proceedings: Public Key Cryptography-PKC 2010*, Lecture Notes in Computer Science, pages 19–34, Paris, France, May 26-28 2010. Springer. 81, 143, 161

[HMSY10] Jinguang Han, Yi Mu, Willy Susilo, and Jun Yan. A generic construction of dynamic single sign-on with strong security. In Sushil Jajodia and Jianying Zhou, editors, *Proceedings: International ICST Conference on Security and Privacy in Communication Networks - SecureComm 2010*, volume 50 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 181–198, Singapore, September 7-9 2010. Spinger. 121

[HN11] Junbeom Hur and Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel and Distributed Systems*, 22(7):1214–1221, 2011. 33, 83

[HSM13a] Jinguang Han, Willy Susilo, and Yi Mu. Identity-based data storage in cloud computing. *Future Generation Computer Systems*, 29(3):673–681, March 2013. 61

[HSM13b] Jinguang Han, Willy Susilo, and Yi Mu. Identity-based secure distributed data storage schemes. *IEEE Transactions on Computers*, 2013. Accepted on January 14, 2013. 32

[HSMY12a] Jinguang Han, Willy Susilo, Yi Mu, and Jun Yan. Attribute-based oblivious access control. *The Computer Journal*, 55(10):1202–1215, October 2012. 142

[HSMY12b] Jinguang Han, Willy Susilo, Yi Mu, and Jun Yan. Efficient oblivious transfers with access control. *Computers and Mathematics with Applications*, 63(4):827–837, Feburary 2012. 163

[HSMY12c] Jinguang Han, Willy Susilo, Yi Mu, and Jun Yan. Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems*, 23(11):2150–2162, November 2012. 80

[ID03] Anca Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *Proceedings: Network and Distributed System Security Symposium - NDSS 2003*, pages 1–20, San Diego, California, USA, February 6-7 2003. The Internet Society. 35

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *Proceedings: 47th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2006*, pages 239–248, Berkeley, California, USA, October 21-24 2006. IEEE. 4, 144

[JJ07] Ari Juels and Burton S. Kaliski Jr. PORs: Proofs of retrievability for large files. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2007*, pages 584–597, Alexandria, Virginia, USA, October 28-31 2007. ACM. 33

[JP94] Trent Jaeger and Atul Prakash. Support for the file system security requirements of computational E-mail systems. In *Proceedings: ACM Conference on Computer and Communications Security-CCS 1994*, pages 1–9, Fairfax Va., USA, November 2-4 1994. ACM. 115

[JSS04] William K. Josephson, Emin G*ü*n Sirer, and Fred B. Schneider. Peer-to-peer authentication with a distributed single sign-on service. In Geoffrey M. Voelker and Scott Shenker, editors, *Proceedings: International conference on Peer-to-Peer Systems - IPTPS 2004*, volume 3279 of *Lecture Notes in Computer Science*, pages 250–258, La Jolla, CA, USA, February 26-27 2004. Springer. 123

[KK05] Vishal Kher and Yongdae Kim. Securing distributed storage: Challenges, techniques, and systems. In Vijay Atluri, Pierangela Samarati, William Yurcik, Larry Brumbaugh, and Yuanyuan Zhou, editors, *Proceedings: ACM Workshop On Storage Security And Survivability - StorageSS 2005*, pages 9–25, Fairfax, VA, USA, November 11 2005. ACM. 33

[KL06] Bert-Jaap Koops and Ronald Leenes. Identity theft, identity fraud and/or identity-related crime - definitions matter. *Datenschutz und Datensicherheit*, 30(9):553–556, September 2006. 163

[Koh10] Markulf Kohlweiss. *Cryptographic Protocols For Privacy Enhanced Identity Management*. PhD thesis, Katholieke Universiteit Leuven, March 2010. 3

[KR00] David P. Kormann and Aviel D. Rubin. Risks of the passport single signon protocol. *Computer Networks*, 33(1-6):51–58, June 2000. 122

[KRS+03] Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, and Kevin Fu. Plutus: Scalable secure file sharing on untrusted storage. In *Proceedings: Conference on File and Storage Technologies - FAST 2003*, pages 29–42, San Francisco, CA, USA, March 31-April 2 2003. USENIX. 34

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. 145

[LCH+11] Zhen Liu, Zhenfu Cao, Qiong Huang, Duncan S. Wong, and Tsz Hon Yuen. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In Vijay Atluri and Claudia Diaz, editors, *Proceeedings: European Symposium on Research in Computer Security - ESORICS 2011*, volume 6879 of *Lecture Notes in Computer Scienc*, page 278297, Leuven, Belgium, September 12-14 2011. Springer. 85, 98, 104

[LCLS08] Huang Lin, Zhenfu Cao, Xiaohui Liang, and Jun Shao. Secure threshold multi-authority attribute based encryption without a central authority. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Proceedings: International Conference on Cryptology in India-INDOCRYPT'08*, volume 5365 of *Lecture Notes in Computer Science*, pages 426–436, Kharagpur, India, December 14-17 2008. Springer. 80, 83, 84

[LDB03] Ninghui Li, Wenliang Du, and Dan Boneh. Oblivious signature-based envelope. In *Proceedings: Annual Symposium on Principles of distributed computing - PODC 2003*, pages 182–189, Boston, Massachusetts, USA, July 13-16 2003. ACM. 164

[LHC+11] Jin Li, Qiong Huang, Xiaofeng Chen, Sherman S. M. Chow, Duncan S. Wong, and Dongqing Xie. Multi-authority ciphertext-policy attribute-based encryption with accountability. In *Proceedings: ACM*

Symposium on Information, Computer and Communications Security - ASIACCS 2011, pages 386–390. ACM, 2011. 80, 85

[Lit74] Warren D. Little. An algorithm for high-speed digital filters. *IEEE Transactions on Computers*, C-23(5):466–469, May 1974. 106

[LLZ+10] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xiaohui Liang, and Xuemin (Sherman) Shen. BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2010. DOI: 10.1109/TPDS.2011.95. 105, 107

[LOS+10] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, Riviera, French, May 30-June 3 2010. Springer. 81, 85, 87, 145, 161

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *Proceedings: 6th Annual International Workshop on Selected Areas in Cryptography - SAC 1999*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199, Kingston, Ontario, Canada Springer, August 9-10, 1999 1999. Springer. 3, 91

[LV08] Benoît Libert and Damien Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In Ronald Cramer, editor, *Proceedings: Public Key Cryptography - PKC 2008*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379, Barcelona, Spain, March 9-12 2008. Springer. 3

[LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 568–588, Tallinn, Estonia, May 15-19 2011. Springer. 80, 84, 91, 104

[Lyn06] Ben Lynn. The pairing-based cryptography (PBC) library, 2006. http://crypto.stanford.edu/pbc/. 12

[Mao03] Wenbo Mao. *Moden Cyptography Theory & Practice*. Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey, USA, 2003. 2, 9, 18

[Mat07] Toshihiko Matsuo. Proxy re-encryption systems for identity-based encryption. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Proceedings: Pairing-Based Cryptography - Pairing 2007*, volume 4575 of *Lecture Notes in Computer Science*, pages 247–267, Tokyo, Japan, July 2-4 2007. Springer. 35, 37, 60, 62

[Mau94] Ueli M . Maurer. Towards the equivalence of breaking the diffie-hellman protocol and computing discrete logarithms. In *Proceedings: Advances in Cryptology - CRYPTO 1994*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281, Santa Barbara, California, USA, August 21-25 1994. Springer. 13

[Mit02] Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Transactions on Networking*, 10(5):604–612, October 2002. 106

[MKE08] Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed attribute-based encryption. In Pil Joong Lee and Jung Hee Cheon, editors, *Proceedings: Information Security and Cryptology-ICISC'08*, volume 5461 of *Lecture Notes in Computer Science*, pages 20–36, Seoul, Korea, December 3-5 2008. Springer. 80, 84, 104

[MO97] Masahiro Mambo and Eiji Okamoto. Proxy cryptosystems: Delegation of the power to decyrpt ciphertexts. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E80A(1):54–63, 1997. 34

[MVO96] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996. 2, 9

[MVS01] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring internet denial-of-service activity. In *Proceeings: USENIX Security Symposium -USENIX 2001*, pages 1–6, Wahington, D.C., USA, August 13-17 2001. USENIX. 106

[Nao02] Moni Naor. Deniable ring authentication. In Moti Yung, editor, *Proceedings: Advances in Cryptology - CRYPTO 2002*, volume 2242 of *Lecture Notes in Computer Science*, pages 481–498, Santa Barbara, California, USA, August 18-22 2002. Springer. 3

[NN08] Hoang Lan Nguyen and Uyen Trang Nguyen. A study of different types of attacks on multicast in mobile ad hoc networks. *Ad Hoc Networks*, 6(1):3246, 2008. 105, 142, 143

[NP99a] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings: ACM Symposium on Theory of Computings - TOC 1999*, pages 245–254, Atlanta, Georgia, USA, May 1-4 1999. ACM. 3, 144, 150, 166

[NP99b] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Proceedings: Advances in Cryptology - CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590, Santa Barbara, California, USA, August 15-19 1999. Springer. 3, 144, 150, 166

[NPR99] Moni Naor, Benny Pinkas, and Omer Reingold. Distributed pseudo -random functions and KDCs. In Jacques Stern, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 327–346, Prague, Czech Republic, May 2-6 1999. Springer. 84

[NT94] B. Clifford Neuman and Theodore Ts'o. Kerberos: An authentication sewice for computer networks. *IEEE Communications Magazine*, 32(9):33–38, 1994. 80

[NT06] Samad Nasserian and Gene Tsudik. Revisiting oblivious signature-based envelopes. In Giovanni Di Crescenzo and Aviel D. Rubin,

editors, *Proceedings: Financial Cryptography and Data Security - FC 2006*, volume 4107 of *Lecture Notes in Computer Science*, pages 221–235, Anguilla, British West Indies, February 27-March 2 2006. Springer. 164

[Odl85] Andrew M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In Thomas Beth, Norbert Cot, and Ingemar Ingemarsson, editors, *Proceedings: Advances in Cryptology - CRYPTO 1984*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314, Paris, France, April 9-11 1985. Springer. 13

[OEC80] OECD. *OECD guidelines on the protection of privacy and transborder flows of personal data*. Organisation for Economic Co-operation and Development, Paris, France, September 1980. http://www.oecd.org/. 122

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *Proceedings: Theory of Cryptography - TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99, New York, NY, USA, March 4-7 2006. Springer. 3

[Ope05] OpenID, 2005. http://openid.net. 123

[Opp03] Rolf Oppliger. Microsoft .net passport: A security analysis. *IEEE Computer*, 36(7):29–35, July 2003. 122

[Osb97] Sylvia L. Osborn. Mandatory access control and role-based access control revisited. In *Proceedings: the Second Workshop on Role-Based Access Control*, pages 31–40, Fairfax, VA, USA, November 6-7 1997. ACM. 1

[OSW07] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2007*, pages 159–203, Alexandria, Virginia, USA, October 28-31 2007. 3, 81, 82, 83, 161

[Ped92] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Proceedings: Advances in Cryptology - CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, California, USA, August 11-15 1992. Springer. 20

[PM03a] Andreas Pashalidis and Chris J. Mitchell. Single sign-on using trusted platforms. In Colin Boyd and Wenbo Mao, editors, *Proceedings: International Conference on Information Security - ISC 2003*, volume 2851 of *Lecture Notes in Computer Science*, pages 54–68, Bristol, UK, October 1-3 2003. Springer. 123

[PM03b] Andreas Pashalidis and Chris J. Mitchell. A taxonomy of single sign-on systems. In Reihaneh Safavi-Naini and Jennifer Seberry, editors, *Proceedings: Australasian Conference Information Security and Privacy - ACISP 2003*, volume 2727 of *Lecture Notes in Computer Science*, pages 249–264, Wollongong, Australia, July 9-11 2003. Springer. 121, 123

[PM03c] Andreas Pashalidis and Chris J. Mitchell. Using gsm/umts for single sign-on. In *Proceedings: Symposium on Trends in Communications - SympoTIC 2003*, pages 138–145, Bratislava, Slovakia, October 26-28 2003. IEEE. 123

[PM06] Andreas Pashalidis and Chris J. Mitchell. Using emv cards for single sign-on. In Sokratis K. Katsikas, Stefanos Gritzalis, and Javier Lopez, editors, *Proceedings: Public Key Infrastructure - EuroPKI 2004*, volume 3093 of *Lecture Notes in Computer Science*, pages 205–217, Samos Island, Greece, June 25-26 2006. Springer. 123

[PSG+03] Adam G. Pennington, John D. Strunk, John Linwood Griffin, Craig A.N. Soules, Garth R. Goodson, and Gregory R. Ganger. Storage-based intrusion detection: Watching storage activity for suspicious behavior. In *Proceedings: USENIX Security Symposium - USENIX 2003*, pages 137–152, Washington, D.C., USA, August 4-8 2003. USENIX. 34

[PTMW06] Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure attribute-based systems. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings: ACM Conference on Computer and Communications Security - CCS 2006*, pages 99–112, Alexandria, VA, USA, October 30 - November 3 2006. ACM. 3, 120

[Rab81] Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University, USA, 1981. 3, 144, 145

[RAD77] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, editor, *Proceedings: Foundations of Secure Computation*, pages 169–180, Atlanta, Georgia, USA, October 3-5 1977. Academic Press. 145

[Reh08] Raffq Ur Rehman. *Get Ready for OpenID*. Conformix Technologies Inc., http://www.conformix.com, 2008. 123

[RLZ06] Kui Ren, Wenjing Lou, and Yanchao Zhang. Providing location-aware end-to-end data security in wireless sensor networks. In *Proceedings: IEEE INFOCOM 2006*, pages 1–12, Barcelona, Spain, April. 23-29 2006. IEEE. 107

[RP10] Alfredo Rial and Bart Preneel. Blind attribute-based encryption and oblivious transfer with fine-grained access control. In *Benelux Workshop on Information and System Security - WISSec'10*, pages 1–20, 2010. 83, 146

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Proceedings: Advances in Cryptology - CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, Santa Barbara, California, USA, August 11-15 1992. Springer. 21

[RSA78] Ronald Linn Rivest, Adi Shamir, and Leonard Max Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):121–126, Feburay 1978. 2, 23

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Proceedings: Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9-13 2001. Springer. 3

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996. 1

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Proceedings: Advances in Cryptology - CRYPTO 1989*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, California, USA, August 20-24 1990. Springer. 2

[SFJ09] Suriadi Suriadi, Ernest Foo, and Audun Jϕsang. Auser-centric federated single sign-on system. *Journal of Network and Computer Applications*, 32(2):388–401, March 2009. 123

[SGK+85] Russel Sandberg, David Goldberg, Steve Kleiman, Dan Walsh, and Bob Lyon. Design and implementation of the sun network file system. In *Proceedings: USENIX Technical Conference - USENIX 1985*, pages 119–130, Portland, 1985. USENIX. 33

[Sha84] Adi Shamir. Identity-based cryptosystems and signature scheme. In G. R. Blakley and David Chaum, editors, *Proceedings: Advances in Cryptology - CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, California, USA, August 19-22 1984. Springer. 34, 81, 105

[Sma03] Nigel P. Smart. Access control using pairing based cryptography. In *The Cryptographers' Track at the RSA Conference - CT-RSA 2003*, volume 2612 of *LNCS*, pages 111–121, 2003. 80

[SV10] Pierangela Samarati and Sabrina De Capitani di Vimercati. Data protection in outsourcing scenarios: Issues and directions. In Dengguo Feng, David A. Basin, and Peng Liu, editors, *Proceedings: ACM*

Symposium on Information, Computer and Communications Security - ASIACCS 2010, pages 1–14, Beijing, China, Aprile 13-16 2010. ACM. 33

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22-26 2005. Springer. 3, 80, 81, 83, 105, 110, 120, 161

[TFS04] Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-times anonymous authentication (extended abstract). In Pil Joong Lee, editor, *Proceedings: Advance in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 308–322, Jeju Island, Korea, December 5-9 2004. Springer. 163

[THJ08] Qiang Tang, Pieter Hartel, and Willem Jonker. Inter-domain identity-based proxy re-encryption. In Moti Yung, Peng Liu, and Dongdai Lin, editors, *Proceedings: Information Security and Cryptology - Inscrypt 2008*, volume 5487 of *Lecture Notes in Computer Science*, pages 332–347, Beijing, China, December 14-17 2008. Springer. 35, 37, 60, 62, 67, 68, 75, 77

[VCF+00] John R. Vollbrecht, Pat R. Calhoun, Stephen Farrell, Leon Gommans, George M. Gross, Betty de Bruijn, and Cees T.A.M. de Laat. AAA authorization framework. Informational RFC: 2904, August 2000. 1

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Proceedings: Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22-26 2005. Springer. 25, 26, 43, 47, 50, 91, 151

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Shai Halevi, editor, *Proceedings: Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636, Santa Barbara, CA, USA, August 16-20 2009. Springer. 87

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Proceedings: Public Key Cryptography - PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70, aormina, Italy, March 6-9 2011. Springer. 81, 82, 85, 105, 145, 151, 161

[WWMO10a] Lihua Wang, Licheng Wang, Masahiro Mambo, and Eiji Okamoto. Identity-based proxy cryptosystems with revocability and hierarchical confidentialities. In Miguel Soriano, Sihan Qing, and Javier López, editors, *Proceedings: International Conference on Information and Communications Security - ICICS 2010*, volume 6476 of *Lecture Notes in Computer Science*, pages 383–440, Barcelona, Spain, December 15-17 2010. Springer. 35, 37, 60, 62

[WWMO10b] Lihua Wang, Licheng Wang, Masahiro Mambo, and Eiji Okamoto. New identity-based proxy re-encryption schemes to prevent collusion attacks. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Proceedings: Pairing-Based Cryptography - Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, pages 327–346, Yamanaka Hot Spring, Japan, December 13-15 2010. Springer. 35, 37, 42, 60, 62

[YG10] Zhen Yu and Yong Guan. A dynamic en-route filtering scheme for data reporting in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 18(1):150–163, February 2010. 107

[YGJK10] Man Lung Yiu, Gabriel Ghinita, Christian S. Jensen, and Panos Kalnis. Enabling search services on outsourced private spatial data. *The VLDB Journal*, 19(3):363–384, 2010. 33

[YK11] Ji Won Yoon and Hyoungshick Kim. A perfect collision-free pseudonym system. *IEEE Communications Letters*, 15(6):686–688, June 2011. 3

[YLLZ04] Fan Ye, Haiyun Luo, Songwu Lu, and Lixia Zhang. Statistical en-route filtering of injected false data in sensor networks. In *Proceedings: IEEE INFOCOM 2004*, volume 4, pages 2446–2457, HongKong, March 7-11 2004. IEEE. 107

[YRL11] Shucheng Yu, Kui Ren, and Wenjing Lou. FDAC: Toward fine-grained data access control in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(4):673–686, 2011. 83

[Yue77] C. K. Yuen. On little's digital filtering algorithm. *IEEE Transactions on Computers*, C-26(3):309–309, March 1977. 106

[YWRL10] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *Proceedings: IEEE INFOCOM 2010*, pages 534–542, San Diego, CA, USA, March 15-19 2010. IEEE. 83, 105

[YYY+05] Hao Yang, Fan Ye, Yuan Yuan, Songwu Lu, and William Arbaugh. Toward resilient security in wireless sensor networks. In P. R. Kumar, Andrew T. Campbell, and Roger Wattenhofer, editors, *Proceedings: ACM Symposium on Mobile Ad-hoc Networking and Computing - MOBIHOC 2005*, pages 34–45, Urbana-Champaign, Illinois, USA., May 25-27 2005. ACM. 107

[ZAW+10] Ye Zhang, Man Ho Au, Duncan S. Wong, Qiong Huang, Nikos Mamoulis, David W. Cheung, and Siu-Ming Yiu. Oblivious transfer with access control : Realizing disjunction without duplication. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Proceedings: Pairing-Based Cryptography - Pairing 2010*, volume 6487 of *6487*, pages 96–115, Yamanaka Hot Spring, Japan, December 13-15 2010. Springer. 4, 145

[ZH10] Zhibin Zhou and Dijiang Huang. On efficient ciphertext-policy attribute based encryption and broadcast encryption. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security - CCS 2010*, pages 753–755, Chicago, Illinois, USA, October 4-8 2010. ACM. 143, 161

[ZLLF06] Yanchao Zhang, Wei Liu, Wenjing Lou, and Yuguang Fang. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selcted Areas in Communictions*, 24(2):247–260, February 2006. 107

[ZSJN04] Sencun Zhu, Sanjeev Setia, Sushil Jajodia, and Peng Ning. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *Proceedings: IEEE Symposium on Security and Privacy - S & P 2004*, pages 259–271, Berkeley, CA, USA, May 9-12 2004. IEEE. 105, 107

Index