Anonymity Protection and Access Control in Mobile Network Environment

by

Bing Li

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2016 by the
Graduate Supervisory Committee:

Dijiang Huang, Chair
Guoliang Xue
Gail-Joon Ahn
Yanchao Zhang

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

Wireless communication technologies have been playing an important role in modern society. Due to its inherent mobility property, wireless networks are more vulnerable to passive attacks than traditional wired networks. Anonymity, as an important issue in mobile network environment, serves as the first topic that leads to all the research work presented in this manuscript. Specifically, anonymity issue in Mobile Ad hoc Networks (MANETs) is discussed with details as the first section of research.

To thoroughly study on this topic, the presented work approaches it from an attacker's perspective. Under a perfect scenario, all the traffic in a targeted MANET exhibits the communication relations to a passive attacker. However, localization errors pose a significant influence on the accuracy of the derived communication patterns. To handle such issue, a new scheme is proposed to generate super nodes, which represent the activities of user groups in the target MANET. This scheme also helps reduce the scale of monitoring work by grouping users based on their behaviors.

The first part of work on anonymity in MANET leads to the thought on its major cause. The link-based communication pattern is a key contributor to the success of the traffic analysis attack. A natural way to circumvent such issue is to use link-less approaches. Information Centric Networking (ICN) is a typical instance of such kind. Its communication pattern is able to overcome the anonymity issue with MANET. However, it also comes with its own shortcomings. One of them is access control enforcement. To tackle this issue, a new naming scheme for contents transmitted in ICN networks is presented. This scheme is based on a new Attribute-Based Encryption (ABE) algorithm. It enforces access control in ICN with minimum requirements on additional network components.

Following the research work on ABE, an important function, delegation, exhibits a potential security issue. In traditional ABE schemes, Ciphertext-Policy ABE (CP-

ABE), a user is able to generate a subset of authentic attribute key components for other users using delegation function. This capability is not monitored or controlled by the trusted third party (TTP) in the cryptosystem. A direct threat caused from this issue is that any user may intentionally or unintentionally lower the standards for attribute assignments. Unauthorized users/attackers may be able to obtain their desired attributes through a delegation party instead of directly from the TTP. As the third part of work presented in this manuscript, a three-level delegation restriction architecture is proposed. Furthermore, a delegation restriction scheme following this architecture is also presented. This scheme allows the TTP to have full control on the delegation function of all its direct users.

To my grandma, who, I truly feel, is watching me from heaven.

ACKNOWLEDGMENTS

First of all, I want to thank my advisor Dr. Dijiang Huang, who through his immense patience has guided me to bear my curiosity throughout the past years in exploring premature ideas. His passion and responsibility on research always inspire and encourage me to discern and analyze the essentials of challenging research problems. What's more important is the way he treats other people. Dr. Huang is more of a caring old brother than a strict teacher to me. I truly appreciate his valuable suggestions, not only on research and future career path choice, but also on life-long decisions. I cannot thank him enough for making my past years an exciting and fruitful expedition!

I would also like to acknowledge my committee members, Dr. Guoliang Xue, Dr. Gail-Joon Ahn, and Dr. Yanchao Zhang, for their helpful suggestions and insightful comments during the past years. I have the honor to have attended several talks that are presented by Dr. Xue, Dr. Ahn, and Dr. Zhang, which provide a lot of useful guidance to my own research work. Dr. Xue is a mentor who not only discusses leading-edge research topics with me, but also shows his care to challenges in people's lives. Dr. Ahn's works on Attribute Based Encryption and Access Control provide me insightful views on such topics. I really appreciate his support in preparing my dissertation. Dr. Zhang's work on anonymous routing in mobile networks is one of the threads that lead me to my own research work on mobile anonymity. In addition to my direct interactions with the committee members, I also cherish so many opportunities that I had in exchanging thoughts and ideas with people from their groups.

I was lucky to have worked as an intern with a number of talented people at Facebook and Huawei Technologies. My two mentors, Dr. Wojciech Galuba and Dr. Jiafeng Zhu, provided me with a lot of help and guidance in their different career directions. I really appreciate these great opportunities to explore interesting and

challenging new areas, which greatly expanded my vision in industry work. I also felt blessed to have spent some great time with all the friends at Facebook and Huawei Technologies.

During the past four and half years, I have received a lot of support and encouragement from my friends and colleagues. I want to deeply thank people at SNAC group for getting my back when I need them: Yang Qin, Tianyi Xing, Chun-Jen Chung, Le Xu, Huijun Wu, Weijia Wang, Bo Li, Abdullah Alshalan, Sandeep Pisharody, Pankaj Khatkar, Mahmoud Saada, Ashwin Prabhu Verleker, Yuli Deng, Ankur Chowdhary, and Duo Lv. I also would like to express my special thanks to my other friends at ASU: Ziming Zhao, Dejun Yang, Xi Fang, Xinhui Hu, Jie Shi, Lingjun Li, Yiming Jing, Jingchao Sun, Darin Tupper, Xinsheng Li, Lin Chen, Xiang Zhang, Xilun Chen, and Mengxue Liu.

I also would like to specially thank Zhuoyang Zhou for all the support from her during the past years.

Finally and most importantly, many thanks to my mother, Fangzhen Jiao, whose unconditional love and support helps me through ups and downs in my life. I also want to thank my father, Mindong Li, whose love in his own way makes me strong inside out.

TABLE OF CONTENTS

Page

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER

1 INTRODUCTION 1

1.1 Background 1

1.2 MANET 2

1.3 Anonymity Issue in Networks 3

1.4 ICN 4

1.5 Objectives 5

2 ANONYMITY IN MANETS 8

2.1 Introduction 8

2.2 Related Work 9

2.2.1 Network Model 10

2.2.2 Traffic Monitoring System 11

2.2.3 Previous Work 12

2.3 Proposed Solution 13

2.3.1 Clustering Algorithm 15

2.3.2 Super-node Generation 21

2.4 Evaluation 23

2.5 Conclusion 25

3 ATTRIBUTE-BASED ACCESS CONTROL IN MOBILE ICN 27

3.1 Introduction 27

3.2 Related Work 30

3.2.1 ICN Solutions 30

CHAPTER Page

3.2.2 ABE Schemes 33
3.3 System and Models 35
3.3.1 Application Scenario 35
3.3.2 Attribute-based Naming and Access Control 37
3.3.3 Comparable Attributes and Attribute Rankings 38
3.3.4 Attack Model 40
3.3.5 Preliminaries of ABE 40
3.4 ABE-based ICN Naming Scheme 42
3.4.1 Creating a Content 42
3.4.2 ABE-based Naming Scheme 44
3.4.3 Attribute Rankings 47
3.4.4 Apply ABE-based Naming Scheme in ICN 52
3.5 Performance Analysis and Evaluation 53
3.5.1 Evaluation of the Naming Scheme 53
3.5.2 Real-world Implementation 54
3.5.3 Complexity Comparison 55
3.6 Security Analysis 58
3.6.1 ABE Security Model 60
3.6.2 Security Proof Sketch 61
3.7 Conclusion 67
4 ATTRIBUTE DELEGATION FOR ID-REVOCABLE ATTRIBUTE BASED ENCRYPTION 68
4.1 Introduction 68
4.1.1 ID-revocable Attribute Based Encryption 68

CHAPTER Page

4.1.2 Attribute Delegation Issue 70
4.2 Related Work 71
4.2.1 Delegation Scheme in Original ABE Schemes 72
4.2.2 ID-revocable ABE Scheme 73
4.3 Discussion on Attribute Delegation 74
4.4 Proposed Delegation Scheme for ID-revocable ABE 79
4.4.1 Preliminary Attribute Delegation Scheme 79
4.4.2 Modified Revocation-supporting Schemes 82
4.4.3 Single-ID Two-Level Revocation for CP-ABE Scheme (SID2LR-CP-ABE) 84
4.4.4 Multiple-ID Two-Level Revocation for CP-ABE Scheme (MID2LR-CP-ABE) 88
4.5 Analysis on the Proposed Scheme 93
4.5.1 Application Scenarios 93
4.5.2 Delegation Agent Federation 97
4.5.3 Performance Complexity Analysis 99
4.5.4 Real-world Implementation 104
4.6 Conclusion 107
5 FUTURE RESEARCH DIRECTIONS 109
5.1 Anonymous Communications in Mobile Environment 109
5.2 Further Development for ABE Schemes 110
5.3 Restrict Delegation Capability in ABE Schemes 111
6 SUMMARY 115
REFERENCES 117

LIST OF TABLES

Table Page

2.1 Notations 16

3.1 Time-consumption of Different Operations (in milliseconds) 55

3.2 Comparison of Computation Cost in Decryption 57

3.3 Comparison of Ciphertext Size 59

3.4 Query Information Accessible to the Adversary 67

4.1 Notations. 100

4.2 Computation Complexity Comparison in Pairing Operations 101

4.3 Computation Complexity Comparison in Exponentiation Operations . . . 101

4.4 Storage Cost Comparison 103

LIST OF FIGURES

Figure		Page
2.1	Localization Error Influence on Traffic Analysis.	14
2.2	Different Transmission Distances Change Connectivities.	17
2.3	Asymmetric Communication Relations Influence Communication Distances.	18
2.4	Sparse Node Connections Cannot be Used for Merging Clusters.	20
2.5	A Dendrogram to Record the Cluster Formations.	21
2.6	The Correct Cluster Ratios are Different using Different Metrics.	24
3.1	Basic ICN System Model.	36
3.2	Creating a Content Name.	39
3.3	Creating a Content.	43
3.4	Computation Performance	56
4.1	Breadth-First Restriction.	75
4.2	Depth-First Restriction.	76
4.3	Preliminary Delegation Solution Application Scenario	94
4.4	Proposed Delegation Solution Application Scenario	96
4.5	Relations between the Amount of Attributes and Time Consumption	105
4.6	Relations between the Amount of Revoked IDs and Time Consumption.	107
5.1	Fine-Grained Restriction.	113

Chapter 1

INTRODUCTION

1.1 Background

With rapid development in communication technologies, various forms of networks emerged with their specific technological advantages for different application scenarios. They can provide not only high transmission rate, wide bandwidth, or low latency, but also high reliabilities and better coverage. Mobile networking, as an important part of daily life, is becoming inseparable to modern society. With more businesses and activities of daily life moved to mobile devices, security, privacy and robustness issues are getting more attentions from researchers, potential criminals, and ordinary users. Common mobile networks are based on infrastructures, like cellular networks and wifi. These infrastructures can sometimes be vulnerable when facing severe situations. For example, at the recent earthquake disaster in Nepal, communications and Internet access were greatly influenced, mostly due to an outage of power and damages to last-mile connectivity infrastructures at certain areas nep (2015a,b). Innovative people-searching services for this earthquake provided by Google, Facebook and other organizations would not be very helpful due to connectivity issue. Rescuers need to rely on human messengers to pass information between areas before network connections are recovered. Under such circumstances, infrastructure-less networks are good options to re-establish communications at the early stage of disaster rescue. A common form of such networks is Mobile Ad hoc Networks (MANETs).

1.2 MANET

A MANET is a wireless network consisting of, in many cases, only mobile devices, which form the network in an ad hoc approach. There is no static network infrastructure available in such network environments. Mobile devices play the roles of both a terminal and a forwarding party. When a client decides to send out a message, it follows its routing protocol to pass the message to one or more of its immediate neighbors. The neighbors will then work as a relay to forward such a message till the message reaches its destination or the message is timed out. Several routing protocols have been proposed Zhang *et al.* (2006); El Defrawy and Tsudik (2011) for such purpose. Under such working principles, it is not guaranteed that a reliable end-to-end connection is always available. However, its little requirement on static infrastructures makes it lightweight and easy to deploy, which is quite suitable for certain application scenarios. As mentioned before, in disaster rescue scenarios, MANET is one of few options for establishing network services within the first several hours or days. Such time frame is critical for saving lives. Another application scenario is military usage. MANET is a good candidate for communications between small groups of units in battlefield, where it is normally impossible to establish a reliable communication infrastructure.

Typical attributes of a MANET include hand-held mobile devices with limited battery life and restrained computation/storage resources. It is relatively easy to monitor communication traffic by either eavesdropping on the wireless channels or by capturing authentic mobile devices. Compromising an authentic device is sometimes assumed to be relatively easy. Such potential risk may provide an attacker with all the capabilities as an authentic user in terms of participating in normal communications. These compromised users are usually difficult to be discovered. The open access to

wireless channels also poses risk to the network. Specifically, anonymity is a key issue in such networks.

1.3 Anonymity Issue in Networks

Anonymity refers to the property that a certain party or element cannot be easily identified from a set of similar parties or elements. In network communications, it includes multiple categories, such as sender anonymity, receiver anonymity, relationship anonymity, etc. As their names imply, sender/receiver anonymity refers to property that the sender/receiver of a certain message cannot be easily identified. Relationship anonymity refers to the case that a communication relationship between two or more parties is not easily identifiable.

Relationship Anonymity is an important issue in secure network communications. In end-to-end communications, it means how difficult it is for an eavesdropper to infer the identities of the two communication parties in a session. In traditional host-based networks, the two parties can be identified by their IP addresses in IP headers. VPN may be able to partially protect the addresses by hiding the original header and adding an additional layer. However, VPN is not designed for such application purpose. It cannot provide a reliable protection for anonymous communications. Relationship anonymity is relatively more critical in wireless networks than in wired networks. This is because it is not easy to eavesdrop on the exact route for a particular communication pair as in wireless channels. Anyone who has an adequately-equipped antenna can easily receive all the wireless traffic within a certain transmission range and frequency range. It is feasible to create a network of mobile devices for such monitoring purpose, which can cover a wide range of geographic area. Such monitoring network is normally referred to as a traffic monitoring system, which is commonly used in wireless traffic analysis.

For MANETs, deploying a monitoring system is easy, effective, and worthwhile for the cost. For example, in battlefield, deploying a set of passive signal monitors helps the troops to easily locate enemy forces or to identify critical enemy communications. With a wide coverage of the entire area, the monitoring system can collect a majority of the overall wireless traffic to break anonymity in enemy's network. It is relatively easy to monitor end-to-end communication patterns with a powerful monitoring system.

1.4 ICN

Relationship anonymity is an inherent shortcoming for host-based network architectures. A direct host-to-host session needs to be established before data is transmitted between the two communication parties through the network. Such session is maintained during the entire process until one of the parties closes it.

Information Centric Network (ICN), as a new network paradigm, changed the requirements for relatively static hosts in network communications. It is designed to provide better support to large scale multicast services. Traditional host-based architecture works well in end-to-end communications between a pair of users. However, in multicast scenarios, where one source is sending the same data to multiple recipients, performance can be further improved. According to Cisco (2015), most of the current network traffic is video sharing, from 64% in 2014 up to 80% by 2019. To further improve the overall network performance, several different ICN network architectures are proposed.

In a typical ICN network, data is represented as contents. Different contents have different identifiers/names. They are transmitted through the network in publisher-subscriber model. When a network party publishes a content, it registers the content name to a naming service. Through this naming service, a subscriber who is interested

in such content will be notified with the name of the content. Here, names are unique identifiers to contents in the network. Once a subscriber gets the name of the content, it is able to retrieve a copy of the content using the name through a name-based routing scheme. Copies of the same content may be scattered all over the network, existing in network caches for a certain period of time, depending on what caching algorithm they are running. Typically, if a copy of a certain content is transmitted from one entity to another, all the network caches along the path will keep a copy of it. In this way, next time when another entity requests for the same content, the network is able to locate a near-by network cache, which maintains a copy of the content, but has a shorter distance from the user.

Name based routing scheme uses content names as identifiers to route a copy of content to its consumers. In traditional session based networking environment, each network party is tagged with a unique identifier, for example, IP address. Network traffic is routed based on the entity's identifier. In ICN network, it is the content that is tagged with a unique identifier. Routers along the network path use the content identifiers to forward traffic. In other words, in traditional network, a routing decision is made based on the destination address of the traffic, while in ICN network, it is based on the content identifier to make such decisions.

1.5 Objectives

The inherent relationship anonymity issue with traditional MANET makes it interesting to study how network movements and modeling affect the study of anonymity. In one of the previous works Qin *et al.* (2014), the monitoring system for traffic analysis is assumed to be able to perfectly identify signals from each individual device without any interference. However, in reality, such assumption does not hold when two devices are closely located together. This is one of several example cases where

the previous perfect assumptions do not apply in real world scenario. To deal with such issue, the proposed idea is to relax assumptions on the monitoring system to explore the influence on traffic analysis accuracy, which constitutes the first part of the proposed work, Chapter 2 of this dissertation.

For the second part (Chapter 3), ICN as a new paradigm has a side-effect to overcome the anonymity issues with traditional host-based network patterns. However, it also introduces some challenging issues that need to be addressed urgently. One of such issue is the difficulty in establishing access control on the published contents. In traditional network patterns, a client (consumer) needs to authenticate itself to the server (publisher) with desired privilege to be granted with access. This process takes place before the server transmits the content to the client. However, in ICN environment, contents are published before a consumer initiates an access request to them. Once a content is published, its publisher/owner can hardly keep control of it. To overcome such issue, an in-depth investigation is carried out on the possibilities of applying Attribute Based Encryption (ABE) algorithms into content access control in ICN scenarios.

Based on the research work in Chapter 3 on ABE schemes, one important and interesting topic is on the delegation capability. Normally, a user in an ABE cryptosystem acquires all of his attribute keys and the global parameters from a Trusted Third Party (TTP). The TTP is in charge of generating, assigning and managing attribute keys, which incurs a heavy workload on computation, storage, and communication. Delegation function allows a user to generate keys, for any attribute he has, for other users. In this way, a delegation-capable user, which is referred to as a delegator or delegation agent in the rest of this dissertation, is able to off-load some of the key generation burdens from the TTP. This mechanism has its merits as well as harms. Existing ABE schemes did not specifically focus on how to control such del-

egation capability from being maliciously used. However, a uncontrolled delegation capability is able to create much more risks than the benefits it can gain. Observed by such fact, this topic is studied in Chapter 4.

Based on the work presented in Chapter 2, 3, and 4, future research directions are further discussed in Chapter 5. The goal of this chapter is to provide helpful guidance to continuous work on the three topics presented above. The entire work is summarized in Chapter 6.

Chapter 2

ANONYMITY IN MANETS

2.1 Introduction

As discussed before, a powerful traffic monitoring system is able to capture most of the wireless traffic of a target MANET within a small area. Certain amount of information can be acquired from the captured traffic generated by individual mobile nodes.

In Huang (2008), a new approach is proposed to measure the unlinkability for MANETs based on evidence collected from traffic statistics. This is carried out by slicing time domain into multiple intervals, so that each node can be either a sender or a receiver during one such interval. Then a point-to-point traffic matrix is constructed for each interval, which records the amount of traffic sent from one node to another. After this step, a traffic-communication relation matrix is created so that accumulative traffic amount from one node to another can be calculated for discovering traffic patterns.

This method is effective only when individual nodes can be differentiated without error. However, localization errors can hardly be avoided or eliminated in practice. Therefore, how to model network communications when localization errors exist is crucial for applying the previous work Qin *et al.* (2014) in practice. This is one of the major motivations for the work presented here.

Another issue with the previous method is scalability. In large scale networks, it requires a large amount of resources to generate and store traffic information in matrices. Based on observations from the application scenarios mentioned above,

nodes are likely to move in group patterns. Therefore, instead of modeling the communication patterns for individual nodes, the research focus is placed on group-based communications. In this way, the scalability issue can be reduced and controlled in the proposed solution.

Following this idea, the proposed work aims to model anonymous communications using super-nodes. In brief, super-nodes are used to model the formation of node groups. The proposed approach consists of two steps: (1) In each time interval, all the nodes are divided into multiple clusters using a novel distance metric proposed in this work. (2) Super-nodes are generated based on cluster formations in all the intervals.

The main expected contributions of this work include: (i) Using super-nodes to solve the issues caused by localization errors. A new approach is presented to generate super-nodes based on traffic information collected from the target network. (ii) Proposing a new metric for measuring the distance between nodes and clusters so that both geographical and communicational factors are used to improve the accuracy of clustering algorithms.

2.2 Related Work

In this section, detailed information will be provided on the network models, the traffic monitoring system, and the previous work. The general scenario is that a MANET consisting of a number of mobile nodes is deployed in an area. Data transmitted in it is encrypted so that packet contents are well protected. A traffic monitoring system exists in the area. It consists of a network of passive receivers that capture signals transmitted in the MANET channel. These receivers are connected through communication channels that are different from the one used by the target MANET. Thus, there are no signal interferences between the monitoring system and

the target MANET. More details on the network model and the monitoring system are illustrated below.

2.2.1 Network Model

The communication model of the MANET is based on 802.11b protocol. It can be extended to other protocols as needed. According to Huang (2008), the transmission range d of a node has a relation with the data transmission rate r as $d \propto \frac{1}{r}$. r is included as part of the Physical Layer Convergence Procedure (PLCP) header, which is part of 802.11b physical layer, for decoding purpose. Under such model, potential receivers of a signal can be identified based on the location of the source and its transmission rate.

Some features in MAC layer can leak useful information to adversarial eavesdroppers. To mitigate such issue, some settings are changed: (1) Both source and destination addresses in a frame are set to broadcasting value, i.e. all 1's. Addressing can be handled by upper layers using mechanisms like pairwise shared keys. (2) Virtual carrier sensing, a.k.a. Network Allocation Vector (NAV), mechanism Sharma (2003) is disabled. Virtual carrier sensing is used to determine whether the channel is busy or idle . When a node needs to send out a packet, a Require To Send (RTS) message is sent out to the authentic receiver. The targeted receiver replies the RTS message with a Clear To Send (CTS) message if the channel is idle. Both the RTS and the CTS messages contain information that is used to determine the length of waiting periods for other nodes to send a message in the same channel. In this way, only the targeted receiver does not have to wait for a certain length of time to send out messages. This mechanism is designed to solve collision problem in media access. However, it exposes point-to-point communication relations to passive eavesdroppers. Therefore, such option should be disabled in the communication model.

As mentioned before, nodal movements exhibit group patterns in MANETs. Therefore, the mobility model of the target MANET is assumed to be group based. But neither the group size nor the group distribution is publicly known. Single nodes can be modeled as groups with size of 1.

From above observations, the communication model in MANETs is summarized as follows:

- Source and destination addresses in MAC frame are set to broadcast value;
- Frame contents are securely protected;
- Upper layer protocols (including network layer routing protocol) are unknown to attackers;
- Virtual carrier sensing option is disabled;
- Device specification and transmission range of the nodes are known to attackers;
- Nodes' transmission ranges are not necessarily equal;
- Localization errors are far smaller than transmission ranges;
- Nodes' movement is group-based.

2.2.2 Traffic Monitoring System

The traffic monitoring system can passively collect traffic information. It is able to locate the source of a signal with a certain localization error e. Locating a wireless node based on the Received Signal Strength (RSS) has been proposed and studied for a long time Faria and Cheriton (2006). It requires collaboration from multiple receivers. To improve location detection performance, using temporal link signature for location distinction is proposed in Patwari and Kasera (2007). It constructs temporal link

signatures using multi-path phenomenon and differentiates node locations based on normalized link signatures. Using this technique, a node can be located when the traffic monitoring system receives a signal from it. The traffic monitoring system is assumed to be able to locate each node in the network as long as it receives a signal from the node.

Another feature of the traffic monitoring system is to distinguish different types of hardware. This can be achieved based on research results in radio fingerprinting Kohno *et al.* (2005) Rasmussen and Capkun (2007). The idea is that signals from any two hardware devices are slightly different in terms of timing, frequency, and so forth, due to the imperfection of manufacturing. However, linking radio fingerprints with the source's identity is difficult as the monitoring system is assumed to have very limited internal knowledge about the target MANET. Thus, the assumption for the monitoring system is that it can differentiate senders of different signals but cannot get senders' identities from such signals. In this way, each node can be represented by its radio fingerprint but cannot be correlated to its real identity.

The ability of traffic monitoring system can be summarized as:

- Monitoring system can capture all the communication traffic of the MANET;
- Monitoring system does not emit signal in the communication channel of the MANET;
- Individual nodes can be located with an error e when they send out signals;
- Signals sent from different devices can be distinguished.

2.2.3 Previous Work

The goal of the previous work in Qin *et al.* (2014) is to measure the anonymity of end-to-end communication relationships under anonymous communication envi-

ronments in MANETs. In order to achieve this goal, the time domain is sliced into multiple intervals (Δt_1, Δt_2, ... , Δt_K). In each interval, a network node can only be either a message sender or a message receiver. The amount of messages sent from every node in one interval is stored in a Point-to-Point Traffic Matrix $\mathbf{W}_i$. In this way, a total number K of such matrices ($\mathbf{W}_1$, $\mathbf{W}_2$, ..., $\mathbf{W}_K$) are generated. Based on this collection of information, a series of algorithms are proposed to derive an End-to-End Traffic Matrix R, which can be used for calculating the source/destination probability distributions and the end-to-end link probability distribution. However, all this work is based on the assumption that the information acquired in each Point-to-Point Traffic Matrix is accurate, i.e., the localization errors can be ignored. This assumption is too strong in practice that the original work could not be widely applied in real world. To solve this problem, a new approach is proposed in the following section to cope with the interference from localization errors.

2.3 Proposed Solution

Monitoring and analyzing individual nodes' communication traffic is desirable but costly, due to the concerns of computation overhead and control complexity when network scale is large. In this section, a clustering algorithm is proposed based on the system model introduced above. It aims to solve two urgent issues with traffic analysis in MANETs: localization errors and scalability. Localization errors make it impossible to locate the source of a signal when several nodes are too close to each other. For example, when the distance between two nodes is less than twice of the localization error e as shown in Figure 2.1, the monitoring system cannot find out which node sends out a signal from the overlapping area A. When a group of nodes are moving together, details such as individual nodes' communications do not provide much more useful information than the entire group communication, i.e. communications when

treating the whole group as one single unit. Therefore, a monitoring system can only monitor inter-group communications instead. In other words, it ignores inner-group communications and only focuses on signals transmitted around group borders.

To analyze group communication, it is proposed to divide the nodes into small groups and treat each individual group as an integrated node, which is called a super-node. The monitoring system then can focus on communication relations between super-nodes instead of individual nodes. Nodes that move around by themselves can be treated as a super-node with size of 1.

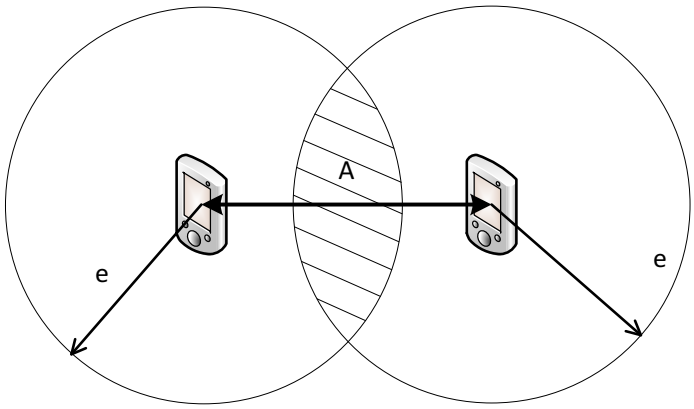

Figure 2.1: Localization Error Influence on Traffic Analysis.

To generate super-nodes, the proposed solution follows the idea of slicing time domain to form multiple time intervals in Qin *et al.* (2014). In each interval, closely related nodes are grouped into clusters. The metric for this close relationship is represented as Distance. Within each interval, the formation of every cluster is considered to be static. Cluster formation can be different in different intervals. The process for creating an interval consists of two steps: (1) Take a snapshot of the network whenever a data packet is captured; (2) Concatenate the snapshots to form a single interval. The criteria for forming a single interval in step (2) are:

- A super-node can be either a sender or a receiver within one interval, but it cannot be both;

- The formation does not change for any cluster within one interval.

Following these criteria, interval lengths are not necessarily equal. Control and management frames are excluded from triggering a snapshot since they are not closely related to communication relations. Super-nodes can be generated based on the node relationship information stored in clusters.

2.3.1 Clustering Algorithm

Before going into details of the clustering algorithm, a summary of notations is given in Table 2.1.

Unlike traditional agglomerative hierarchical clustering algorithms such as single-linkage clustering and complete-linkage clustering, a new distance metric is used in the proposed algorithm. In single-linkage clustering, the distance between the closest nodes in two clusters is used as the distance metric to determine how close two clusters are. Clusters are merged recursively. A Proximity Matrix (PM) is used for merging the closest clusters together. Elements in PM is proportional to the inverse of linkage criteria metric. That is to say, when the distance between two clusters is small, the corresponding element value is large. Thus, they are more likely to be merged in the next round. In the target application scenario, the same metric can be used because the shortest route between two clusters is likely to include the two closest nodes in them. However, geographical distance is not sufficient to represent the relationship between clusters. As stated before, clusters are used to represent the close relationship between nodes in network. If two nodes are geographically close while their transmission ranges are too short for them to communicate directly, then they are not closely related in this interval. (As shown in Figure 2.2A, a third node is needed to relay their communication.) To the contrary, if two nodes are geographically distant from each other but the corresponding transmission ranges are long enough

Table 2.1: Notations

Terms	Meaning
Proximity Matrix	a matrix for hierarchical clustering
Communication Distance Matrix	a matrix representing the number of hops between individual nodes
Accumulative Distance Matrix	a matrix representing accumulative average distances between individual nodes
Geographical Distance Matrix	a matrix representing the distances between individual nodes
Dendrogram	a data structure recording cluster formations
Accumulative Cluster Matrix	a matrix recording cluster formation inform--ation
Super-node Formation Array	an array recording super-node formation

to make direct communication possible, they are closely related (Figure 2.2B). Based on this observation, communication distance in hops is included as a factor in the definition of *Distance*. Another possible situation is that in one interval the distance between two clusters may be short for some reason while the overall node formation history exhibits a distant relationship between these clusters. In this situation, it is better to assign a longer distance between them for that interval. Following this idea, it is proposed to modify the distance metric with records from history to better represent the connections among mobile devices.

To add communication distance into the metric, the number of hops needs to be calculated between any two nodes in an interval. Since no information regarding the routing protocols is known, any shortest path algorithm for directed graph can be used instead. If one node a locates in another node b's transmission range as shown in Figure 2.3A, then there is a directed edge from b to a. The weight assigned to this edge is one. However, since b is out of a's transmission range, there is no directed edge from a to b. When two nodes cannot be connected through a series of edges, the communication distance is ∞. The communication distance in one interval is recorded in a matrix called Communication Distance Matrix (CDM) where $CDM(i, j)$ represents the number of hops between node N_i and N_j.

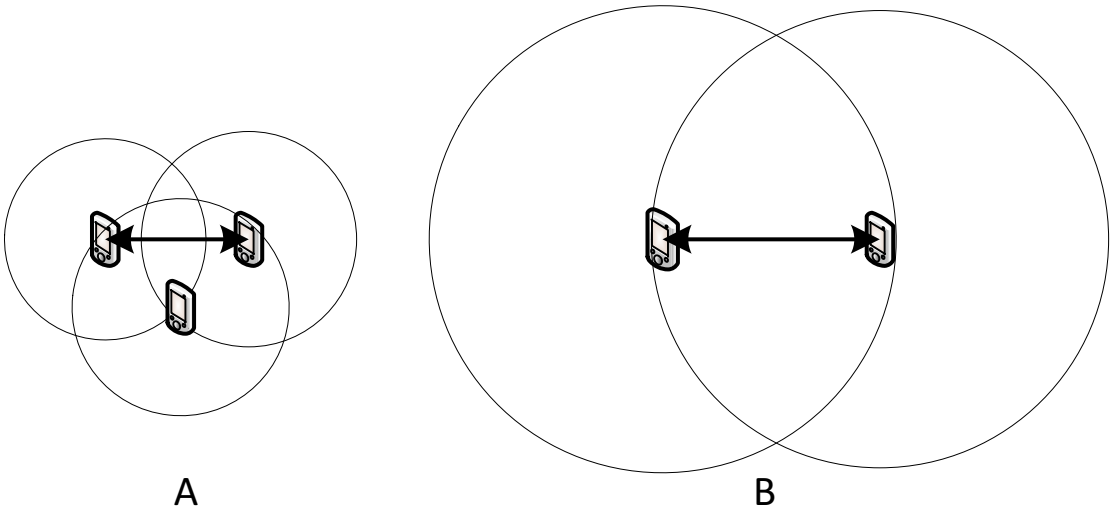

Figure 2.2: Different Transmission Distances Change Connectivities.

As stated in the network model section, the nodes are moving in group based patterns. If two nodes are closely located to each other in history, they are likely to belong to the same group. To model this property, a matrix called Accumulative Distance Matrix (ADM) is defined. In this matrix, each element $ADM(i, j)$ represents the accumulative average geographical distance between node N_i and node N_j. Initially, all the elements are set to 0. When a new interval is created, the distances between nodes in this interval are recorded in a matrix called Geographical Distance Matrix

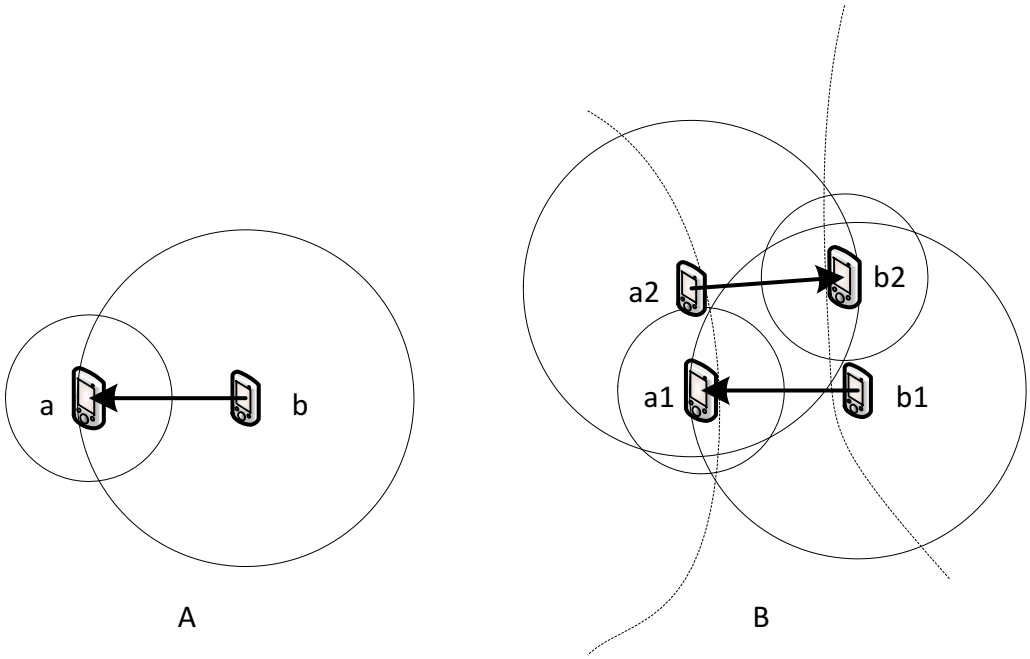

Figure 2.3: Asymmetric Communication Relations Influence Communication Distances.

(GDM). GDM represents the closeness among mobile nodes in terms of geographical distance during the current time interval.

The *Distance* used in this solution is defined as: $Distance(i,j) = \alpha GDM(i,j) + \beta ADM(i,j) + \gamma CDM(i,j)$, where α, β, and γ are chosen according to the mobility of the MANET. Elements of PM are generated and updated following Algorithm 1.

Algorithm 1 *PM Generation and Update*

1: In the k-th time interval, each element $PM(i,j) = 1/Distance(i,j)$;

2: Update $ADM(i,j) = (ADM(i,j) \times k + GDM(i,j))/(k+1)$, here k *starts from 2.*

Based on the distance metric and the system assumption, the hierarchical clustering algorithm can be modified as in Algorithm 2. The function $Distance(C_s, C_t)$ represents the distance under the proposed distance metric between cluster C_s and cluster C_t, which can be expressed as: $Distance(C_s, C_t) = \min(Distance(i,j)), N_i \in C_s, N_j \in C_t$. Note that the amount of nodes in different intervals may not be the same

since some nodes may not emit any signal during one or several intervals. However, the monitoring system is able to acquire the total number of distinct nodes in network based on radio fingerprints from all the intervals.

Algorithm 2 Modified Clustering Algorithm

1: Assign: clusters $C_i = N_i$, $i = 1, 2, ..., M$;

// M is the number of nodes in network, N_i denotes the i-th node in network

2: counter $c = M$;

3: **DO:**

4: $c = c - 1$;

5: Find the nearest clusters C_s and C_t;

6: **IF**$(Distance(C_s, C_t) < (\alpha + \beta)S(C_s, C_t) + \gamma T$

&& $|Between(C_s, C_t)| > R)$

7: Merge C_s and C_t;

8: **ELSE**

9: Skip merging C_s and C_t;

10: **END-IF-ELSE**

11: **UNTIL:** $c == 1$.

Here, R is a predefined threshold. $Between(C_s, C_t) = \{N_i || GeoDistance(N_i, C_s) - GeoDistance(N_i, C_t)| < \delta, |GeoDistance(N_i, C_s)| + |GeoDistance(N_i, C_t)| < \epsilon\}$, where δ and ϵ are predefined thresholds. It returns a set of nodes. Every node in this set satisfies that the distances between the node itself and the two clusters respectively are very close. (The difference between the two distances is less than δ.) Meanwhile, every node is very close to both clusters. (The sum of the two distances is less than ϵ.) The condition $|Between(C_s, C_t)| > R$ makes sure that there are sufficient individual nodes scattered between the two clusters when they are merged. When two distant clusters are connected with a line of sparse nodes, they will not be merged

(rectangle area in Figure 2.4). $S(C_s, C_t)$ is a conditional function: if no node in C_s can perform as b in Figure 2.3A to set up a directional one-hop link with any node in C_t or no node in C_t can do the similar thing, then it returns 0; otherwise, return the smaller one of the longest directional geographical distances between C_s and C_t. For instance, if $b_1 \rightarrow a_1$ is the longest one-hop link from any node in C_s to any node in C_t and $a_2 \rightarrow b_2$ is the longest from C_t to C_s as shown in Figure 2.3B, then $S(C_s, C_t) = \min(|b_1 \rightarrow a_1|, |a_2 \rightarrow b_2|)$. When the distance between two clusters is greater than the transmission range of nodes and no intermediate nodes can forward the traffic, they are not able to establish communications. Therefore, it is meaningless to merge them together as one single cluster. The threshold T represents an upper bound on the communication hops between two clusters when merging them. If they are too far away in hops, then they are not merged as well.

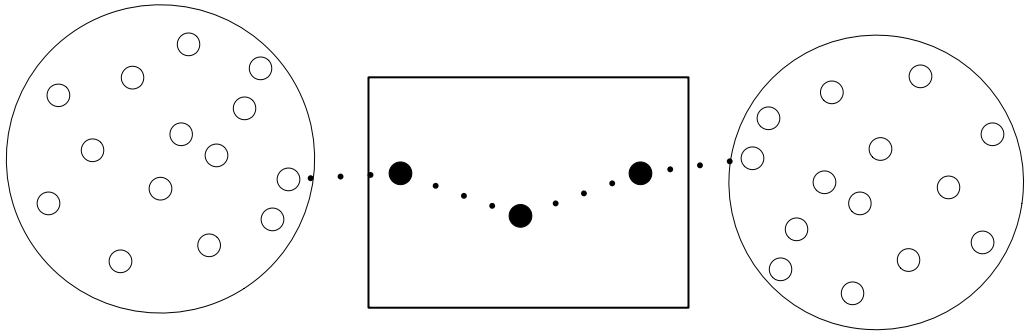

Figure 2.4: Sparse Node Connections Cannot be Used for Merging Clusters.

Outputs from hierarchical clustering algorithms are normally represented in dendrograms (Figure 2.5). The y axis represents at what distance two clusters are merged. As in the example of Figure 2.5, y_1 is the distance between cluster {A,B} and cluster {C}. y_2 is the distance between cluster {A} and {B}. It is easy to see that {A} and {B} are closer than either {A} and {C} or {B} and {C}. Dendrograms can accurately record the hierarchy of cluster generation as the distance grows.

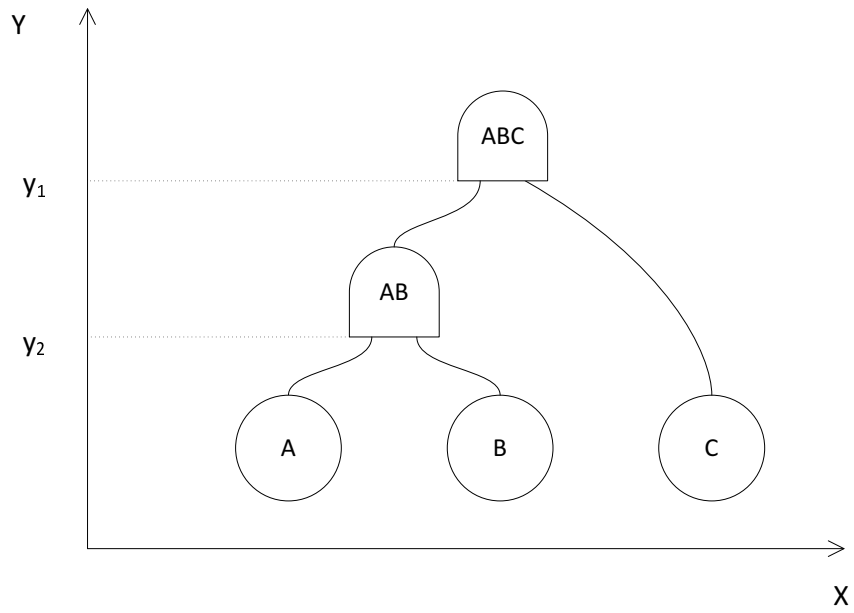

Figure 2.5: A Dendrogram to Record the Cluster Formations.

2.3.2 Super-node Generation

Clusters constructed from Algorithm 2 cannot be directly used as super-nodes. This is because in some intervals two or more groups may merge together as one cluster while in other intervals they are separate. Also, if a node emits a signal in one interval and keeps silent for several intervals before sending out another signal, even if it stays in the same group, the cluster formations are changed. Therefore, it is necessary to extract super-nodes from clusters in multiple time intervals. A matrix is defined and used to record the frequency that any two nodes locate in the same cluster. This matrix is called Accumulative Cluster Matrix (ACM). The method for generating an ACM is illustrated in Algorithm 3. It traverses through all intervals to record in $ACM(j,t)$ the number of intervals in which two nodes N_j and N_t are grouped in the same cluster.

Super-node formation can be deduced from ACM. The formation is represented in an array, which is named Super-node Formation Array (SFA). The method to create an SFA from ACM is depicted in Algorithm 4. In these algorithms, function

$Cluster(N_j)$ and $Super(N_j)$ respectively return the cluster and the super-node that node N_j belongs to. $SizeOf$ returns the amount of nodes in a cluster or a super-node.

Algorithm 3 ACM Generation

1: **FOR** $i = 0; i < K; i++$
   $//K$ is the total number of time intervals
2: **FOR** $j = 0; j < M; j++$
   $//M$ is the amount of nodes in network
3: **FOR** $t = 0; t < M, t \neq j; t++$
4: **IF** $Cluster(N_j) == Cluster(N_t)$
5: $ACM(j, t)++;$
6: **END-IF**
7: **END-FOR**
8: **IF** $SizeOf(Cluster(N_j)) == 1$
9: $ACM(j, j)++;$
10: **END-IF**
11: **END-FOR**
12: **END-FOR**

In Algorithm 4, γ and δ are predefined thresholds. γ restricts on the number of intervals in which any two nodes need to be in the same cluster so that they can be treated as in the same super-node. δ represents that if a node N_j locates in the same cluster with N_l for much more time intervals than it locates with any other node in the network, then it is treated as in the same super-node with N_l. If no such property exists, a node will be categorized as a super-node of itself.

2.4 Evaluation

In this section, the proposed approach is evaluated in terms of effectiveness and complexity. To analyze its effectiveness, the solution is applied in a trace using OMNeT++ omn (2013). For complexity analysis, complexity estimations are provided under the worst case scenario.

In evaluation setup, two moving groups are created with 4 nodes in each group within an area of $3000 \times 3000m^2$. Every node in a group is moving in the same direction with the same velocity ($10m/s$). For simplicity, it's assumed to have the same transmission power ($100mW$) on all the nodes. The estimated transmission range is $150m$. Two scenarios are chosen for evaluation. In Scenario 1, the groups are initially located at distant locations and are moving toward each other. In Scenario 2, they initially locate together and then move in opposite directions. These two scenarios represent typical cases concerning the relationship between two groups: merger and separation. Other possible moving scenarios between two groups can be represented by modifying and combining these two scenarios. To test the worst case on the proposed algorithm, the first criterion for determining the length of an interval, which relates to packet sending, is disabled during the entire process. Thus, fewer intervals are created. A single interval, instead of multiple intervals, is used to represent the fact that nodes stay in their respective groups for a long period of time. However, the proposed approach is able to overcome this problem by setting the value γ in Algorithm 4.

Evaluation results show that the proposed approach can successfully generate two super-nodes corresponding to the two groups respectively. The minimum number of time intervals needed for Scenario 1 is 1. This is because initially two groups are separate. During the first interval, the clustering algorithm can easily generate one

cluster for each group. As the groups are merging, cluster formations in the following intervals do not provide more accurate information than the first interval until they are completely separated. For Scenario 2, the minimum number of time intervals needed is 5. At interval 5, the two groups separate from each other completely and the condition in line 6 of algorithm 4 is satisfied. At this moment, the cluster formation correctly represents the group formation.

Additionally, the percentage that correct clusters are generated hierarchically in each interval using the geographical distance and the proposed distance metric are calculated respectively for Scenario 1. A cluster is correct if all its members belong to the same group. The results are shown in Figure 2.6. As shown in the graph, only two intervals are generated using the proposed metric. If Algorithm 4 is not used to integrate the cluster formation in different intervals together, it is almost impossible to infer the group formation solely based on the cluster formation in a single interval in Scenario 1 since the correct cluster ratio is less than 90% in 3 out of 7 intervals.

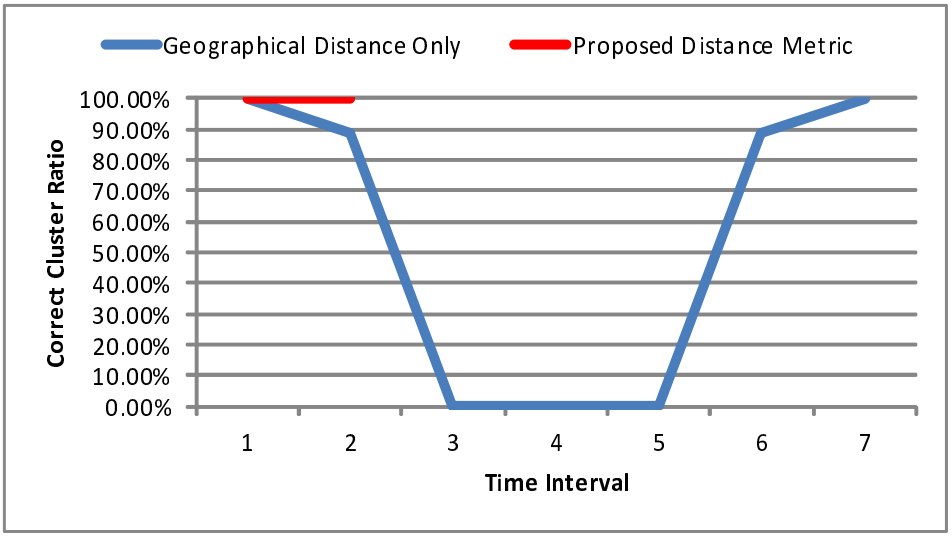

Figure 2.6: The Correct Cluster Ratios are Different using Different Metrics.

For complexity evaluation, the worst case scenario is the geographical position and the transmission coverage area of every node change during every interval. Therefore,

all the data stored in the matrices need to be updated. If M is used to denote the amount of nodes in network and K as the amount of intervals, the complexity for calculating GDM is $O(M^2)$. The number of ADM update operations is also $O(M^2)$. CDM can be calculated using Johnson's algorithm Johnson (1977) in $O(M^2 \log M)$. The complexity of $Between()$ function is less than $O(M^2)$. This is because the worst case is that the number of nodes left between two clusters is at the same order as M while the size of the two clusters are at the same order as well. Thus, the cost for Algorithm 2 is $O(M^3 \log M)$. The complexity of Algorithm 3 and Algorithm 4 are $O(M^2K)$ and $O(M^2)$ respectively.

It is clear that Algorithm 2 is the most complex algorithm. Considering normal hierarchical clustering algorithms have a complexity of $O(M^3)$, the proposed solution provides an effective modeling method without greatly increasing the overall complexity.

2.5 Conclusion

In this chapter, it is proposed to solve the problem of localization errors and the scalability issue with traffic analysis in MANETs by using super-nodes. Super-nodes are used to recover the group formation of network nodes. By slicing the time domain and grouping the nodes in each interval, information about geographical location and communication distance is recorded in clusters. Super-nodes are generated based on this information to represent the closeness between nodes. The proposed method provides an accurate approach for modeling MANET communication relations with a polynomial complexity.

Algorithm 4 SFA Generation

1: **FOR** $j = 0; j < M; j++$

$//M$ is the amount of nodes in network

2: **IF** $Super(N_j) == NULL$

3: Create a super-node for N_j;

4: **END-IF**

5: **FOR** $t = 0; t < M, t \neq j; t++$

6: **IF** $ACM(j,t) > \gamma K$

$//K$ is the total number of time intervals

7: Add N_t to $Super(N_j)$;

//So that $Super(N_t) == Super(N_j)$

8: Set $N_t.Processed = True$;

9: **END-IF**

10: **END-FOR**

11: **END-FOR**

12: **FOR** $j = 0; j < M; j++$

13: **IF** $N_j.Processed! = True$

14: Set l s.t. $ACM(j,l) = MAX(ACM(j,t))$, $1 \leq t \leq N$;

15: Set h s.t. $ACM(j,h) = MAX(ACM(j,t'))$, $1 \leq t' \leq N, t' \neq l$;

16: **IF** $ACM(j,l) - ACM(j,h) > \delta$

17: Add N_j to $Super(N_l)$;

18: **ELSE**

19: Create a super-node for N_j;

20: **END-IF-ELSE**

21: **END-IF**

22: **END-FOR**

Chapter 3

ATTRIBUTE-BASED ACCESS CONTROL IN MOBILE ICN

3.1 Introduction

In the previous chapter, the relation anonymity issue is discussed in MANET. In such traditional networking schemes, if a network entity wants to access some information content, it has to locate and connect to the server that provides such service following network routing protocols. As a result, the information is tightly associated with the location of the server. The entire network is centered around the connections between content consumers and content providers, making connection status an important factor to the network.

Witnessed by the fact that most of the network traffic is video sharing Cisco (2015), various ICN architectures Carzaniga *et al.* (2004); Koponen *et al.* (2007); Dannewitz *et al.* (2010); Fotiou *et al.* (2012b); named data (2015) are proposed. In ICN architecture, the core of networking is shifted from consumer-server connections to consumer-content connections. In this way, instead of identifying content owners' addresses, the network changes to identify authentic content copies scattered in network. Consumers do not need to know where copies of a content are located, i.e. the IP addresses of content owners. Content names are used to direct consumers to content copies. Content owners publish contents, which can be copied and distributed all over the network using network caches Psaras *et al.* (2012); Sun *et al.* (2014). Network caches are normally servers that are specifically designed for storage purpose or normal network entities with limited storage capabilities. This design enables contents being efficiently delivered to consumers with a higher efficiency. For example,

it is able to retrieve the nearest (according to some metrics) copy of a content to a consumer. In contrast, in the traditional Internet networking framework, a consumer gets a content only from its original owner.

Though the design of ICN is efficient in retrieving contents, it brings great challenges to security issues during content caching and retrieving. One of them is that traditional access control mechanisms cannot be easily enforced in such environment. This is because, in ICN, content owners and consumers are not directly connected. Content owners have no control over the distributed network caches. To enforce access control to the content, several frameworks have been proposed Fotiou *et al.* (2012a); Singh (2012). Most of them require additional authorities or secure communication channels in network to authenticate each content consumer. These schemes are sound but have too much reliance on traditional control schemes, making them inefficient in practice, especially in mobile ICN environment. Therefore, instead of enforcing the data access control mechanism on each caching server, a natural approach is to secure the content by enforcing the data access control through cryptographic approaches. If designed properly, only legitimate users who have proper cryptographic keys are able to access the data content. As each content is identified by the content name, it is easy for any network entity to access the content as long as such name is consistent among all the copies of the same content.

In this chapter, an attribute-based access control for ICN naming scheme is proposed. In this scheme, attributes defined by different authorities can be synchronized more efficiently than traditional approaches. Content consumers do not need to negotiate their attribute keys when they request contents from other authorities. To facilitate the application in mobile environment, the proposed approach aims to reduce the burden of a Trusted Third Party (TTP) and distribute part of its duties to several distributed attribute authorities.

The core of the proposed solution is an ABE-based naming scheme. This approach is inspired by Attribute Based Encryption (ABE) schemes Bethencourt *et al.* (2007); Yu *et al.* (2008); Lewko and Waters (2011). Instead of incorporating a set of additional components, it only requires one additional trusted third party (TTP) in the network. In addition, it can be seamlessly incorporated into existing flat-name ICN architectures. In the proposed approach, each network entity is assigned with a set of attributes with the help of a TTP according to their real identities. The access control policy for the content is based on combinations of the attributes in terms of AND and OR operations. This policy is enforced according to the content names instead of the contents. Moreover, privacy-preservation is provided for the content access policies , i.e., only legitimated content consumers are able to get part of the encryption policies and decrypt the data content. This feature can greatly improve the privacy protection on ICN data when they are distributed in the public domain. Especially, in wireless network, an access policy without privacy-preservation can be easily captured and monitored by any passive adversaries eavesdropping on the wireless channels. The proposed approach also provides a user with the capability to identify its eligibility of the accessed contents through the encrypted names before actually accessing and processing the data content. To further support the use of ontology in attribute management, the proposed scheme enables comparison between attributes, which gives the capability to rank attributes and associate different privileges accordingly. In summary, the expected contributions of this work can be listed as follows:

- It enables attribute rankings and access privilege management, making it flexible to construct a data access policy in real-world scenario. The content access policy is confidentially preserved. Ineligible consumers cannot derive the data access policies even if they collude together;

- It proposes a naming scheme for ICN network which combines the flexible attribute management solution with the privacy preserving access policy;
- It significantly reduces the computation and communication overhead for a potential consumer to determine his eligibility to access the content.

3.2 Related Work

In this chapter, an ABE-based scheme is proposed to enforce a secure access control mechanism in ICN network. Before introducing details of the proposed approach, related research results on ICN and ABE are presented respectively.

3.2.1 ICN Solutions

Several ICN architectures have been proposed in the past years. Although these approaches are different from each other in several aspects, the main idea is centered on information processing and management. Combined Broadcast and Content Based (CBCB) Routing Carzaniga *et al.* (2004) is a solution that runs on the application layer. It uses publish/subscribe scheme to publish contents. Each consumer broadcasts its interest in the form of attribute combinations. These interests are propagated through the network. At each router, the interests associated with an interface are updated in the form of predicates. When content is transferred through the network, the content is compared with the predicates on every interface to determine through which interfaces to forward the content.

Data Oriented Network Architecture (DONA) Koponen *et al.* (2007) is deployed above IP layer. The name of a content is in the form of $P : L$, where P represents the hash of the owner's public key, L is a unique label the owner assigns to the content. The owner registers the content into the name resolution system when it is ready to

publish. Consumers use the name resolution system to find the nearest copy of the content. The system returns with the content copy or the IP address of the content location. Network of Information (NetInf) Dannewitz *et al.* (2010) follows a similar naming scheme as DONA. But instead of using the owner's public key to generate the digest, it uses a separate pair of public/private keys for the content. Multi-level Distributed Hash Table (DHT) is used for name resolution purpose. A content owner needs to register its content in all the three levels of the DHT and content lookups are carried out from the lowest level upwards. If it is not successful, then a dedicated resolution system will be used for further assistance. Publish Subscribe Internet Technologies (PURSUIT) Fotiou *et al.* (2012b) is another solution that uses a similar naming scheme as DONA. However, it has a much different structure for retrieving content locations, which involves topology information and load balance. Besides, it uses Bloom filter for source oriented routing to forward content copies to the consumers.

Named Data Networking (NDN) named data (2015) doesn't specifically define the name structure. A name in NDN consists of multiple components, each of which can be a human-readable string or a digest of the content. Content providers are required to guarantee the uniqueness of name components. This solution uses names to execute a routing process that is similar to the current IP-based routing. Name tables, which are similar to route tables in IP network, maintain the prefix of names and the corresponding interfaces or data. In this way, a response to a content request can be the content itself. Also, this solution aims to provide a replacement to IP instead of being a layer above IP, which is different from approaches mentioned before.

Several research works have been conducted on applying NDN in mobile network environment. In Angius *et al.* (2012), the authors proposed a gossip algorithm to disseminate messages with a minimum number of transmissions. It is based on a

modification from traditional NDN solution. In Yu-Ting *et al.* (2014), a dedicated network architecture for mobile ad hoc ICN network is proposed. It supports both pull and push transport in multi-hop communications. Existing mobile ICN research works are mainly focused on lower-level networking mechanisms. For upper-level mechanisms, access control as an example, there is not much difference between a traditional ICN and a mobile ICN, except for the underlying networking related factors, such as mobility and mobility-related delay.

All these ICN solutions focus on the efficiency and security aspects of the network while access control to content and content privacy are not well addressed. In Fotiou *et al.* (2012a), an independent access control system is introduced to support the need in ICN. This system connects to the ICN structure through a component called the Relaying Party (RP). An additional component called Access Control Provider (ACP) is in charge of creating access policies and enforcing the policies to consumers' credentials. This system incorporates access control into ICN systems, but requires much more network interactions for a consumer to get the content. For content privacy purposes, Arianfar *et al.* (2011) proposed a design in which each file is divided into blocks. A block from the file is mixed with blocks from "cover" content using randomizing transformations and the generated mixture is published to the network. In this way, adversaries could not retrieve the original file easily. To recover the file, an authentic consumer needs to get more information related to the file from a secure channel. With such information, the consumer requests related chunks from the network to generate the original file. This approach meets the security and privacy requirement to some extent, but through a complicated process. The requirement for a secure channel is very difficult to satisfy in many ICN application scenarios.

3.2.2 ABE Schemes

ABE schemes originate from Identity-Based Encryption (IBE), which aims to use the user's id as the public key for asymmetric encryptions. After that, an ABE scheme named Ciphertext-Policy ABE (CP-ABE) Bethencourt *et al.* (2007) was introduced by J. Bethencourt *et al.* This scheme assigns each user with a set of attributes according to their real-life identities and roles. There is one private key component corresponding to each attribute for each user. A policy specifying under what conditions the ciphertext can be successfully decrypted is constructed by the encryptor. This policy is transmitted together with the ciphertext, but in plaintext form. In other words, it is exposed to the network channel. Users who do not possess a satisfactory combination of attributes are not able to decrypt the ciphertext. This scheme enables providing access control to individual messages. A content owner is able to specify the required attribute combinations without knowing the receivers' key credentials. In addition, this scheme is secure against collusion attackers.

The original CP-ABE scheme is a possible candidate for enforcing access control in ICN, but it is not a good solution in such scenarios. The reason why CP-ABE is not suitable for ICN usage is that the policy is transmitted in clear text. In traditional network, a user is authenticated before access is granted. However, once a content is published in ICN, the owner has no control on it. In this way, any network user who has access to the ciphertext is able to access the policy. Attackers can deduce the sensitivity of the message as well as inferring the identities of those who are involved in the message transmission. For example, a message encrypted with the policy $\{Chairman\}\, AND\, \{CEO\}$ from a hospital is highly likely to be more important and valuable than a message with policy $\{Nurse\}\, AND\, \{Intern\}$. Thus, attackers can easily identify the high-value users and concentrate attacks of different forms on them.

What is needed for CP-ABE is the capability to hide the policy into the ciphertext. For such purpose, several works Yu *et al.* (2008); Nishide *et al.* (2008) are proposed. An attacker cannot get any information about the policy even if it actually executes the decryption process. However, these solutions sacrifice efficiency for security in that any party that tries to decrypt the ciphertext will have to go through the entire decryption process which involves a heavy computation overhead. For instance, in Yu *et al.* (2008), the decryption process includes a bit-by-bit decryption for the decrypting party's ID.

To save computation resources for the unsatisfactory users, D. Huang *et al.* proposed a scheme Huang *et al.* (2010) to expose the policy attributes step by step. Only one attribute is exposed to the decrypter at one step. In this way, the decrypter is able to stop the decryption process as soon as it fails at a specific step. However, the price for such a feature is that one additional attribute, which is the one that fails the decrypter, is exposed. Besides, this approach supports AND-gates only, which limits the flexibility of the policy.

For attribute management purpose, it is desirable to enable the comparison between attributes so that nominal attributes can be mapped into ordinal values, e.g. $\{Nurse\} < \{Physician\}$. In Zhu *et al.* (2012); Wang *et al.* (2015), Y. Zhu *et al.* proposed an encryption scheme using interval comparisons based on bilinear mappings. In this chapter, the idea for interval comparisons is adopted and applied to hidden-policy attribute based encryption algorithms.

Comparatively, the proposed scheme achieves better flexibility by allowing the use of OR gates, and fully preserves the attribute policy in that ineligible nodes cannot get any information on any attribute in the policy that they do not have.

3.3 System and Models

In this section, a basic medical ICN framework, an overview of the attribute-based naming and access control model proposed in the rest of the chapter, and the attack model are presented. A preliminary mathematical description of CP-ABE scheme is also provided.

3.3.1 Application Scenario

In a typical ICN system, there are three roles: content owner, content consumer and content cache. A content owner creates the content and publishes it into the network. A consumer is a network entity that requests for the content. It gets the content with the help of the ICN infrastructure. A cache is an entity that keeps a copy of the content for a period of time in its own local storage so that whenever a request for the same content arrives, it directly responds to the request with a copy of the content to the consumer. All these three network roles are exchangeable for individual network entities. That is to say, a network entity can simultaneously be a publisher, a consumer and a cache for different contents. In the following, an example in medical care is used through out the rest of this chapter to show how the proposed scheme works. As shown in Figure 3.1, the content owner can be a Patient, a content consumer can be a Nurse or a Physician, and the content caches are servers storing encrypted contents.

In an ICN network, users get content names from a Name Searching Service (NSS) and use the names to get the content through a Name-based Routing (NR) system. A user gets content names from the NSS and the NR is able to retrieve the content based on the names. Details on how these two systems are implemented is out of the focus of this work. Interested readers can refer to Koponen *et al.* (2007), named data

(2015), and Carzaniga *et al.* (2004) for more information. Additionally, the proposed model includes a Trusted Third Party (TTP) that sets up Attribute-Based Access Control (ABAC) and Attribute-Based Encryption (ABE) related public parameters for the network. It also helps assigning and managing attributes to entities.

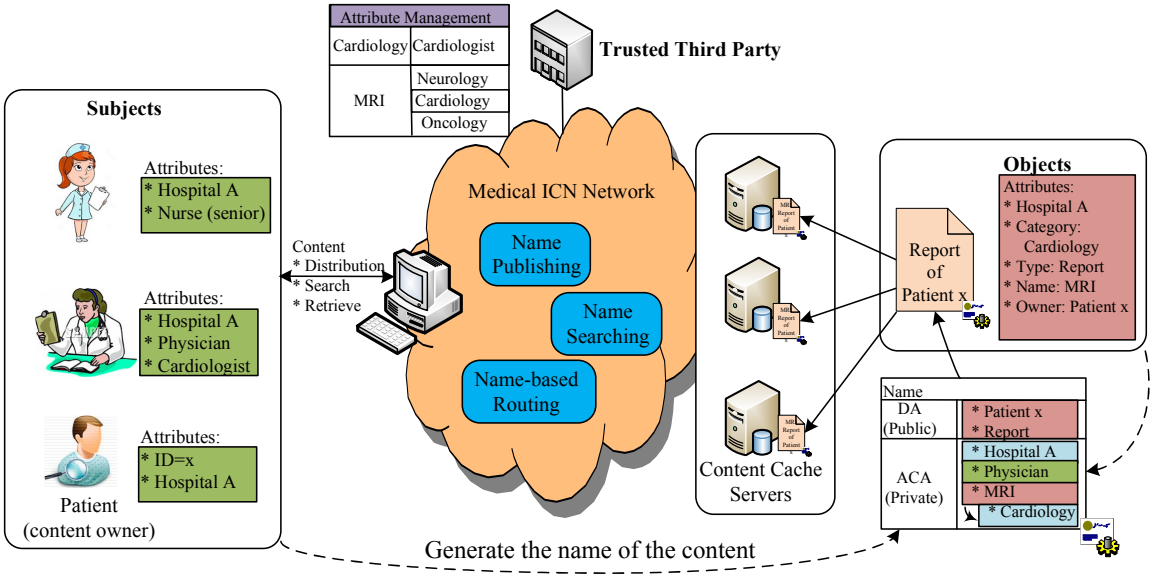

Figure 3.1: Basic ICN System Model.

In the proposed scheme, every network entity is associated with a unique identifier (UID) and a set of attributes. UID itself can be treated as a special attribute. A TTP is in charge of setting up global parameters for the entire network. An attribute (other than UIDs) can be defined and managed by any entity in network. But the definition and management process on an attribute should be carried out by the same entity. This entity is denoted as the authority of that attribute. As in this example, the attributes include: $\{HospitalA, Nurse, Physician, Cardiologist, MRI\}$. In the

proposed network model, multiple attribute authorities can be present at the same time. Thus, not only all the network users are organized in a distributed manner, the attribute authorities are also distributed. This property is supported by the specially-designed naming scheme proposed in this chapter. Each of the authorities is in charge of an independent and non-overlapping set of attributes.

A content owner is able to set up an access policy for its content under this scheme. The policy is represented as a combination of related attributes with AND and OR gates. For example, if a content owner wants to create a file that should be accessible only to people working as a $Physician$ or as a $Nurse$ at Hospital A, then the policy could be $\{A\}\, AND\, \{\{Physician\}\, OR\, \{Nurse\}\}$. In this way, the owner does not need to know explicitly who should access the content. He can identify the attributes and the combination so that as long as a consumer satisfies the policy, he is able to access the content. Any entity that does not satisfy the policy will not be able to access the data in this content.

3.3.2 Attribute-based Naming and Access Control

Attributes in an ICN network can be categorized into subject attributes and object attributes. As shown in Figure 3.1, attributes in green are subject attributes while the red attributes are object attributes of the report. When they are used in ICN, there are some relations between the subject attributes and object attributes. For example, $\{Cardiology\}$ and $\{Cardiologist\}$ are a subject attribute and an object attribute, respectively. They can be treated as equal since a cardiologist is assumed to always work on cardiology. Another example is $\{MRI\}$. As a useful tool, several medical subjects make use of MRI for diagnosis, such as neurology, cardiology, and oncology. To model such relationship, $\{Neurology, Cardiology, Oncology\}$ are defined as sub-attributes of $\{MRI\}$. When a content owner publishes the con-

tent, he decides which attributes are used for access control and which are for content search and content description. They are denoted as Access Control Attributes (ACAs) and Descriptive Attributes (DAs), respectively. As in the example of Fig. 3.1, $\{Hospital\ A, Physician, MRI, Cardiology\}$ are used as ACAs. $\{Patient\ x, Report\}$ are used as DAs. Thus, network entities only see that this content is a report of Patient x as the DAs are publicly search-able. The decision on ACA/DA classification is crucial to the privacy of the protected content and it's up to the content owner to make such decisions.

3.3.3 Comparable Attributes and Attribute Rankings

In addition to the above-mentioned attribute setups, comparison between attributes is also supported. To illustrate this property, the example policy in 3.3.1, $\{A\}\ AND\ \{\{Physician\}\ OR\ \{Nurse\}\}$, is used. If for some reason, a modification to the policy is needed as: all the staff working at hospital A that rank higher than nurse are allowed to access the file, then in the traditional approach, it is necessary to enumerate all the attributes that are allowed and construct a very complex policy as $\{A\}\ AND\ \{\{Physician\}\ OR\ \{Nurse\}\ OR\ ...\}$. However, if a comparison relationship is set up between $Physician$ and $Nurse$ as $Physician > Nurse$, meaning that a $Physician$ attribute includes all the privileges of a $Nurse$ attribute but with more that are not possessed by $Nurse$, then the original policy can be simplified. Suppose such a comparison relationship has been established with all the related occupation roles, then $\{A\}\ AND\ \{\{Physician\}\ OR\ \{Nurse\}\ OR\ ...\}$ can be reduced to $\{A\}\ AND\ \{Nurse\}$, which is easier for management purpose.

An Illustrative Example

In the example of Fig. 3.1, there are three subjects: a Nurse, a Physician, and a Patient. Their attributes are as shown in the figure. The patient publishes his MRI report in the network as the content. He, as the content owner, specifies an access policy as shown in Fig. 3.2 for the MRI report. Its object attributes are listed in Fig. 3.1. The content name is created following the procedure in Fig. 3.3, which will be further illustrated in Section 3.4.1. When the nurse tries to access this content, she can successfully use her $\{Hospital\ A\}$ attribute to decrypt the first node but will get stuck at $\{Physician\}$, meaning this content is not prepared for her. When the Physician accesses the content, she can successfully decrypt the entire decryption process from the leaf to the root level-by-level to reveal the random data encrypting key. Here, $\{MRI\}$ is substituted with $\{Cardiology\}$ since $\{Cardiology\}$ is a sub-attribute. This is shown with the arrow in Fig. 3.2. Also, $\{Cardiology\}$ equals to $\{Cardiologist\}$ in this case. Then, the Physician uses the NR system to get the nearest copy of the content and uses the random data encrypting key derived from the name to decrypt the MRI report.

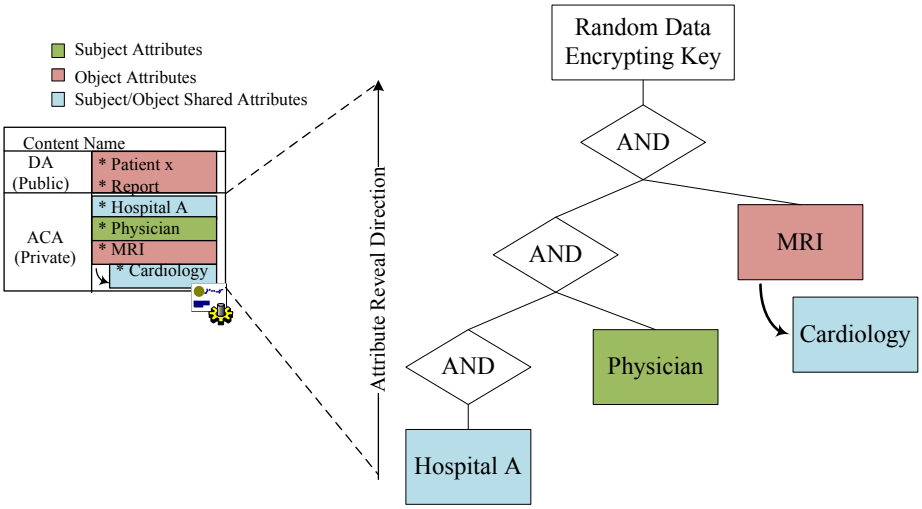

Figure 3.2: Creating a Content Name.

3.3.4 Attack Model

In order to guarantee the integrity of content, a digital digest signed by its owner is included in the content meta-data. Since data integrity is not the focus of this chapter, detailed information on this subject will not be provided. An attacker to the proposed system is assumed to focus on the proposed ABE scheme.

In the rest of this chapter, the attackers are assumed to have two primary goals in compromising the ICN access control scheme:

- acquiring unauthorized privilege to the data protected under the proposed ABE scheme;
- retrieving constitutional information of access policies to gain more information about the content, the owner, and the consumers. In other words, breaking the protection on the policies.

Sensitive information in this context includes but is not limited to the identity of the owner or consumers, the sensitivity of the content and the potential value of data in the content. For the first goal, attackers have to break the confidentiality mechanism of the protected data. Feasible methods include collusion attacks and vulnerability exploitation. The second attack goal is less important to an attacking party as a successful attack does not reveal information directly related to the protected secret. For the second goal, attackers need to analyze the proposed ABE-based scheme to identify possible ways to reveal the policy.

3.3.5 Preliminaries of ABE

The foundation of ABE-type algorithms is bilinear pairing computation. In this chapter, the design from Zhu *et al.* (2012) is adopted in terms of algebraic structure.

Suppose there are two groups: an additive group G_0 and a multiplicative group G_1 with a same order $n = sp'q'$, where p' and q' are two large prime numbers. A bilinear map is defined as $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. This map has three properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, for any $P, Q \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$;
- Nondegeneracy: $e(g, h) \neq 1$, where g and h are generators of $\mathbb{G}_0$;
- Efficiency: Computing the pairing can be efficiently achieved.

In CP-ABE, there are three types of keys: master key, public key and private key. A TTP is required to generate a set of public parameters and securely store the master key. The TTP will not be involved in the network communication. It can be offline all the time. The scheme of CP-ABE consists of four basic algorithms: **Setup**, **Encrypt**, **KeyGen** and **Decrypt**. In **Setup**, the TTP chooses two random exponents $\alpha, \beta \in Z_p$. A public key is formatted as $< G_0, g, h, f, e(g, g)^{\alpha} >$ while the master key is (β, g^{α}). Here $h = g^{\beta}$, $f = g^{\frac{1}{\beta}}$. The public key is published by the TTP before deployment. When a party wants to encrypt a message M, it runs the **Encrypt** algorithm. The inputs of this algorithm are the public key, the message M and a policy tree T. The output is a ciphertext. The **KeyGen** algorithm is used to generate private keys based on its inputs: the master key and a set of attributes. For each network node, the TTP runs the **KeyGen** algorithm once to generate a private key according to attributes assigned to that node. When a node receives the ciphertext, it runs the **Decrypt** algorithm to get the encrypted data. This algorithm takes the ciphertext and the node's private keys as inputs.

In traditional CP-ABE schemes, the TTP is involved in both when a node joins in the system (node management) and when an attribute is created and assigned to a node (attribute management). When the scale of the system is large, the TTP will

turn to a bottleneck for performance concerns. For this purpose, the proposed scheme aims to isolate the duty of node management and attribute management. It off-loads the attribute management functions to other entities.

3.4 ABE-based ICN Naming Scheme

In this section, the detailed design for the proposed ABE-based naming scheme in ICN network is illustrated. This scheme is based on a previous work Huang *et al.* (2010); Li *et al.* (2013).

3.4.1 Creating a Content

Initially, the TTP sets up global parameters for the entire network. Then, any entity in network can create attributes and assign them to other entities. Detailed process on how attributes are distributed is out of the scope of this work. Interested readers can refer to attribute allocation problem solutions for large scale networks such as Biswas *et al.* (2006). Once the attributes are assigned, network entities are able to create contents, i.e. start network communications. As shown in Fig. 3.3, when an entity publishes a file, as the content owner, it creates an access policy for the content. The policy is represented as a combination of related attributes with AND and OR gates. For example, if a content owner wants to create a record that is accessible only to physicians and nurses working at hospital A, the policy can be constructed as: $\{A\}\, AND\, \{\{Physician\}\, OR\, \{Nurse\}\}$. In this way, content owners do not need to know explicitly who should access the content before constructing the policy. Instead, all they need to do is to identify the attributes and the combinations of attributes for a qualified content user so that as long as a consumer satisfies the policy, it is able to access the content. Any entity who does not satisfy the policy will automatically be deprived of the privilege to access the information in this content.

No additional network participants are needed during this entire process to monitor the access control enforcement.

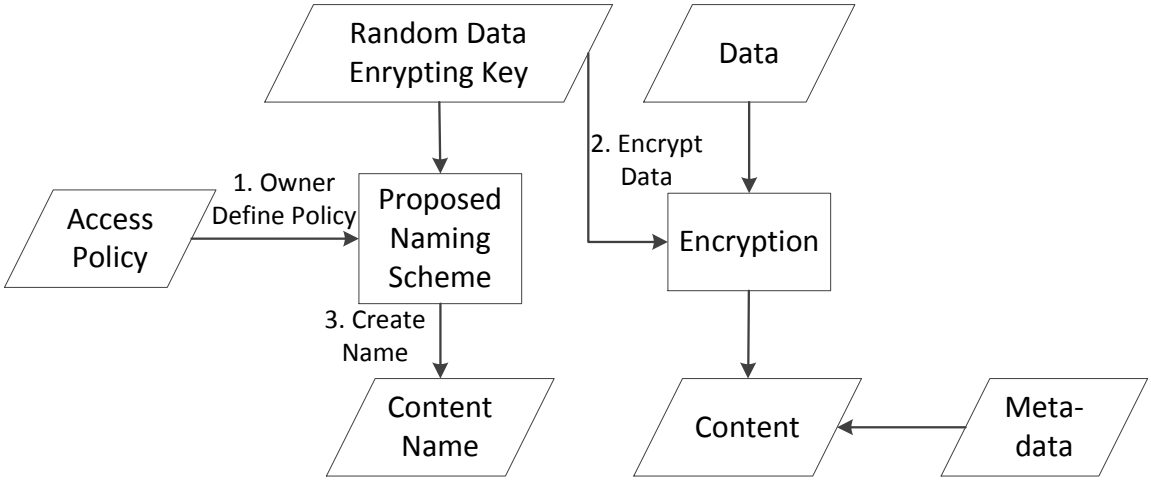

Figure 3.3: Creating a Content.

After creating the policy, the owner generates a random data encrypting key and uses it to encrypt the file. This encryption process can be any type of cryptosystem, the choice of which is not directly related to the proposed scheme. The encryption result is set as the data part of the content item. The meta-data part includes public parameters used for data integrity assurance and data decryption, like the type of cryptosystem used for the random data encrypting key.

The content owner creates a name for the content. He uses the proposed scheme to encrypt the random key under the policy he has specified. The result is used as the content name. Here, it is necessary to emphasize that the generated name hides the content access policies so that no one can get the entire policy from the name. In fact, the content name is a ciphertext after a series of encryption operations. It exhibits as a random sequence of bits to any viewer.

A consumer who needs this file is able to get a copy of the content by its name through the ICN network. Before he retrieves the content, he can use his own attributes to decrypt the name. If his attributes satisfy the hidden policy embedded in the name, he can get the random data-encrypting key protected in the name. The data of the content then can be decrypted using the random key to recover the original file. If a consumer cannot successfully decrypt the content name, it implies the consumer is not allowed to access the original file. Thus, even if he downloads the content, he still does not have the random data encrypting key to decrypt it. A benefit of the propose scheme is that a normal user can delay the downloading process of the content till he successfully decrypts the content name, which helps reduce the workload of underlying network.

3.4.2 ABE-based Naming Scheme

In this section, a composite order group G_0 with an order $n = p^2q^2$ is used, where p and q are two large prime numbers. In other words, the composite value s in Section 3.3.5 is set to pq. Two subgroups G_s and G_t of G_0 are chosen such that $s = pq$, $t = pq$, and G_s is orthogonal to G_t. Such composite-order group configuration is deliberately configured mainly because the proposed scheme is designed to support attribute rankings in G_s. The core idea of such configuration follows RSA conditions to enforce one-direction deduction between attribute values. This is why the value of s and t are set to be products of two large prime numbers. Details on such process will be illustrated in Section 3.4.3.

Attributes of an entity can be any value in strings. In CP-ABE, these values are converted into mathematical values by hash functions. In the proposed scheme, each attribute string A_i corresponds to a triplet (I_i, k_i, h_i), where $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$. S_i and T_i in **Algorithm 7** are assigned by the TTP. Their values are determined by the

generators under each sub-group and the value of h_i. The mapping from a string to such a three-tuple is determined by the authority of attribute A_i. An access policy can be expressed in Disjunctive Normal Form (DNF) of attributes. In each conjunctive clause of the DNF, the sequence of attributes is determined by the encrypting party, i.e. the content owner. The sequence of encrypting a conjunctive clause (encryption sequence) is opposite to the decryption sequence. To help identify the decryption thread, a public attribute A_{Pub} is defined in the scheme. Unlike other attributes, A_{Pub} is associated with a triplet $(S_{Pub}, T_{Pub}, I_{Pub})$, which are publicly known. For each conjunctive clause, the encryptor adds A_{Pub} at the end of the encryption sequence. In other words, the special attribute A_{Pub} is always the last attribute in encryption and the first attribute in decryption process. Additionally, the encryptor is required to simplify the DNF so as to reduce the size of attribute policy.

In the proposed scheme, a **GlobalSetup** algorithm is run by the TTP to generate global parameters for the system. For each node joining in the network, the TTP runs **NodeJoin** algorithm once to generate a unique secret for the node. The input of **NodeJoin** is the node's UID and the outputs are $\{D_{UID}, X_{Pub,UID}, Y_{Pub}, Z_{Pub,UID}\}$. For each attribute, the authority in charge runs the **AuthoritySetup** algorithm to generate secrets associated with that attribute. Besides, this naming scheme includes three more basic algorithms: **KeyGen**, **Encrypt**, and **Decrypt**. Once set up, the authority of an attribute runs **KeyGen** for each node carrying this attribute to allocate the inherent attribute secrets. **Encrypt** and **Decrypt** are used by encryptors and decrypters respectively for message processing.

The **GlobalSetup** algorithm generates global parameters $\{\mathbb{G}_s,\ \mathbb{G}_t,\ \varphi,\ \psi,\ \varphi^\beta,\ e(\varphi,\psi)^\alpha,\ Enc_k(\cdot),\ Dec_k(\cdot),\ (P_{Pub}, S_{Pub}, T_{Pub}),\ ROOT\}$, and global secrets $\{\beta,\ g^\alpha\}$, where α and β are random values and $Enc_k(\cdot)$, $Dec_k(\cdot)$ are a pair of symmetric encryption/decryption functions.

Algorithm 5 GlobalSetup

1: Choose two bilinear groups $\mathbb{G}_0$ and $\mathbb{G}_1$ with a composite order $n = p^2q^2$, where p and q are two large prime numbers. g is the generator of $\mathbb{G}_0$;

2: Choose two subgroups $\mathbb{G}_s$ and $\mathbb{G}_t$ of $\mathbb{G}_0$ such that: the order of $\mathbb{G}_s$ and $\mathbb{G}_t$ are both $n' = pq$; $\mathbb{G}_s$ and $\mathbb{G}_t$ are orthogonal to each other;

3: Choose two generators $\varphi \in \mathbb{G}_s$ and $\psi \in \mathbb{G}_t$;

4: Choose two random values $\alpha, \beta \in \mathbb{Z}_{n'}^*$;

5: Define a constant $ROOT \in \mathbb{G}_1$ as identification of the secret message;

6: Choose a pair of symmetric encryption functions $Enc_k(\cdot)$ and $Dec_k(\cdot)$ in $\mathbb{G}_1$;

7: Define a public attribute, $(S_{Pub}, T_{Pub}, I_{Pub}), S_{Pub} \in \mathbb{G}_s, T_{Pub} \in \mathbb{G}_t, I_{Pub} \in \mathbb{Z}_{n'}^*$;

8: The global parameters are $\{\mathbb{G}_s,\ \mathbb{G}_t,\ \varphi,\ \psi,\ \varphi^\beta,\ e(\varphi,\psi)^\alpha,\ Enc_k(\cdot),\ Dec_k(\cdot),\ (S_{Pub}, T_{Pub}, I_{Pub}),\ ROOT\}$, global secrets are $\{\beta,\ \psi^\alpha\}$.

The **NodeJoin** algorithm is defined as in **Algorithm 6**.

Each individual authority that manages an attribute A_i will have to run **AuthoritySetup** to set up attribute secrets.

The **KeyGen** algorithm generates private keys corresponding to each attribute for each node holding this attribute. It is defined in **Algorithm 8**. When the node receives keys from the authority, it checks if $L_{UID}^{P_{UID}} = T_{Pub}$ is true. If it's true, it updates P_{UID} with P_{UID}^2 and accepts the keys. This update is intended to defend against replay attack on L_{UID}. If not true, it will discard the keys.

The **Encrypt** algorithm works following the encryption sequence of each clause. In the following, each attribute is denoted from I_1 to I_m, m is the number of attributes in the clause. In the example of Fig. 3.2, $I_1 = MRI$, $I_2 = Physician$, $I_3 = Hospital\ A$, $I_4 = A_{Pub}$, $m = 4$. Any encryptor needs to choose a random value $s \in Z_p$, set $I_0 = s$ and follow **Algorithm 9**.

Algorithm 6 NodeJoin

1: For each node with UID in network, generate a random number $r_{UID} \in \mathbb{Z}^*_{n'}$;

2: Calculate $D_{UID} = \psi^{(\alpha + r_{UID})/\beta}$;

3: Calculate:

$$X_{Pub,UID} = \varphi^{r_{UID}} S_{Pub}^{r_{Pub}},$$

$$Y_{Pub} = \varphi^{r_{Pub}},$$

$$Z_{Pub,UID} = e(\varphi, \psi)^{r_{UID} I_{Pub}}.$$

where $r_{Pub} \in \mathbb{Z}^*_{n'}$ is a random number for each node;

4: Choose a random value $P_{UID} \in \mathbb{Z}^*_{n'}$;

5: Assign to the node $\{D_{UID}, X_{Pub,UID}, Y_{Pub}, Z_{Pub,UID}, P_{UID}\}$.

Algorithm 7 AuthoritySetup

1: For each attribute A_i, choose random numbers $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$;

2: For each attribute A_i, generate $S_i \in \mathbb{G}_s$ and $T_i \in \mathbb{G}_t$, where $S_i = \varphi^{h_i}$ and $T_i = \psi^{h_i}$.

The **Decrypt** algorithm works following the decryption sequence. Note that the first attribute in decryption sequence is always A_{Pub}. A decryption process follows **Algorithm 10**.

When **Decrypt** algorithm succeeds, S_k is the random data encyrpting key embedded in C.

3.4.3 Attribute Rankings

The proposed ABE scheme extends capabilities of traditional ABE schemes and is able to support comparison between values of the same attribute. In real world scenario, this means, for instance, two attribute values $Physician$ and $Nurse$ of attribute **Occupation** can be compared and have the relationship $Physician > Nurse$. In other words, it means that the $Physician$ attribute subsumes all the privileges

Algorithm 8 KeyGen

1: The authority passes I_i, S_i and T_i to TTP;

2: TTP computes and sends back to the authority:

$$X_{i,UID} = \varphi^{r_{UID}} S_i^{r_i},$$

$$Y_i = \varphi^{r_i},$$

$$Z_{i,UID} = e(\varphi, \psi)^{r_{UID} I_i},$$

$$L_{UID} = T_{Pub}^{1/P_{UID}}.$$

where $r_i \in \mathbb{Z}_{n'}^*$ is a random number;

3: The authority assigns $X_{i,UID}$, Y_i, $Z_{i,UID}$, and L_{UID} to the node together with I_i, h_i and k_i.

the $Nurse$ has, but the $Nurse$ does not have any of the additional privileges the $Physician$ has. Such capability is applicable and desirable when the privilege of the lower-ranking role ($Nurse$) is a subset of that of the higher-ranking role ($Physician$). In traditional ABE solutions, each attribute value ($Physician$ and $Nurse$ in the above example) corresponds to a set of cryptographic components that are designated for that specific attribute (**Occupation** in the example) of a specific user. Components for different values of the same attribute are not related. In other words, the key components of $Physician$ are independent to those of $Nurse$. To establish ranking relations between attribute values, certain connections need to be established between the corresponding key components. Specifically, a one-direction relation between values of the same attribute is supported in the proposed scheme. It allows a higher-ranking user ($Physician$) to be able to legally derive the corresponding lower-ranking role ($Nurse$) key components for himself. However, the lower-ranking role cannot derive anything useful regarding the higher-ranking role.

Algorithm 9 Encrypt

1: Calculate $C = Ke(\varphi, \psi)^{\alpha s}$, $C' = \varphi^{\beta s}$ and $C'' = Enc_K(ROOT)$;

2: For each attribute A_n, **if** a triplet $(C_{1,n}, C_{2,n}, C_{3,n})$ has already been calculated, move to the next attribute A_{n+1} and restart step 3 with A_{n+1}; **else**, **goto** step 4;

3: Choose a random number $l_n \in \mathbb{Z}^*_{n'}$;

4: Calculate:

$$C_{1,n} = \psi^{(I_{n-1}-I_n)l_n},$$

$$C_{2,n} = T_n^{(I_{n-1}-I_n)l_n},$$

$$C_{3,n} = (k_n l_n)^{-1}.$$

$1 \leq n \leq m$;

5: Calculate $C_{1,m+1} = \psi^{(I_m - I_{Pub})}$, $C_{2,m+1} = T_{Pub}^{(I_m - I_{Pub})}$.

Such capability can be achieved by deliberately assigning appropriate values in **KeyGen** algorithm. Specifically, as in the previous example, the scheme assigns h_P for *Physician* and h_N for *Nurse* such that $h_P = h^{\alpha_P}$, $h_N = h^{\alpha_N}$, $h \in \mathbb{Z}^*_{n'}$, and $\alpha_P < \alpha_N$. Thus, it is easy to derive $S_P = \varphi^{h_P}$ and $S_N = \varphi^{h_N}$. This is different from traditional ABE scheme, where both S_P and S_N are randomly chosen. Such difference is the connection that is established between comparable values (*Physician* and *Nurse*) of the same attribute (**Occupation**).

Recall when the order of $\mathbb{G}_s$ is defined, it is written as $n' = pq$, where p and q are two large prime numbers. In other words, n' is a composite number satisfying RSA algorithm requirements. If a user U_P is assigned with $S_P = \varphi^{h^{\alpha_P}}$, i.e. the key for *Physician*, the user is able to calculate the corresponding key S_N for *Nurse* as long as $\alpha_P < \alpha_N$. This process can be done as:

$$S_N = \varphi^{h^{\alpha_N}} = (\varphi^{h^{\alpha_P}})^{h^{\alpha_N - \alpha_P}} = (S_P)^{h^{\alpha_N - \alpha_P}} \tag{3.1}$$

Algorithm 10 Decrypt

1: Start from the public attribute A_{Pub};

2: For each attribute A_n that the decrypter possesses, compute:

$$\frac{Z_{n,UID_{dec}} \cdot e(X_{n,UID_{dec}}, (C_{1,n})^{k_n C_{3,n}})}{e(Y_n, (C_{2,n})^{k_n C_{3,n}})}$$

$$= e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})};$$

3: **If** $e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})}$ is the decrypter's private key, go to step 2 with attribute A_{n-1}; **else** go to step 4;

4: Calculate

$$S_k = C/(e(C', D_{UID})/e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})}).$$

if $Dec_{S_k}(C'') == ROOT$, **Success**; **else Failure**.

This means when attributes are assigned to U_P, it is optional to assign the value $h^{\alpha_N - \alpha_P}$ to the user together with S_P. Thus, when the user needs to decode some message dedicated for $Nurse$, he can easily calculate S_N following equation (3.1). However, if another user U_N has the attribute $Nurse$, he cannot deduce S_P following the same equation in a similar way. This is because in this case, $\alpha_P - \alpha_N < 0$. Under RSA assumption, h^{-1} cannot be efficiently computed due to the secrecy of n'.

A benefit of such extension to the original scheme is that it allows the ranking relations among attributes without incurring too much workload on TTP. Only eligible users, $Physician$ owners in this example, can use such capability and the value $h^{\alpha_N - \alpha_P}$ is only useful to eligible users.

It is necessary to clarify that the attribute authority can decide whether to assign the value $h^{\alpha_N - \alpha_P}$ to a specific $Physician$ owner or not. In other words, a $Physician$ owner does not automatically have the capability to derive his $Nurse$ components unless it acquires such value. Such derivation capability is carried out under the control of TTP.

With such knowledge, the TTP can assign two more values Δh and Δr to user U_P in **KeyGen** algorithms. When needed, the user can derive his key values corresponding to attribute $Nurse$ afterwards. The modified step 3 of **KeyGen** is as:

$$X_{P,UID} = \varphi^{r_{UID}} S_P^{r_P},$$

$$Y_P = \varphi^{r_P},$$

$$Z_{P,UID} = e(\varphi, \psi)^{r_{UID} I_P},$$

$$L_{UID} = T_{Pub}^{1/P_{UID}},$$

$$\Delta h = h^{(\alpha_N - \alpha_P) r_P},$$

$$\Delta r = \Delta h I_N / I_P.$$

Thus, the r_{UID} for U_P's $Nurse$ attribute is changed to $r'_{UID} = r_{UID} \Delta h$. Correspondingly, the following can be computed:

$$X_{N,UID} = (X_{P,UID})^{\Delta h} = \varphi^{r_{UID} \Delta h} S_N^{r_P} = \varphi^{r'_{UID}} S_N^{r_P},$$

$$Y_N = Y_P,$$

$$Z_{N,UID} = (Z_{P,UID})^{\Delta r} = (e(\varphi, \psi)^{r_{UID} I_P})^{\Delta h I_N / I_P}$$

$$= e(\varphi, \psi)^{r_{UID} \Delta h I_P} = e(\varphi, \psi)^{r'_{UID} I_P},$$

$$L'_{UID} = L_{UID}.$$

Here, it is necessary to point out that to make sure the values of h for two comparable attributes are the same, comparable attributes need to be managed by the same authority. This means one single authority defines the relative order between these attribute values. This requirement is reasonable in real-world scenario since in most cases a single authority (the hospital in this example) defines values of the same attribute (job position). It is rare to require two separate authorities to define separate values for the same attribute. Even if such conditions are needed, it can be easily handled with Ontology-based attribute management solutions.

3.4.4 Apply ABE-based Naming Scheme in ICN

With the above naming scheme, the following capabilities can be achieved in ICN scenarios:

- A content owner is able to specify the access control policy without knowing the consumers' keys;
- The policy confidentiality can be fully protected from being leaked to adversaries;
- Step-by-step attribute exposure is enforced for consumers to determine their eligibility efficiently in computation;
- Flexible attribute management is supported.

Using this scheme, any entity who wants to publish data contents needs to create the content following the procedures shown in Fig. 3.3.

The owner firstly creates a random symmetric key K. Then the data to be published is encrypted using K. The resulting ciphertext C is then used to generate a metadata of C. Both the metadata and C are parts of the final content. Then the owner needs to specify an access policy P of attributes, which identifies what attribute requirements an authentic consumer should satisfy. After that, the owner uses this policy to encrypt K following **Encrypt** algorithm. The result is used as the content name.

In this way, the owner does not need to know individual public keys of all the potential consumers in advance, which is required in traditional methods.

3.5 Performance Analysis and Evaluation

The performance improvement provided by the proposed scheme is evaluated in this section. In the following, the computation and communication performance is presented in two parts: real-world implementation and complexity analysis.

3.5.1 Evaluation of the Naming Scheme

In this section, the ABE-based naming scheme is evaluated from performance aspect. This includes analysis on its computation and communication (storage) overheads. The computation consumption analysis is carried out by comparing the proposed scheme with existing ABE schemes. The communication comparison is carried out on both the content name and the content itself respectively since they both are transmitted in the network.

From computation perspective, the time consumptions for key generation, encryption and decryption processes are tested. In real world application, the time consumption for a consumer to decrypt the content's name is much more important than that for other functions. This is because each content is encrypted once, but decrypted by multiple users for multiple times. In addition to testing the real-world time consumption for each function, a comparison is conducted on the decryption overhead with existing ABE solutions: CP-ABE Bethencourt *et al.* (2007), CN scheme Cheung and Newport (2007), NYO scheme (the 2nd construction in Nishide *et al.* (2008)), YRL scheme Yu *et al.* (2008) and GIE scheme Huang *et al.* (2010). The idea is to compare the number of most time-consuming operations needed in each scheme. Such comparison is carried out in complexity analysis.

3.5.2 Real-world Implementation

For real-world implementation, a machine with a four-core 2.80GHz processor and 4GB memory running Ubuntu 10.04 is used for experiment. Pairing-Based Cryptography (PBC) Library Lynn (2014) is used to handle the pairing computations. A type-A1 curve Lynn (2006) is generated using the parameter generating tools included in this library for the following tests. It randomly generates the prime numbers used for the curve, with a length of 512 bit for each of them.

Each operation is run for ten times for key generation, encryption and decryption (Fig. 3.4). Here the policies are set to be a conjunctive clause of different number (shown in x-axis) of attributes. This is because given a fixed number of attributes, this form requires the most time for computation. In other words, it directly represents the correlation between the number of attributes involved and the time needed for computations. The reason why the encryption function consumes more time when the number of attributes is small is that the cost for computing C in Algorithm 9 requires an additional pairing operation, which is independent to the number of attributes. When few attributes are involved, this additional pairing takes a high portion of the entire time consumption. This portion reduces as the attribute number grows, which explains why the time consumption for encryption eventually becomes the smallest among the three functions.

In theory, the time consumption should be linear to the number of attributes involved. The curve in Fig. 3.4 is not perfectly linear, but it meets the expected growing trend. There are several reasons why it is not strictly linear. Before decrypting the message attribute by attribute, in the implemented program, there are some necessary steps to initialize global parameters, read files and allocate memory space. Similarly, at the end of the algorithm, there are some clean-up work involved,

such as writing files and releasing memory space. Such time consumption is related to the number of attributes involved but not strictly proportional. Also, at step 4 of Decryption algorithm, there is one additional pairing operation. Thus, when the number of attributes is small, this additional operation takes more portion of the total time than when the number of attributes is large. If the possible variance introduced by system level factors, for instance the resource consumption from other processes, are also considered, the variance in the figures is reasonable in practice.

3.5.3 Complexity Comparison

For comparison purpose, every atomic operation is tested for fifty times and the average values are chosen as benchmarks for further comparison. Results of the experiment (Table 3.1) show that pairing operation takes longer than any other operations. Therefore, the comparison metric is set to be the number of pairing operations in decryption process.

Table 3.1: Time-consumption of Different Operations (in milliseconds)

	Pairing	Exponentiation	Multiplication	Inversion
Time	7.675	0.491	0.029	0.024

Following the above-mentioned idea, there are some terms that need to be defined: N_{attr} is used to denote the number of attributes a consumer has, N_{all} refers to the total number of attributes defined in the network ($N_{all} \gg N_{attr}$). The proposed naming scheme is denoted as ICN-ABE in the rest of this manuscript. Since the policy is publicly known in CP-ABE and CN, decrypting parties are able to decide what attributes to use in decryption. Therefore, for those who satisfy the policy, the time taken for decryption in CP-ABE is proportional to the number of attributes involved, which is denoted as N_{invo}, $N_{invo} \leqslant N_{attr}$. The time taken for a successful

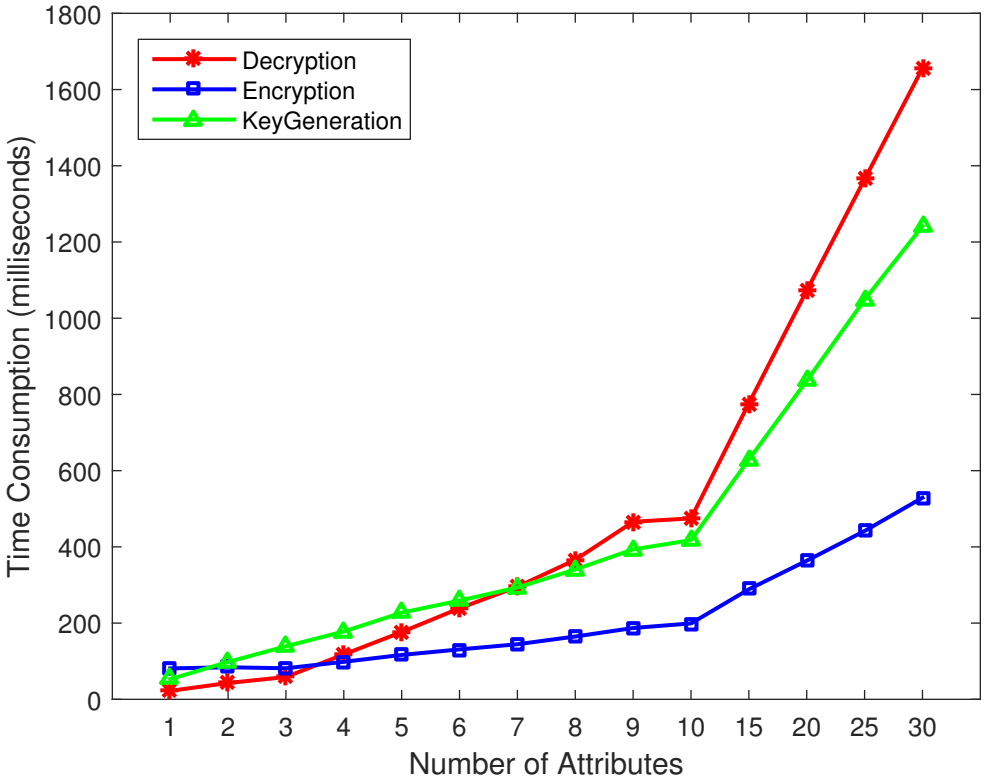

Figure 3.4: Computation Performance

decryption in CN is related to the number of attributes defined in the entire system. This is because each user is assigned with a value (Positive, Negative, and Wildcard) for every attribute defined in CN. It is obvious that unauthorized users would not bother to try decryption, which is why an alternative result in both schemes is that it takes 0 in time as the user would halt the decryption process.

An unauthorized user in GIE or ICN-ABE is not able to proceed with the decryption process if it cannot satisfy the next attribute. In this situation, N_{part} is used to denote the number of attributes that the consumer has already decrypted, where $N_{part} \leqslant N_{invo}$. Therefore, there are two possibilities for the computation cost in GIE and ICN-ABE, one for a successful decryption and the other for a failed one. Since OR-gate is not widely supported by all the ABE-schemes mentioned before, the performance is tested with policies consisting of attributes and AND-gates. Test

Table 3.2: Comparison of Computation Cost in Decryption

Scheme	Hidden Policy	Number of Pairings
CP-ABE	No	$2N_{invo} + 1$ or 0
CN	No	$N_{all} + 1$ or 0
NYO	Yes	$2N_{attr} + 1$
YRL	Yes	$2N_{attr} + 2$
GIE	Yes	$3N_{invo}$ or $3N_{part}$
ICN-ABE	Yes	$2N_{invo} + 1$ or $2N_{part}$

result is shown in Table 3.2. It is necessary to point out that in real world, N_{all} is much larger than N_{attr}. Therefore, CN scheme has the largest cost. Among all the anonymity schemes, GIE and the proposed scheme cost less than NYO and YRL. As a matter of fact, the cost of the proposed scheme is around 2 thirds of that of GIE.

To evaluate the communication costs, the size of content names are compared. The purpose to compare network names is to make sure that names generated by the proposed scheme do not consume much more storage space than existing solutions. In PBC library Lynn (2014), a data structure element_t with size of 8 bytes is used to represent an element. For the proposed scheme, a block of 24 bytes is needed to store the network name. Compared with this name size, a content in CBCB Carzaniga *et al.* (2004) is identified by a set of attributes with corresponding values. The size of this attribute set is determined by the content owners. Thus, it is reasonable to model the names as a human-readable string of an undetermined size. NDN named data (2015) shares a similar problem with the name size since the names in NDN also consists of a number of human-readable strings. As mentioned before, DONA

Koponen *et al.* (2007), NetInf Dannewitz *et al.* (2010) and PURSUIT Fotiou *et al.* (2012b) share the same naming scheme. Therefore, only the size of DONA's name is used for comparison. In Koponen *et al.* (2007), the size of a name is confined to 40 bytes in its protocol header. Thus, the size of network names in the proposed scheme is small enough to fit in existing ICN solutions.

The number of attributes used in ciphertext is denoted as N_{ciph}. For each attribute in the policy, the corresponding ciphertext consists of 2 elements from $\mathbb{G}_0$ and 1 element from $\mathbb{Z}_p$ in ICN-ABE. The total size of a ciphertext is $1\mathbb{G}_1 + (2N_{ciph}+4)\mathbb{G}_0 + N_{ciph}\mathbb{Z}_p$. This means the ciphertext consists of 1 element from $\mathbb{G}_1$, $2N_{ciph}+4$ elements from $\mathbb{G}_0$ and N_{ciph} elements from $\mathbb{Z}_p$. Comparison results are shown in Table 3.3. Here the sizes of attribute policy in CP-ABE and CN are not considered. CP-ABE has the smallest ciphertext size. Among the four schemes supporting anonymity, the ciphertext sizes in NYO and YRL are much larger than those in GIE and ICN-ABE. This is because these two schemes encrypt the ciphertext for all the attributes in the network. GIE and ICN-ABE are of the same order of magnitude with ICN-ABE performing better.

3.6 Security Analysis

From security perspective, the strength of the proposed scheme is analyzed based on the attack model presented in Section 3.3.4. For the first attack goal, a security theorem is provided with its corresponding security proof as in Section 3.6.2. For the second goal, the scheme is analyzed based on details of the algorithms.

Theorem 1 *Let G_0 and G_1 defined as in Section 3.4.4. For any adversary A, the advantage it can gain from the interaction with the security game defined in Section 3.6.1 is negligible.*

Table 3.3: Comparison of Ciphertext Size

Scheme	Ciphertext Size
CP-ABE	$1\mathbb{G}_1 + (2N_{ciph} + 1)\mathbb{G}_0$
CN	$1\mathbb{G}_1 + (N_{all} + 1)\mathbb{G}_0$
NYO	$\geqslant 1\mathbb{G}_1 + (2N_{all} + 1)\mathbb{G}_0$
YRL	$1\mathbb{G}_1 + (3N_{all} + 3)\mathbb{G}_0$
GIE	$N_{ciph}\mathbb{G}_1 + 3N_{ciph}\mathbb{G}_0$
ICN-ABE	$1\mathbb{G}_1 + (2N_{ciph} + 4)\mathbb{G}_0 + N_{ciph}\mathbb{Z}_p$

The proof for this theorem is provided in Section 3.6.2. In the proof, it is verified that the attacker cannot break the encryption algorithm to get any data exposed. Furthermore, it is also proved that attackers cannot conduct collusion attacks onto the system. This is because if collusion attacks are feasible, the adversary in the security game of Section 3.6.1 can overcome the constrain that no single user can satisfy the policy and still get the secret information decrypted. Thus, the attacker is able to gain a non-negligible advantage in this game.

For the second attack goal, the attacker will stop at the first attribute, A_k, that he doesn't own in the decryption process. If he can get to know this additional attribute, he must get it from step 3 in Algorithm 10. This means that the attacker possesses the secret key $Z_{i,UID}$ of the attribute A_k, which contradicts to the assumption that he does not possess such an attribute.

The rest of this section focuses on the proof of Theorem 1. Before going into details of the proof, the security model in terms of a security challenge game is presented in Section 3.6.1.

3.6.1 ABE Security Model

In this section, the focus is placed on the naming scheme, which can be modeled in the form of a game between a challenger and an adversary. The challenger simulates the operations of the TTP and the attribute authorities, while the adversary tries to impersonate as a number of normal network nodes. The game consists of the following five steps:

- **Setup.** The challenger runs the **GlobalSetup** algorithm and returns to the adversary the global parameters.

- **Phase 1.** The adversary can ask for a certain number of attribute keys in the name of a number of different users from the challenger. The amount of allowed keys and users are arbitrary. The challenger runs the **NodeJoin** algorithm for each user involved in the requests and returns the corresponding secret information. The adversary then plays in the roles of these users to request for attributes from the challenger. The challenger runs the **AuthoritySetup** algorithm to create parameters for authorities and runs the **KeyGen** algorithm to generate the corresponding attribute keys that are requested by the adversary on behalf of the authorities and the TTP. In other words, **KeyGen** in this game is conducted all by the challenger itself. The challenger creates new authorities only when it is necessary.

- **Challenge.** The adversary provides two messages M_0 and M_1 to the challenger together with an access policy A. A satisfies that none of the users created by the challenger has attributes satisfying A. It is possible that a combination of attributes belonging to different users who are impersonated by the adversary

can satisfy policy A. The challenger flips a coin b and encrypts M_b using A as:

$$C = \begin{cases} e(\varphi, \psi)^{\alpha s} & \text{if } b = 1 \\ e(\varphi, \psi)^{\theta} & \text{if } b = 0 \end{cases}$$

It then sends the ciphertext back to the adversary.

- **Phase 2.** The adversary can ask for more attributes and users from the challenger. But if any single user can gain satisfactory attribute combinations for A, the challenger aborts the game. Up to now, all the attributes or attribute keys mentioned in the game description refer to private keys. The adversary can always request for any public keys, which is only for encryption purpose.

- **Guess.** The adversary makes a guess b' on the real value of b.

The adversary's advantage in this game can be defined as $ADV = P[b' = b] - \frac{1}{2}$. The proposed scheme is secure if for all the polynomial time adversaries, the advantage is at most negligible in the game.

3.6.2 Security Proof Sketch

In this section, the sketch for security proof is provided following the structure in Bethencourt *et al.* (2007). Before going into details of the proof, the security game described in section 3.6.1 is modified. This modification follows the same idea as in Bethencourt *et al.* (2007) and it is intended to change from differentiating two random messages M_0, M_1 to differentiating $e(\varphi, \psi)^{\alpha s_j}, e(\varphi, \psi)^{\theta_j}$ so that the generated intermediate results can be represented using the four mappings that are to be introduced in this section. The goal of such modification is essentially to facilitate the subsequent security proof. To differentiate these two games, the one in section 3.6.1 is referred to as **Game1** and the modified game as **Game2**.

Modified Game

Game2 consists of five steps similar to **Game1**. The steps **Setup**, **Phase1**, and **Phase 2** are the same as in **Game1**. The **Challenge** step is different in that the challenger does not choose one message to construct the ciphertext C. Instead, it outputs C_j as:

$$C_j = \begin{cases} e(\varphi, \psi)^{\alpha s_j} & \text{if } b = 1 \\ e(\varphi, \psi)^{\theta_j} & \text{if } b = 0 \end{cases}$$

Here, all the θ_j are randomly chosen from $Z^*_{n'}$ following independent uniform distribution.

Suppose an adversary **adv1** in **Game1** has the advantage of ϵ, his corresponding adversary **adv2** in **Game2** can be constructed according to the following strategy:

- Forward all the messages between **adv1** and the challenger during **Setup**, **Phase1**, and **Phase 2**;
- In the **Challenge** step, **adv2** gets two messages M_0 and M_1 from **adv1** and the challenge C from the challenger. **adv2** flips a coin δ and sends $C' = M_\delta C$ to **adv1** as the challenge for **adv1** in **Game1**. **adv2** generates its guess based on the output δ' from **adv1**. If $\delta' = \delta$, then the guess is 1; otherwise, it is 0.

The advantage that **adv2** has in this game can be calculated as $\frac{\delta}{2}$.

In the following, it will be shown that no polynomial adversary can distinguish between $e(\varphi, \psi)^{\alpha s}$ and $e(\varphi, \psi)^{\theta}$. Therefore, no adversary can have non-negligible advantage in the security model.

Security Guarantee in the Modified Game

In this section, the proof sketch follows the generic group model introduced in Shoup (1997) and uses a simulator to model the modified security game between the challenger and the adversary. The simulator chooses random generators $\varphi \in G_s$ and $\psi \in G_t$. It then encodes any member in G_s and G_t to a random string following two mappings: $f_0, f_1 : \mathbb{Z}_{n'} \rightarrow \{0,1\}^{\lceil \log n' \rceil}$. It also encodes any member in G_1 to a random string in a similar way: $f_2 : \mathbb{Z}_n \rightarrow \{0,1\}^{\lceil \log n \rceil}$. One additional mapping f_3 is used to convert elements in $\mathbb{Z}^*_{n'}$ to string representations: $f_3 : \mathbb{Z}^*_{n'} \rightarrow \{0,1\}^{\lceil \log n' \rceil}$. These four mappings should be invertible so that the simulator and the adversary can map between the strings and the elements of corresponding algebraic structures in both directions. Four oracles are provided to the adversary by the simulator to simulate the group operations in G_s, G_t, G_1, and the pairing respectively. Only the string representations can be applied to the oracles. The results are returned from the simulator in such string representations as well. These oracles will strictly accept inputs from the same group, i.e. strict enforcement on the input from the same group for the respective group operations. The simulator plays the role as the challenger in the modified game.

- **Setup.** The simulator chooses G_s, G_t, G_1, e, φ, ψ, and random values α, β. It also defines the mappings f_0, f_1, f_2 and the four oracles mentioned above. The simulator chooses the public attribute parameters $I_{Pub} \in \mathbb{Z}^*_{n'}$, $S_{Pub} = f_0(\mu) \in G_s$, $T_{Pub} = f_1(\lambda) \in G_t$, and $ROOT \in G_1$, where λ and μ are random strings. The public parameters are G_s, G_t, $\varphi := f_0(1)$, $\psi := f_1(1)$, $\varphi^\beta := f_0(\beta)$, $e(\varphi,\psi)^\alpha := f_2(\alpha)$, $(S_{Pub}, T_{Pub}, I_{Pub})$, and $ROOT$.

- **Phase 1.** When the adversary runs **NodeJoin** for a new user with UID, the simulator generates a random number $r_{UID} \in \mathbb{Z}^*_{n'}$. It returns to the adversary

with $D_{UID} = f_1((\alpha + r_{UID})/\beta)$, $X_{Pub,UID} = f_0(r_{UID})f_0(\mu r_{Pub,UID}) = f_0(r_{UID} + \mu r_{Pub,UID})$, $Y_{Pub} = f_0(r_{Pub})$, and $Z_{Pub,UID} = f_2(r_{UID}I_{Pub})$, here $r_{Pub,UID} \in \mathbb{Z}^*_{n'}$ is a random number chosen by the simulator. When the adversary requests for a new attribute A_i that has not been used before, the simulator randomly chooses $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$ and $S_i = f_0(h_i) \in G_s$, $T_i = f_1(h_i) \in G_t$ to simulate the process for setting up an attribute authority for this new attribute. For each attribute key request made from the adversary, the simulator computes $X_{i,UID} = \varphi^{r_{UID}} S_i^{r_i} = f_0(r_{UID} + h_i r_i)$, $Y_i = \varphi^{r_i} = f_0(r_i)$, and $Z_{i,UID} = e(\varphi, \psi)^{r_{UID} I_i} = f_2(r_{UID} I_i)$, where r_i is a random number chosen from $\mathbb{Z}^*_{n'}$. The simulator passes all these values to the adversary as the attribute keys associated with A_i.

- **Challenge.** When the adversary asks for a challenge, the simulator flips a coin b and chooses a random value $s \in \mathbb{Z}^*_{n'}$. If $b = 1$, the simulator calculates $C = f_2(\alpha s)$; if $b = 0$, it picks a random value $s' \in \mathbb{Z}^*_{n'}$ and calculates $C = f_2(s')$. In addition, it calculates $C' = \varphi^{\beta s}$ and $C'' = Enc_K(ROOT)$. It also computes other components of the ciphertext following **Encrypt**: $C_{1,n} = f_1((I_{n-1} - I_n)l_n)$, $C_{2,n} = f_1(h_n(I_{n-1} - I_n)l_n)$, and $C_{3,n} = f_3((k_n t_n)^{-1})$, where $h_n \in \mathbb{Z}^*_{n'}$ is a random number chosen by the simulator.

- **Phase 2.** The simulator interacts with the adversary in a similar way as in **Phase 1** with the exception that the adversary could not acquire attribute keys enabling a single user to satisfy the access policy $\mathbb{A}$. The output of this step is similar to that of Phase 1 except that the simulator obtains more user IDs and attributes in this step.

From the above game, it can be seen that the adversary only acquires string representations of random values in $\mathbb{Z}^*_{n'}$, $\mathbb{Z}_n$ and combinations of these values. All the queries can be modeled as rational functions. It can further be assumed that

different terms always result in different string representations Bethencourt *et al.* (2007). As shown in Bethencourt *et al.* (2007), the probability that two terms share the same string representation is $O(q^2/n)$, where q is the number of queries made by the adversary. It is assumed in the rest of the proof that no such collision happens.

Now an argument can be made that the adversary's views are identically distributed between the two cases when $C = f_1(\alpha s)(b = 1)$ and when $C = f_1(s')(b = 0)$. As a matter of fact, what the adversary can view from the modified game with the simulator are independent elements that are uniformly chosen and the only operation that the adversary can do on these elements is to test if two of them are equal or not. Thus, the situation that the views of the adversary differ can only happen when there are two different terms ν_1 and ν_2 that are equal when $b = 1$. Since αs and s' only occur in group G_1, the results from f_1 cannot be paired. Queries by the adversary can only be in the form of additive terms. Then it can be derived: $\nu_1 - \nu_2 = \gamma\alpha s - \gamma' s'$, where γ is a constant. By transformation, it can be written as: $\nu_1 - \nu_2 + \gamma' s' = \gamma\alpha s$. This implies that by deliberately constructing a query $\nu_1 - \nu_2 + \gamma' s'$, the adversary may be able to get the value of $e(g, g)^{\gamma\alpha s}$. Now it needs to be proved that such a query cannot be constructed by the adversary based on the information it gets from the modified game.

In fact, the information that an adversary can acquire from this game is listed as in Table 3.4. This table excludes values related to L_{UID} as it is not related to αs. To construct the desired value, the adversary can map two elements from G_s and G_t into one element in G_1. He can also use elements in Z_n to change the exponentials. From this table, it can be easily seen that to obtain a value containing αs, the adversary can pair βs and $(\alpha + r_{UID})/\beta$ to get $\alpha s + r_{UID} s$ in G_1. In fact, this is the only way to get a term containing αs. But it is not feasible. Both βs and $(\alpha + r_{UID})/\beta$ belong to G_t while the pairing requires one element from G_s and one from G_t respectively.

A more detailed illustration for the above argument is that: by conducting the query on behalf of the users that the adversary has established in **Phase 1**, the adversary can get a polynomial $\gamma\alpha s + \sum_{UID \in U_{query}} \gamma r_{UID} s$, where U_{query} is the set of UIDs used by the adversary. To eliminate the second part in this polynomial, the adversary can use items in the table containing $I_{n-1} - I_n$ and r_{UID} to construct desirable polynomial. But this is impossible for the adversary under the game assumption because:

- Firstly, the adversary can't reconstruct s from either $t_n(I_{n-1} - I_n)h_n$ or $(I_{n-1} - I_n)h_n$ since the h_ns are chosen as random values for each attribute that it is impossible to get $s = \sum_{n \in P_a}(I_{n-1} - I_n) + I_{Pub}$ from them without peeling off the h_ns. Here, P_a represents the set of attributes satisfying the policy;

- Secondly, the adversary can't reconstruct s from I_{Pub} and I_i in Z_p. This is because no single user is assumed to satisfy the attribute policy that the adversary cannot reconstruct a valid attribute combination satisfying the policy. Thus, he cannot find the constitution of P_a for the equation $s = \sum(I_{n-1} - I_n) + I_{Pub}$.

- Thirdly, the item with r_{UID} cannot be canceled.

Therefore, based on the information an adversary can get from the proposed scheme, the attacker can not differentiate a random ciphertext from an authentic one. The security of the proposed scheme is proved. ■

Table 3.4: Query Information Accessible to the Adversary

μ	β	$r_{UID} + \mu r_{Pub,UID}$
r_{Pub}	h_i	$r_{UID} + h_i r_i$
r_i	$(I_{n-1} - I_n)h_n$	$t_n(I_{n-1} - I_n)h_n$
λ	$(\alpha + r_{UID})/\beta$	βs
h_i		
α	$r_{UID} I_{Pub}$	$r_{UID} I_i$
I_{Pub}	I_i	k_i
$(k_n t_n)^{-1}$	h_i	

3.7 Conclusion

In this chapter, a comprehensive access control solution for ICN network is proposed. This solution is based on a privacy-preserving ABE-based naming scheme. This scheme greatly reduces the communication and computation overhead compared to existing ABE solutions. Also, this scheme is designed in a public-key pattern, making it more flexible for attribute management. From security and privacy perspective, the ABE-based naming scheme achieves a high security level as CP-ABE, but with attribute anonymity protection for policy privacy and flexible attribute rankings. Experiments and analysis confirm the effectiveness of the proposed solution.

Chapter 4

ATTRIBUTE DELEGATION FOR ID-REVOCABLE ATTRIBUTE BASED ENCRYPTION

4.1 Introduction

Following the research work on Attribute Based Encryption (ABE) in the previous chapter, a notable issue came up in real world application scenarios. Under certain circumstances, the size of attribute policies would be very large in order to accommodate the need for a set of targeted users.

4.1.1 ID-revocable Attribute Based Encryption

The complex attribute policy issue comes from the fact that the policies in ABE aim to use combinations of attributes to represent or denote a set of individual users. When there are a large portion of shared attributes among these users, the policy can be constructed nice and simple. However, if such condition is difficult to satisfy, a very complex policy tree may be the only solution with attributes. The corresponding computation and communication/storage costs would therefore be too large for ABE to be applied in such circumstances, when it is compared with traditional cryptography algorithms or access control mechanisms.

One extreme example of such scenarios is when the targeted users are all the entities who own a certain attribute, except for a specific user. Following the example in health care, this attribute can be *Physician*. Under certain circumstances, a specific physician (who is denoted as John) at a hospital is rules out of a secret message that is prepared for all the other physicians at the hospital. There are several

possible cases when such scenario is true. For instance, John may be temporarily suspended from his duty. Or the content of the message has certain relation to John. Therefore, to avoid conflicts of interest, John is determined by hospital administrators not suitable for this message. Similar reasons can be listed on and on, none of which is rare in real world scenarios.

In traditional Attribute Based Encryption solutions, to avoid involving John as this message's intended receiver, the only option is to use attributes that are not owned by John. In this example, *Physician* is definitely not a choice. Instead, it is necessary to find out all the attribute combinations that are shared by part of the rest of physicians. This can be done level-by-level to rule out the targeted revocation user, John. For example, medical department, such as Cardiology and Pediatrics, can be used to identify people that are not from the same department as John. Other attribute genre will need to be used to further identify the physicians who work with John in the same department. Obviously, the resulting policy is much more complex than using a single attribute *Physician*.

The above-mentioned example is only for the cases when a single user needs to be excluded from a group. Cases can become much more complicated when the number of users for such exclusion is bigger, the size of the user group is larger, and the constitution of the group is very complicated, e.g. people from different organizations. Under such circumstances, relying solely on attributes to construct the policy is not an efficient solution. Intuitively, it can be easier and simpler if it is possible to rule out the limited number of outliers by their IDs.

Following this idea, several researchers at SNAC group of Arizona State University carried out a study on the feasibilities of enabling ID-revocation in Attribute Based Encryption schemes. Two schemes were eventually developed and is ready for publication. With such schemes, user IDs are embedded into the key components of

user attributes when they are assigned by the Trusted Third Party (TTP). These two schemes are referred to as the baseline schemes in the rest of this chapter.

4.1.2 Attribute Delegation Issue

The work in this chapter is based on such ID-revocation ABE scheme to further explore approaches to make it more applicable in real-world scenarios. Specifically, this chapter aims to solve the issue with key delegation features in such scheme.

Attribute delegation function is supported from the first Ciphertext Policy-ABE scheme Bethencourt *et al.* (2007). Most of the subsequent schemes proposed afterwards do support such delegation function as one of its five core functions. The delegation function allows a user with attribute set A to be able to generate and assign attribute keys of set A', $A' \subseteq A$, to other users.

The goal of such function is to help reduce the work load of the TTP in assigning attribute keys. However, as the f parameter is included in the public key PK in Bethencourt *et al.* (2007), there is no limitation on the number or the kind of attributes that a user is allowed to generate/delegate. Thus, any authentic user in the system is able to generate and assign unlimited amount of attribute keys that are in its own attribute set to other users, who are not guaranteed to be authentic users.

In real world application scenarios, this means that the authenticity of the intended attribute owners may be jeopardized. When the TTP assigns attribute keys to users, there is a authentication and authorization process associated with it. Such process protects the attributes from being assigned to wrong owners or even adversary parties. However, when a normal user performs the delegation function to generate attribute keys to other users, there is a risk that the users with the newly assigned keys may not be the ones that should be allowed to possess the attributes. In other words, the standards of trust at the TTP and a normal user may be greatly different.

Therefore, the delegation function provides a potential adversary with an additional way to gain an attribute that he is not authorized to own.

Following the health care example used in the previous chapter, if the TTP is assumed to be in charge of attribute assignment at hospital A, then the traditional delegation approach allows any *Physician* at this hospital to assign this attribute to any users not in this system. In real life, it is much easier to gain trust from a human user than the TTP through different approaches, social engineering as an example. Thus, it is desirable to design a delegation approach that can help prevent abuse of delegation functions in ABE schemes.

This chapter aims to conduct a preliminary study on such issue. An exploration of possible approaches and their relationships with each other is conducted. Based on that, a restricted delegation approach for the ID-revocable ABE scheme is proposed. The contribution of the work in this chapter can be summarized as:

- This is the first work, according to the author's knowledge, on how to control the use of delegation functions in ABE schemes. Existing ABE schemes use a component in the public key to enable delegation, which is in a lack of control on such capability;
- For the ID-revocable ABE scheme proposed before, a corresponding delegation scheme is proposed in this chapter to enforce control over the delegation function.

4.2 Related Work

In this section, related works are introduced in two parts: the delegation scheme in original ABE schemes and the ID-revocable ABE scheme that this chapter studies on.

4.2.1 Delegation Scheme in Original ABE Schemes

In the original CP-ABE scheme Bethencourt *et al.* (2007), the TTP sets up the entire system and publishes the public key to every user within the system. One of the component in the public key is $f = g^{1/\beta}$. This is used in the Delegate function to generate authentic secret key components for another user. The newly generated key components need to satisfy two conditions:

- It is authentic in that it can be used in encryption and decryption in the same way as a key directly obtained from the TTP;
- The newly generated key should be different from the keys belonging to the delegation user. Otherwise, it does not make much sense as a simple copy of the delegation user's key materials will do the same work.

In other words, one of the assumptions hold in the rest of this chapter is that no user will give a copy of his own keys to any other party in the system.

As the value of f is publicly known to every participants in the ABE system, any party who has a set of assigned attribute keys from the TTP is able to generate unlimited amount of new keys for these assigned attributes. The TTP itself has no control over such behavior. What is worse, the TTP does not even know who has been assigned with attributes through the delegation process. In this way, one extreme resulting scenario is that every participant is able to get a delegation key component for any attribute from a delegation party in the ABE system. In other words, the entire ABE system may fail if the delegation function is abused without proper restrictions.

4.2.2 ID-revocable ABE Scheme

As mentioned in Section 4.1.1, it is necessary to provide a flexible ABE scheme that supports the revocation of individual identities in policy construction. In this way, it provides an additional metric (identity) to existing attribute-only policy constructions. In some ABE schemes, users may also treat individual identities as attributes. However, these schemes uses identities in the same way as traditional attributes in policy construction. Therefore, they do not help simplify the attribute policy, but make it more complex.

In the rest of this chapter, identities are treated as a new set of metrics that are used differently from attributes. If attributes are treated as one dimension to describe a subject, identities can be viewed as another dimension that has no overlapping common set with attributes. Specifically, one identity is mutually exclusively owned by one single user. No two or more users share the same identity.

In the ID-revocable ABE Scheme that this chapter is based on, identities for each user is embedded in his assigned attribute key components. In other words, the TTP uses the users identities as one of the input to generate the attribute keys for corresponding users. A normal user cannot forge his attributes even though he knows his own identity. A benefit of such approach is that if a user leaks his attribute keys to other unauthorized users, his identity is embedded in the keys. This makes it possible to trace back to the person who initially leaked the keys. In traditional ABE scheme, it is impossible to do so.

When a certain party wants to encrypt a message, he not only creates an attribute policy for the encryption process. He also needs to provide a list of identities that should not be allowed to access the message. The combination of the attribute policy and the identity list determines the group of authorized users. Any user whose

identity is on this list is not able to decrypt the message. This is enabled through a mathematical trap that makes the denominator of a fraction become zero. For authorized users, the fraction denominator equals to the numerator, thus making the overall value equals to the identity element. The list of unauthorized users' identities are published together with the ciphertext to make sure anybody who receives a copy of the ciphertext is aware of such list.

4.3 Discussion on Attribute Delegation

Problem Statement: The goal of restricted attribute delegation is to block an attribute assignee from further acting as an delegation party for certain attributes assigned by the TTP.

Restricted delegation is in fact a very important feature for attribute delegations. Without a properly designed system to support it, authentic users may be able to grant privilege to untrusted parties. For instance, an authentic user may have a lower standard or an incomplete view of the standard for assigning a certain attribute. He may assign the attribute to an attacker based on his partial judgment. In other cases, an authentic user may accidentally or intentionally assign an attribute to unqualified users. Under such circumstances, the TTP is virtually deprived of his control over the attribute management.

How to view the attribute delegation capability is an important starting point. Under a general assumption, the entire delegation hierarchy is a tree structure rooted at the TTP. From the TTP's perspective, ruling out the attribute capability of a certain node is removing a subtree. As long as the root of the subtree is revoked, the children in that subtree are revoked. In other words, what is desired is an on-off switch associated with a certain node in the tree so that the TTP or any other parent to that node is able to toggle the switch.

Under different application scenarios, the requirements on restricted delegation are different. Here, mainly three scenarios are proposed and discussed: *breadth-first restriction*, *depth-first restriction*, and *hybrid restriction*.

For breadth-first restriction, the goal is to enable a parent to be able to block some of his children, which are referred to as blocked users, from further delegating an attribute or a set of attributes. All the children of the blocked users are deprived of the attribute. In other words, an on-off switch needs to be implemented for the parent so that he can make a binary decision on whether allowing further delegation of certain attributes or not. If the delegation relationship is viewed as a tree structure, the goal of this category is to disable a sub-tree of the original delegation tree. Such decision is made at the root of the sub-tree. As in the example of Figure 4.1, if the TTP in the delegation tree specifies blocking delegation of his child A, user A is allowed for possessing the attribute while none of the grandchildren (User D and E) is able to do so. The delegation capability of A is blocked though A still possesses such an attribute.

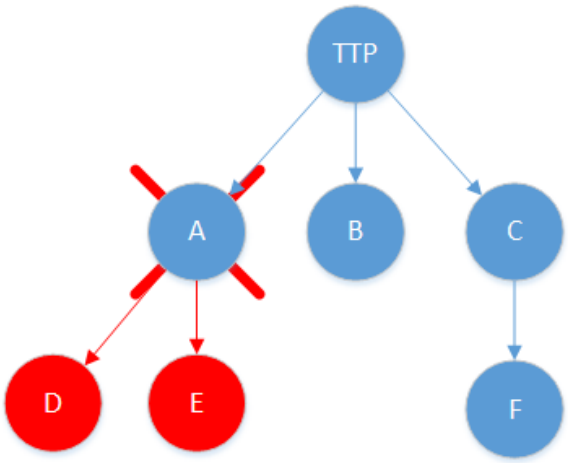

Figure 4.1: Breadth-First Restriction.

For depth-first restriction, the goal is to allow a parent to designate the maximum depth (or number of levels) of delegation relationship allowed in the delegation tree

structure. It differs from the previous category in that the parent can block not only the immediate children, but also grandchildren depending on how many levels he specifies. In the example of Figure 4.2, the root node TTP specifies that the target attribute can be delegated by its children for one more level, meaning only up to its grandchildren can have such attribute. In this case, the delegation capability of D and E are blocked, making it impossible for F to get such attribute from D.

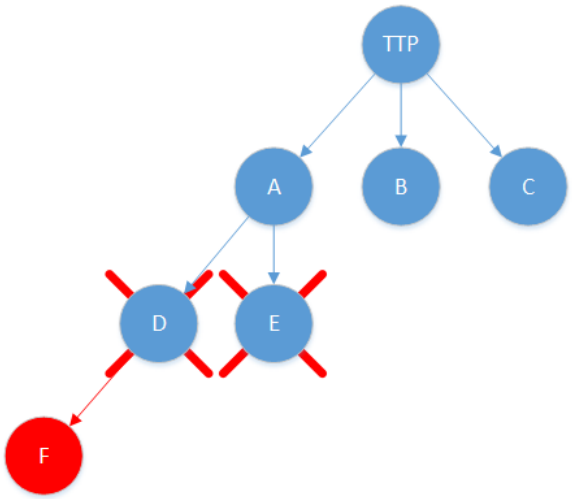

Figure 4.2: Depth-First Restriction.

It is necessary to stress that in both Figure 4.1 and 4.2, delegation restrictions can be enforced by any node that possesses the attribute (blue nodes in both figures), not necessarily the root of the whole tree, i.e. the TTP. Thus, at each node, the effect of delegation restrictions is a combination of all the rules set by every node along the path from itself to the root. For example, the effect at node F in Figure 4.1 is a combination of rules set by TTP and node C. Following such combination idea is the hybrid restriction.

For hybrid restriction, the goal is to block specific individual child or grandchild from being able to further delegate the target attributes. This category is more general, yet more difficult to achieve. By specifying black list of unwanted attribute delegates, the TTP is able to rule out the delegation capability of certain nodes.

This requirement is very similar to attribute revocation. The only difference is in the capability that is being revoked: attribute delegation v.s. attribute ownership. Hybrid restriction is the most flexible delegation control form that suits most of the application scenarios.

Before going into further details, there is one more assumption that needs to be emphasized. Obviously, if an attribute owner A sends an exact copy of his attribute components to an illegal attribute assignee B, the assignee does possess the same capability as an authentic owner of that attribute. Under such circumstances, it is obviously impossible to prevent A from enabling B with the attribute capability. However, such action will not be treated as a violation of the restricted attribute delegation capability this work is targeting for. This is because there is no new attribute components created during the entire process. It is more of a key management issue than attribute delegation. Thus, one assumption to follow is that:

Assumption:

A delegation key is guaranteed to share no key component with the original key it is derived from. This includes both the K part and the K_x part.

Starting from the simplicity point of view, it would be possible if a trap can be added to the original CP-ABE attribute key component. The trap plays as the on-off switch so that the attribute assigner is able to decide whether to allow the assignees to further delegate.

An interesting problem with attribute delegation is: how to blindly block a group of users from being able to use an attribute. The issue is that in the original CP-ABE scheme, there is no way to exclude an organization of users out of a certain decryption other than specifying a certain attribute that exclusively identifies that organization.

This incurs heavy attribute management workload in applications. Such issue comes from a lack of connections between the TTP and its attribute assignees in the attribute key components. With the proposed schemes, such connection is established through the ID value in the key component. Following this idea, anonymous attribute delegation is further illustrated with a use case.

A typical application scenario where the anonymous attribute delegation plays an important role is when the key of an organization A is compromised. Under such circumstances, every attribute key assigned under organization A needs to be revoked. Following the previous examples, an application scenario in health care is: the administrator at hospital A is compromised and therefore all the attributes within the hospital should be voided. Next time when Department of Health and Human Services(HHS) sends out a message, those attribute owners in hospital A should not be able to use their attributes for a successful decryption.

In the proposed preliminary scheme in the next section, this scenario maps A to a user who also has the capability for attribute delegation from TTP. Following the idea of delegation algorithm, a user's key is generated using A's ID as the ID value in the generated key components. Thus, when needed, all the keys can be anonymously revoked by TTP using A's ID, even though the key owners' IDs are not known to TTP (HHS in this case). With such capability, the TTP does not have to be aware of the individual users under each organization. Instead, when needed, the keys can be more efficiently revoked with the anonymous attribute delegation. The fact that the TTP does not need to know individual users' identities provides certain degree of anonymity to the system. All the TTP needs to know is the identity of the delegation party, in this example, the hospital.

The preliminary scheme solves the issue of enforcing permission on delegation functions by the TTP. However, it only supports one identity in each attribute key

component. If used as the delegation agent's identity, the individual users' identities are not included in the keys. Thus, it is impossible to revoke those identities when used with the id-revocable ABE scheme. What's more desirable is a scheme that is able to support revocation based on both delegation agency's identities and the attribute key users' identitites.

4.4 Proposed Delegation Scheme for ID-revocable ABE

Continuing from previous discussions, two categories of attribute delegation schemes are proposed. As a preliminary solution, the first scheme allows only one identity in the key attribute components for delegation. Based on this scheme, a further step on attribute delegation is that both individual users and delegation agents, organizations for instance, can be revoked at the same time. To enable such capability, two IDs need to be embedded, one for individual users and the other for delegation agents, into ciphertexts and attribute keys. Following this idea, a modification to the preliminary approach will be presented.

4.4.1 Preliminary Attribute Delegation Scheme

Before going into details of the preliminary delegation scheme, it is necessary to provide details of the scheme functions of the ID-revocable ABE scheme. It is a joint work by the author with his advisor and two other members of the research group. The scheme is as follows:

a. Setup$(\mathcal{U}, \mathcal{I})$

The *Setup* algorithm takes an attribute set $\mathcal{U}$ and an identity set $\mathcal{I}$ as inputs, where $|\mathcal{U}| = m$ and $|\mathcal{I}| = n$. It chooses a group $\mathbb{G}$ of prime order p, a generator g of the group, and m random group elements $h_1,\ h_2, \cdots, h_m \in \mathbb{G}$ that are associated with

the m attributes in the system. It also chooses random exponents $\alpha, b \in \mathbb{Z}_p$, which are kept as the master secrets for the entire system.

Therefore, the public key is in the form:

$$PK = \{g, g^b, g^{b^2}, e(g,g)^\alpha, h_1^b, \cdots, h_m^b\}.$$

The master secret key is in the form:

$$MSK = \{\alpha, b\}.$$

b. KeyGen(MSK, S, ID)

S is the attribute set of user $ID \in \mathcal{I}$. *KeyGen* algorithm chooses a random $t \in \mathbb{Z}_p$ and generates secret keys for user ID as follows:

$$SK = (K = g^\alpha g^{b^2 t}, \{K_x = (g^{b \cdot ID} h_x)^t\}_{\forall x \in S}, L = g^{-t}).$$

c. Encrypt(PK, (M, ρ), M, ID_j)

Encrypt algorithm takes as inputs: an LSSS access infrastructure (M, ρ) and the function ρ associates each row of M to corresponding attributes. ID_j is the identity to be revoked. Let M be an $l \times n'$ matrix. The *Encrypt* algorithm chooses a random vector $v = (s, y_2, \cdots, y_{n'}) \in \mathbb{Z}_p^{n'}$. These values will be used to share an encryption exponent s. For $x \in [1, l]$, it calculates $\lambda_x = v \cdot M_x$, where M_x is the vector corresponding to the x-th row of M. The *Encrypt* algorithm chooses random values $r_1, \cdots, r_l \in \mathbb{Z}_p$. Then, for message $\mathcal{M}$, the ciphertext is presented as follows:

$$C = \mathcal{M} e(g,g)^{\alpha s},$$

$$C_0 = g^s,$$

$$\hat{C} = \{C_k^* = g^{b \cdot \lambda_k}, C_k' = (g^{b^2 \cdot ID_j} h_{\rho(k)}^b)^{\lambda_k}\}_{\forall k = 1, \ldots, l}.$$

d. Decrypt($\mathbf{CT}, \mathbf{SK}$)

CT is the input ciphertext with access structure (M, ρ) and SK is a private key for an attribute set S:

$$CT = (C, C_0, \hat{C}, (M, \rho)).$$

Suppose that S satisfies the access structure and let $I \subset \{1, 2, ..., l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\Sigma_{i \in I} \omega_i \lambda_i = s$. If the identity ID embedded in SK is not equal to the revocation identity ID_j in the ciphertext, the decryption process can be performed as:

$$\frac{e(C_0, K)}{(\prod_{i \in I}[e(K_{\rho(i)}, C_i^*) \cdot e(L, C_i')]^{\omega_i})^{1/(ID - ID_j)}}$$

$$= \frac{e(g^s, g^\alpha g^{b^2 t})}{(\prod_{i \in I}[e((g^{b \cdot ID} h_{\rho(i)})^t, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, (g^{b^2 \cdot ID_j} h_{\rho(i)}^b)^{\lambda_i})]^{\omega_i})^{1/(ID - ID_j)}}$$

$$= \frac{e(g^s, g^\alpha) \cdot e(g^s, g^{b^2 t})}{(\prod_{i \in I}[e(g^{b \cdot ID \cdot t}, g^{b \cdot \lambda_i}) \cdot e(h_{\rho(i)}^t, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, g^{b^2 \cdot ID_j \cdot \lambda_i}) \cdot e(g^{-t}, h_{\rho(i)}^{b \cdot \lambda_i})]^{\omega_i})^{1/(ID - ID_j)}}$$

$$= \frac{e(g, g)^{\alpha s} \cdot e(g, g)^{b^2 st}}{(\prod_{i \in I}[e(g, g)^{b^2 t \lambda_i (ID - ID_j)}]^{\omega_i})^{1/(ID - ID_j)}}$$

$$= \frac{e(g, g)^{\alpha s} \cdot e(g, g)^{b^2 st}}{(\prod_{i \in I} e(g, g)^{b^2 t \lambda_i \omega_i})}$$

$$= \frac{e(g, g)^{\alpha s} \cdot e(g, g)^{b^2 st}}{e(g, g)^{b^2 t \sum_{i \in I} \lambda_i \omega_i}}$$

$$= e(g, g)^{\alpha s}$$

e. Preliminary Delegation

$$\tilde{K} = K \cdot g^{b^2 t'} = g^{\alpha} g^{b^2 (t+t')},$$

$$\tilde{K_x} = K_x^{((t+t')/t)} = (g^{b \cdot ID} h_x)^{(t+t')} = K_x \cdot g^{b \cdot ID \cdot t'} \cdot h_x^{t'},$$

$$L = g^{-t}.$$

Here, h_x is a random element selected by the TTP. It corresponds to one attribute defined in the cryptosystem. Without a knowledge of h_x, an entity cannot generate an authentic $\tilde{K_x}$ value. When TTP allows a certain party to delegate an attribute, it assigns h_x to the entity. The entity selects a random value t'. The other parts of $\tilde{SK}$ can be generated by the delegation agent based on public keys. Specifically, the component $\tilde{K}$ can be calculated as product of K and $g^{b^2 t'}$, where K is a part of the delegation agent's own key. $\tilde{K_x}$ can be calculated as the product of K_x, $g^{b \cdot ID \cdot t'}$, and $h_x^{t'}$. Here, ID is the identity of the delegation agent instead of the targeted user. g^b is one of the public parameters that are published by the TTP with Setup() function.

With this approach, the value h_x is what is needed to turn an ordinary user into a delegation agent. Revealing the h_x value to a user becomes a feasible approach for restricting the delegation capability of the delegation agent. By providing a false value, i.e. a mismatching t' to h_x, to the agent, he cannot generate authentic key components of the key.

4.4.2 Modified Revocation-supporting Schemes

As discussed before, the preliminary scheme cannot sufficiently meet the demand of revoking a delegation agency and/or an individual user at the same time. It is desirable to be able to embed both delegation agency's identity, which is named as

delegation identity or delegation ID in the rest of this chapter, and user's own identity, which is referred to as user identity or user ID, in the same policy.

An ideal situation is that the delegation ID is a prefix of the user ID and a powerful algorithm is able to revoke either a prefix or the entire length of the identity. However, as explained before, the success of the ID-revocable scheme is based on the fact that a denominator cannot be equal to zero. If prefix is used as in the perfect algorithm, such foundation fact cannot be satisfied. Thus, this "perfect" solution is not feasible in real-world application scenarios.

Comparatively, a less perfect, but more realistic solution is to assign two identities (a delegation ID and a user ID) into the attribute key components. A user is not allowed to successfully decrypt a ciphertext if both his delegation ID and his user ID are in the list of revoked identity list constructed by the encrypting party. In other words, this approach uses two lists, one for delegation IDs and the other for user IDs, to realize the expected revocation goal. A user is not revoked from a successful decryption if only his delegation ID or only his user ID is included in the lists. In other words, a pair of (delegation ID, user ID) is used to pin-point to individual users. This feature is supported based on the same fact that a denominator cannot be equal to zero in decryption process.

In the rest of this section, two schemes are introduced. The first scheme is based on the single-ID revocation ABE scheme that is a result of a joint research work by the author and his colleagues. The second one corresponds to the scheme from the same work for multiple ID revocation purpose. Both schemes presented below are significantly modified versions following the discussion as illustrated in the previous paragraphs.

4.4.3 Single-ID Two-Level Revocation for CP-ABE Scheme (SID2LR-CP-ABE)

In the following, five functions are presented for SID2LR-CP-ABE scheme. The first four functions are deep modifications of the previous research work, while the last function is presented for the first time for restricted delegation purpose.

a. Setup$(\mathcal{U}, \mathcal{I})$

The *Setup* algorithm is run by the TTP to set up global parameters for the entire cryptosystem. It takes an attribute set $\mathcal{U}$, where $|\mathcal{U}| = m$. Depending on how the keys for a specific user are generated, each user is assigned with a user identity from the set $\mathcal{I}$, $|\mathcal{I}| = n$, and a delegation identity, which represents the delegator's identity. It chooses a group $\mathbb{G}$ of prime order p, a generator g, and m random group elements $h_1,\ h_2, \cdots, h_m \in \mathbb{G}$ that are associated with the m attributes in the system. It also chooses random exponents $\alpha, \beta, b \in \mathbb{Z}_p$.

Therefore, the public key is published to all the entities in the system following the form:

$$PK = \{g, g^b, g^{b^2}, e(g,g)^\alpha, e(g,g)^\beta, h_1^b, \cdots, h_m^b\}.$$

The Master secret key is in the form:

$$MSK = \{\alpha, \beta, b\}.$$

They are the global secret that is only known to the TTP. A leak of the MSK will compromise the entire cryptosystem.

b. KeyGen$(\mathbf{MSK}, \mathbf{S}, \mathbf{ID})$

This algorithm is run once for each user ID by the TTP. It generates the corresponding attribute keys for the user. Here S is the attribute set of user $ID_u \in \mathcal{I}$, whose keys are

taken from delegator ID_d. *KeyGen* algorithm chooses a random $t \in \mathbb{Z}_p$ and generates secret keys for user ID as follows:

$$SK = (K = g^{(\alpha+\beta)} g^{b^2 t}, \{K_x = (g^{b \cdot ID_u} h_x)^t\}_{\forall x \in S}, \{K'_x = (g^{b \cdot ID_d} h_x)^t\}_{\forall x \in S}, L = g^{-t}).$$

c. Encrypt$(\mathbf{PK}, (\mathbf{M}, \rho), \mathcal{M}, \mathbf{ID_j})$

This is the encryption algorithm that can be run by any user who encrypts a certain message. *Encrypt* algorithm takes an LSSS access infrastructure (M, ρ) as input. The function ρ associates each row of M to corresponding attributes. ID_j is the user identity to be revoked, ID_h is the delegator identity to be revoked. Let M be an $l \times n'$ matrix. The *Encrypt* algorithm first chooses a random vector $v = (s, y_2, \cdots, y_{n'}) \in \mathbb{Z}_p^{n'}$. These values will be used to share an encryption exponent s. For $x \in [1, l]$, it calculates $\lambda_x = v \cdot M_x$, where M_x is the vector corresponding to the x-th row of M. The *Encrypt* algorithm chooses random $r_1, \cdots, r_l \in \mathbb{Z}_p$. Then, for message $\mathcal{M}$, the ciphertext is presented as follows:

$$C = \mathcal{M} e(g, g)^{(\alpha+\beta)s},$$

$$C_0 = g^s,$$

$$\hat{C}_u = \{C^*_{uk} = g^{b \cdot \lambda_k}, C'_{uk} = (g^{b^2 \cdot ID_j} h^b_{\rho(k)})^{\lambda_k}\}_{\forall k=1,\ldots,l},$$

$$\hat{C}_d = \{C^*_{dk} = g^{b \cdot \lambda_k}, C'_{dk} = (g^{b^2 \cdot ID_h} h^b_{\rho(k)})^{\lambda_k}\}_{\forall k=1,\ldots,l}.$$

d. Decrypt$(\mathbf{CT}, \mathbf{SK})$

This is the decryption algorithm that can be run by any user in this system. Here CT is the input ciphertext with access structure (M, ρ) and SK is a private key for a set S of attributes:

$$CT = (C, C_0, \hat{C}, (M, \rho)).$$

Suppose that S satisfies the access structure and let $I \subset \{1, 2, ..., l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\Sigma_{i \in I} \omega_i \lambda_i = s$. If the user identity ID_u combined in the SK is not equal to the revocation identity ID_j in the ciphertext and the delegation identity ID_d is not equal to ID_h in the ciphertext, the decryption process can be performed as:

$$
\begin{aligned}
& \frac{e(C_0, K)}{(\prod_{i \in I}[e(K_{\rho(i)}, C^*_{ui}) \cdot e(L, C'_{ui}) \cdot e(K'_{\rho(i)}, C^*_{di}) \cdot e(L, C'_{di})]^{\omega_i})^{\Delta_{j,h}}} \\
= & \frac{e(g^s, g^{(\alpha+\beta)} g^{b^2 t})}{(\prod_{i \in I}[e((g^{b \cdot ID_u} h_{\rho(i)})^t, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, (g^{b^2 \cdot ID_j} h^b_{\rho(i)})^{\lambda_i}) \cdot e((g^{b \cdot ID_d} h_{\rho(i)})^t, g^{b \cdot \lambda_i})} \\
& \overline{\cdot e(g^{-t}, (g^{b^2 \cdot ID_h} h^b_{\rho(i)})^{\lambda_i})]^{\omega_i})^{\Delta_{j,h}}} \\
= & \frac{e(g^s, g^\alpha) \cdot e(g^s, g^\beta) \cdot e(g^s, g^{b^2 t})}{(\prod_{i \in I}[e(g^{b \cdot ID_u \cdot t}, g^{b \cdot \lambda_i}) \cdot e(h^t_{\rho(i)}, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, g^{b^2 \cdot ID_j \cdot \lambda_i}) \cdot e(g^{-t}, h^{b \cdot \lambda_i}_{\rho(i)})} \\
& \overline{\cdot e(g^{b \cdot ID_d \cdot t}, g^{b \cdot \lambda_i}) \cdot e(h^t_{\rho(i)}, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, g^{b^2 \cdot ID_h \cdot \lambda_i}) \cdot e(g^{-t}, h^{b \cdot \lambda_i}_{\rho(i)})]^{\omega_i})^{\Delta_{j,h}}} \\
= & \frac{e(g, g)^{(\alpha+\beta)s} \cdot e(g, g)^{b^2 st}}{(\prod_{i \in I}[e(g^{b \cdot ID_u \cdot t}, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, g^{b^2 \cdot ID_j \cdot \lambda_i}) \cdot e(g^{b \cdot ID_d \cdot t}, g^{b \cdot \lambda_i}) \cdot e(g^{-t}, g^{b^2 \cdot ID_h \cdot \lambda_i})]^{\omega_i})^{\Delta_{j,h}}} \\
= & \frac{e(g, g)^{(\alpha+\beta)s} \cdot e(g, g)^{b^2 st}}{(\prod_{i \in I}[e(g, g)^{b^2 t \lambda_i (ID_u - ID_j)} \cdot e(g, g)^{b^2 t \lambda_i (ID_d - ID_h)}]^{\omega_i})^{\Delta_{j,h}}} \\
= & \frac{e(g, g)^{(\alpha+\beta)s} \cdot e(g, g)^{b^2 st}}{(\prod_{i \in I} e(g, g)^{b^2 t \lambda_i \omega_i})} \\
= & \frac{e(g, g)^{(\alpha+\beta)s} \cdot e(g, g)^{b^2 st}}{e(g, g)^{b^2 t \sum_{i \in I} \lambda_i \omega_i}} \\
= & \ e(g, g)^{(\alpha+\beta)s}, where \Delta_{j,h} = 1/(ID_u - ID_j + ID_d - ID_h)
\end{aligned}
$$

e. Delegate($\mathbf{SK}, \tilde{\mathbf{S}}$)

To create a set of authentic attribute keys for a user A from a delegator's own attribute key, it requires the delegator to acquire some key elements from the TTP. Here, $\tilde{S} \subset S$ is a subset of the attributes possessed by a delegator ID_d, which is also the set of

attributes for user A. The ID_u of the delegator in K_x of KeyGen algorithm is in fact the ID'_d in $\tilde{K}_x$. The goal is to create the key materials for ID'_u, i.e. user A, based on the key materials for the delegator. The generated attribute keys for user A are:

$$\tilde{K} = K \cdot g^{b^2 t'} = g^{(\alpha+\beta)} g^{b^2(t+t')},$$

$$\{\tilde{K_x} = (g^{b \cdot ID'_u} h_x)^{(t+t')}\}_{\forall x \in \tilde{S}},$$

$$\{\tilde{K'_x} = (g^{b \cdot ID'_d} h_x)^{(t+t')} = K_x \cdot g^{b \cdot ID'_d t'} \cdot h_x^{t'} = K_x \cdot g^{b \cdot ID_u t'} \cdot h_x^{t'}\}_{\forall x \in \tilde{S}},$$

$$L = g^{-t}.$$

As a delegation agent, the required information from the new user A is his identity ID'_u. The delegator chooses a random value t' for the user and calculates $\tilde{K}$ as product of the delegator's own K and $g^{b^2 t'}$. Here, g^{b^2} is a public parameter. Similar to the preliminary scheme, h_x is required from the TTP to convert a normal user to a delegation agent. With t' and h_x, a delegator is able to calculate $\tilde{K'_x}$. Here, the ID_u is the identity of the delegator. $\tilde{K_x}$ can be viewed as the product of $(g^{b \cdot ID'_u} h_x)^t$ and $(g^{b \cdot ID'_u} h_x)^{t'}$. It's not difficult for the delegator to calculate the value of $(g^{b \cdot ID'_u} h_x)^{t'}$. However, for $(g^{b \cdot ID'_u} h_x)^t$, the TTP has two options to enable the delegation capability. The first option is that the delegator sends the user's identity ID'_u to the TTP and the TTP calculates $(g^{b \cdot ID'_u} h_x)^t$ for the delegator. The second approach is that the TTP assigns the value pair (g^{bt}, h_x^t) to the delegator to allow the delegation capability. The first option allows the TTP to control not only which attribute an agent is allowed to delegate, but also which user is allowed to take the attribute from the delegator. The cost for such advanced capability is the computation cost happened at the TTP. The

second approach only allows the TTP to control which attribute an agent is allowed to delegate. In either case, without the information from the TTP, the delegation capability will be blocked. In other words, without such information, an ordinary user cannot generate legit attribute key components for any other user in the system.

4.4.4 Multiple-ID Two-Level Revocation for CP-ABE Scheme (MID2LR-CP-ABE)

The above-mentioned scheme is designed for enabling restricted delegation capability while revoking a single user, i.e. a single pair of (delegation ID, user ID). To further extend such a capability in real world application scenarios, a modified scheme is presented below to support multiple ID revocation.

a. Setup$(\mathcal{U}, \mathcal{I})$

This algorithm is run to establish the entire cryptosystem. The TTP generates global security parameters through this algorithm. The algorithm takes an attribute set $\mathcal{U}$ and an identity set $\mathcal{I}$ as input, where $|\mathcal{U}| = m$ and $|\mathcal{I}| = n$. It chooses a group $\mathbb{G}$ of prime order p, a generator g and m random group elements $h_1, h_2, \cdots, h_m \in \mathbb{G}$ that are associated with the m attributes in the system. It also chooses random exponents $\alpha, \beta, b \in \mathbb{Z}_p$, which are kept as global secret keys.

Therefore, the public keys are output as:

$$PK = \{g, g^b, g^{b^2}, e(g,g)^\alpha, e(g,g)^\beta, h_1^b, \cdots, h_m^b\}.$$

These keys are published globally to every participant in this system.

The Master secret key is:

$$MSK = \{\alpha, \beta, b\}.$$

MSK is stored securely at the TTP and is only known to the TTP itself. A leak of this key will directly compromise the entire system.

b. KeyGen(MSK, S, ID)

This algorithm is run by the TTP to generate individual attribute keys for each user. The TTP runs this algorithm for each member of the system when they join in it. In this algorithm, S is the attribute set of user $ID_u \in \mathcal{I}$. ID_d is the delegation ID of the same user. The algorithm chooses a random $t \in \mathbb{Z}_p$ and derives the secret keys for the user as follows:

$$SK = (K = g^{(\alpha+\beta)} g^{b^2 t}, \{K_x = (g^{b \cdot ID_u} h_x)^t\}_{\forall x \in S}, \{K'_x = (g^{b \cdot ID_d} h_x)^t\}_{\forall x \in S}, L = g^{-t}).$$

Here, K_x and K'_x are generated corresponding to each attribute that the user owns. Thus, the overall size of SK differs from user to user and it is closely related to the number of attributes a user has.

c. Encrypt(PK, (M, ρ), M, S)

This algorithm is designed for every user in the system. It takes an LSSS access infrastructure (M, ρ) and a function ρ associates rows of M to attributes as input. Let M be an $l \times n'$ matrix. The algorithm chooses a random vector $v = (s, y_2, \cdots, y_{n'}) \in \mathbb{Z}_p^{n'}$. These values will be used to share the encryption exponent s. For $x \in [1, l]$, it calculates $\lambda_x = v \cdot M_x$, where M_x is the vector corresponding to the x-th row of M. Let $r = |S|$. ID_{uj} is used to denote the j-th user identity in S and ID_{dj} denotes the j-th delegation identity. The algorithm chooses random $\mu_1, ..., \mu_r \in \mathbb{Z}_p$ such that

$\mu = \mu_1 + ... + \mu_r$. The generated ciphertext is in the form as follows:

$$C = \mathcal{M}e(g,g)^{(\alpha+\beta)s\mu}, C_0 = g^{s\mu}$$

$$C^*_{1,1} = g^{b\cdot\lambda_1\mu_1}, C_{u1,1}{}' = (g^{b^2\cdot ID_{u1}}h^b_{\rho(1)})^{\lambda_1\mu_1}, C_{d1,1}{}' = (g^{b^2\cdot ID_{d1}}h^b_{\rho(1)})^{\lambda_1\mu_1}\cdots$$

$$C^*_{l,1} = g^{b\cdot\lambda_l\mu_1}, C_{ul,1}{}' = (g^{b^2\cdot ID_{u1}}h^b_{\rho(l)})^{\lambda_l\mu_1}, C_{dl,1}{}' = (g^{b^2\cdot ID_{d1}}h^b_{\rho(l)})^{\lambda_l\mu_1};$$

$$C^*_{1,2} = g^{b\cdot\lambda_1\mu_2}, C_{u1,2}{}' = (g^{b^2\cdot ID_{u2}}h^b_{\rho(1)})^{\lambda_1\mu_2}, C_{d1,2}{}' = (g^{b^2\cdot ID_{d2}}h^b_{\rho(1)})^{\lambda_1\mu_2}\cdots$$

$$C^*_{l,2} = g^{b\cdot\lambda_l\mu_2}, C_{ul,2}{}' = (g^{b^2\cdot ID_{u2}}h^b_{\rho(l)})^{\lambda_l\mu_2}, C_{dl,2}{}' = (g^{b^2\cdot ID_{d2}}h^b_{\rho(l)})^{\lambda_l\mu_2};$$

$$\cdots$$

$$C^*_{1,r} = g^{b\cdot\lambda_1\mu_r}, C_{u1,r}{}' = (g^{b^2\cdot ID_{ur}}h^b_{\rho(1)})^{\lambda_1\mu_r}, C_{d1,r}{}' = (g^{b^2\cdot ID_{dr}}h^b_{\rho(1)})^{\lambda_1\mu_r}\cdots$$

$$C^*_{l,r} = g^{b\cdot\lambda_l\mu_r}, C_{ul,r}{}' = (g^{b^2\cdot ID_{ur}}h^b_{\rho(l)})^{\lambda_l\mu_r}, C_{dl,r}{}' = (g^{b^2\cdot ID_{dr}}h^b_{\rho(l)})^{\lambda_l\mu_r}.$$

d. Decrypt($\mathbf{CT}, \mathbf{SK}$)

In the decryption process, CT is the input ciphertext with access structure (M, ρ) and SK is a private key for an attribute set S. Suppose that S satisfies the access structure and let $I \subset \{1, 2, ..., l\}$ be defined as $I = \{i : \rho(i) \in S\}$. Let $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\Sigma_{i\in I}\omega_i\lambda_i = s$. If the identity ID_u embedded in the SK is not equal to the revocation identity ID_{uj} in the ciphertext and the identity ID_d embedded in the SK is not equal to the revocation identity ID_{dj} in the ciphertext, the decryption process can be carried out as:

$$
\begin{aligned}
& \frac{e(C_0, K)}{\prod_{i\in I}(\prod_{j=1}^{r}[e(K_{\rho(i)}, C^*_{i,j}) \cdot e(L, C'_{ui,j}) \cdot e(K'_{\rho(i)}, C^*_{i,j}) \cdot e(L, C'_{di,j})]^{\Delta_j})^{\omega_i}} \\
= & \frac{e(g^{s\mu}, g^{\alpha+\beta} g^{b^2 t})}{\prod_{i\in I}(\prod_{j=1}^{r}[e((g^{b\cdot ID_u} h_{\rho(i)})^t, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, (g^{b^2\cdot ID_{uj}} h^b_{\rho(i)})^{\lambda_i\mu_j})} \\
& \overline{\cdot e((g^{b\cdot ID_d} h_{\rho(i)})^t, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, (g^{b^2\cdot ID_{dj}} h^b_{\rho(i)})^{\lambda_i\mu_j})]^{\Delta_j})^{\omega_i}} \\
= & \frac{e(g^{s\mu}, g^{\alpha}) \cdot e(g^{s\mu}, g^{\beta}) \cdot e(g^{s\mu}, g^{b^2 t})}{\prod_{i\in I}(\prod_{j=1}^{r}[e(g^{b\cdot ID_u\cdot t}, g^{b\cdot\lambda_i\mu_j}) \cdot e(h^t_{\rho(i)}, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, g^{b^2\cdot ID_{uj}\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, h^{b\cdot\lambda_i\mu_j}_{\rho(i)})} \\
& \overline{\cdot e(g^{b\cdot ID_d\cdot t}, g^{b\cdot\lambda_i\mu_j}) \cdot e(h^t_{\rho(i)}, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, g^{b^2\cdot ID_{dj}\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, h^{b\cdot\lambda_i\mu_j}_{\rho(i)})]^{\Delta_j})^{\omega_i}} \\
= & \frac{e(g,g)^{\alpha s\mu} \cdot e(g,g)^{\beta s\mu} \cdot e(g,g)^{b^2 s\mu t}}{\prod_{i\in I}(\prod_{j=1}^{r}[e(g^{b\cdot ID_u\cdot t}, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, g^{b^2\cdot ID_{uj}\cdot\lambda_i\mu_j})} \\
& \overline{\cdot e(g^{b\cdot ID_d\cdot t}, g^{b\cdot\lambda_i\mu_j}) \cdot e(g^{-t}, g^{b^2\cdot ID_{dj}\cdot\lambda_i\mu_j})]^{\Delta_j})^{\omega_i}} \\
= & \frac{e(g,g)^{\alpha s\mu} \cdot e(g,g)^{\beta s\mu} \cdot e(g,g)^{b^2 s\mu t}}{\prod_{i\in I}(\prod_{j=1}^{r}[e(g,g)^{b^2 t\lambda_i\mu_j(ID_u - ID_{uj})} \cdot e(g,g)^{b^2 t\lambda_i\mu_j(ID_d - ID_{dj})}]^{\Delta_j})^{\omega_i}} \\
= & \frac{e(g,g)^{(\alpha+\beta)s\mu} \cdot e(g,g)^{b^2 s\mu t}}{\prod_{i\in I}(\prod_{j=1}^{r} e(g,g)^{b^2 t\lambda_i\mu_j})^{\omega_i}} \\
= & \frac{e(g,g)^{(\alpha+\beta)s\mu} \cdot e(g,g)^{b^2 s\mu t}}{\prod_{i\in I}(e(g,g)^{(\sum_{j=1}^{r}\mu_j) b^2 t\lambda_i})^{\omega_i}} \\
= & \frac{e(g,g)^{(\alpha+\beta)s\mu} \cdot e(g,g)^{b^2 s\mu t}}{e(g,g)^{b^2 t\mu\sum_{i\in I}\lambda_i\omega_i}} \\
= & \ e(g,g)^{(\alpha+\beta)s\mu}, where \Delta_j = 1/(ID_u - ID_{uj} + ID_d - ID_{dj})
\end{aligned}
$$

With the value of $e(g,g)^{(\alpha+\beta)s\mu}$ and the ciphertext C, the message $\mathcal{M}$ can be recovered successfully.

e. Delegate($\mathbf{SK}, \tilde{\mathbf{S}}$)

Due to the fact that the key structure of both SID2LR-CP-ABE and MID-2LR-CP-ABE are highly resemble to each other, the delegation algorithms are similarly designed. To create a set of authentic attribute keys for a user A from a delegator's own attribute key, it requires the delegator to acquire some key elements from the

TTP. Here, suppose $\tilde{S} \subset S$ is a subset of the attributes possessed by a delegator ID_d, which is also the set of attributes for user A. The ID_u of the delegator in K_x of KeyGen() function is in fact the ID'_d in $\tilde{K_X}$. The goal is to create the keys for ID_u, i.e. user A, based on the key materials for the delegator. The generated attribute keys are:

$$\tilde{K} = K \cdot g^{b^2 t'} = g^{(\alpha+\beta)} g^{b^2(t+t')},$$

$$\{\tilde{K_x} = (g^{b \cdot ID'_u} h_x)^{(t+t')}\}_{\forall x \in \tilde{S}},$$

$$\{\tilde{K'_x} = (g^{b \cdot ID'_d} h_x)^{(t+t')} = K_x \cdot g^{b \cdot ID'_d t'} \cdot h_x^{t'} = K_x \cdot g^{b \cdot ID_u t'} \cdot h_x^{t'}\}_{\forall x \in \tilde{S}},$$

$$L = g^{-t}.$$

Similar as in SID2LR-CP-ABE, the required information from the new user A is his identity ID'_u. The delegator chooses a random value t' for the user and calculates $\tilde{K}$ as product of the delegator's own K and $g^{b^2 t'}$. Here, g^{b^2} is a public parameter published by the TTP through Setup() function. Similar to the preliminary scheme, h_x is a required component to convert a normal user to a delegation agent. With t' and h_x, a delegator is able to calculate $\tilde{K'_x}$. Here, ID_u is the delegator's identity instead of the user A's. $\tilde{K_x}$ can be viewed as the product of $(g^{b \cdot ID'_u} h_x)^t$ and $(g^{b \cdot ID'_u} h_x)^{t'}$. It's not difficult for the delegator to calculate the value of $(g^{b \cdot ID'_u} h_x)^{t'}$. However, for $(g^{b \cdot ID'_u} h_x)^t$, the TTP has two options to enable the delegation capability. The first option is that the delegator sends the user's identity ID'_u to the TTP and the TTP calculates $(g^{b \cdot ID'_u} h_x)^t$ for the delegator. The second approach is that the TTP assigns the value pair (g^{bt}, h_x^t) to the delegator to allow the delegation capability. As discussed

before, the first option allows the TTP to control not only which attribute an agent is allowed to delegate, but also which user is allowed to take the attribute from the delegator. The cost for such advanced capability is the computation cost happened at the TTP. The second approach only allows the TTP to control which attribute an agent is allowed to delegate. In either case, without the information from the TTP, the delegation capability will be blocked. In other words, without such information, an ordinary user is not able to generate legit attribute key components for any other user in the system.

4.5 Analysis on the Proposed Scheme

The schemes proposed in the previous section enable restricted delegation capability for ABE schemes. In this section, how these schemes can be used in different application scenarios are discussed. Moreover, the performance overhead introduced by these schemes, compared to previous solutions, is also analyzed in this section.

4.5.1 Application Scenarios

For the usage of the preliminary delegation scheme, a typical application scenario is illustrated in Figure 4.3. The TTP defines three attributes globally in this example. Following the restricted delegation feature, the TTP selects users ID1 and ID2 as delegation agents. These two agents are able to help reduce the burden of key generation for the TTP. To some extent, they can help distribute and balance the workload of the TTP. As illustrated in Figure 4.3, ID1 generates the attribute keys for users ID3 and ID4 in green color while ID2 generates the keys for ID5 and ID6 in red color.

The illustrated example is simple in that the number of attributes defined and the amount of users involved are small. When extended to real world scenarios with larger

amount of users and attributes, the effectiveness of such load-balanced delegation is much more significant.

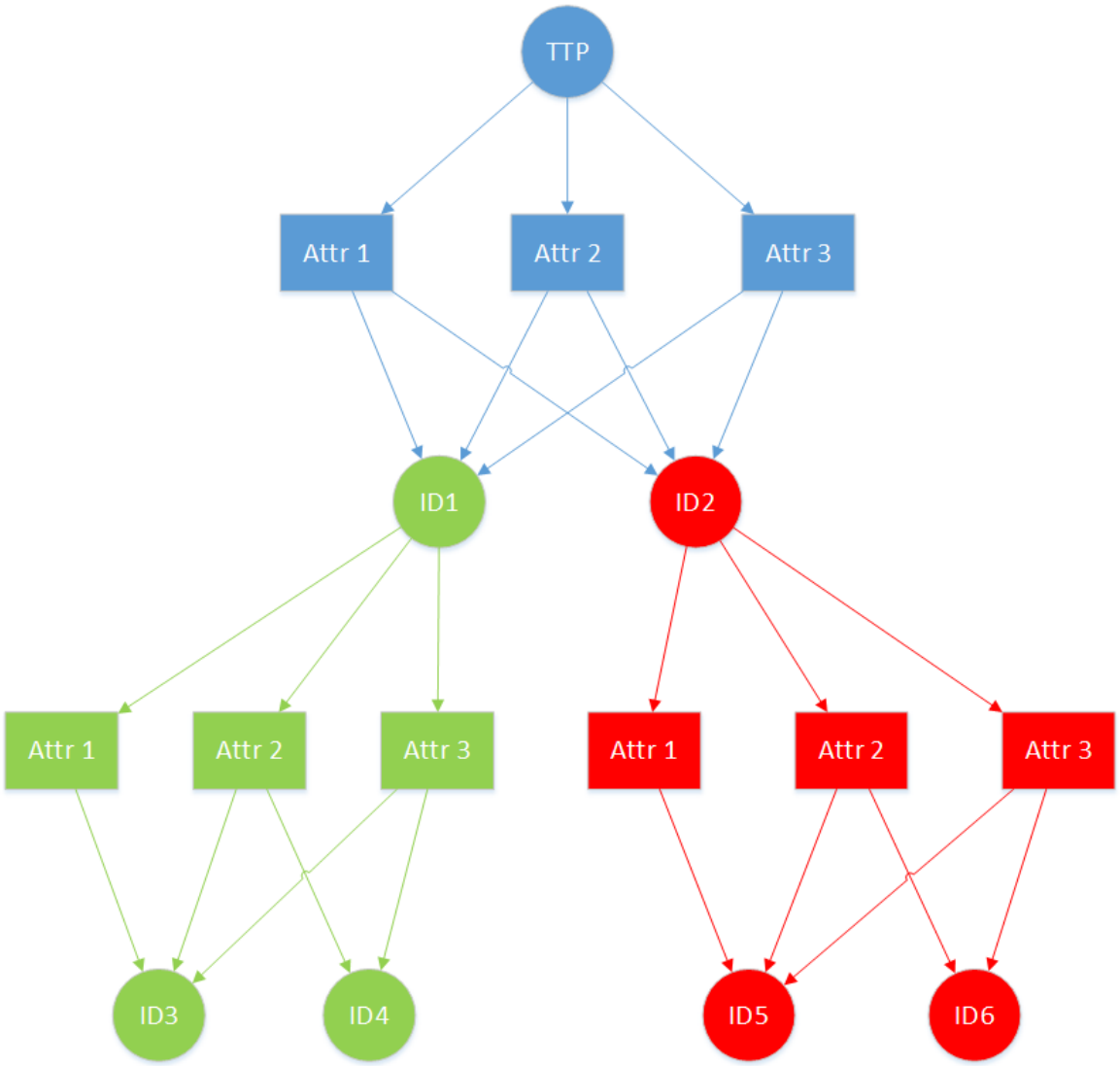

Figure 4.3: Preliminary Delegation Solution Application Scenario

The designs of SID2LR-CP-ABE and MID2LR-CP-ABE schemes are for the same goals in terms of ID revocation functionality. Therefore, their application scenarios are discussed together in this section. As illustrated before, using attributes only to describe a certain set of users has its application limitations. Depending on how each attribute is defined and assigned, cases are different. Using IDs together with

attributes allows a user to represent a group of users with more concise policy constructions. Moreover, under certain circumstances, it also makes it possible to identify certain sets of user combinations.

From delegation perspective, the goal is different. As mentioned before, instead of allowing any user to generate unlimited key components, only designated users are allowed to generate authentic new key components in the proposed scheme. Moreover, this scheme overcomes the identity issue with the preliminary scheme discussed above. In other words, the message encryption party is able to identify a specific (delegation ID, user ID) pair to be revoked from decryption. In the preliminary scheme, as there is only one ID embedded in the key components, a key generating party can either assign a delegation ID or a user ID to the new keys, not both of them at the same time. Therefore, the proposed scheme is more robust and more applicable for real-world applications.

As shown in Figure 4.4, the TTP assigns three attributes to ID1 and ID2 respectively. With the assigned attribute key components, neither ID1 nor ID2 is able to generate new key components by themselves or by collusion with each other. To further generate keys for ID3 and ID4, ID1 requests for permission from the TTP. From the angle of the TTP, it is able to disable a certain delegation capability from three levels:

- Delegation Agent: As shown in Figure 4.4, ID2 is revoked from further delegating his attributes to other users in the system. In other words, with the proposed scheme, the TTP is able to deprive any user of his delegation capability;

- Delegation Attribute: This corresponds to the case of Attr1 under ID1 in Figure 4.4. The TTP is able to identity which specific attribute that a user is requesting for delegation purpose based on the proposed schemes;

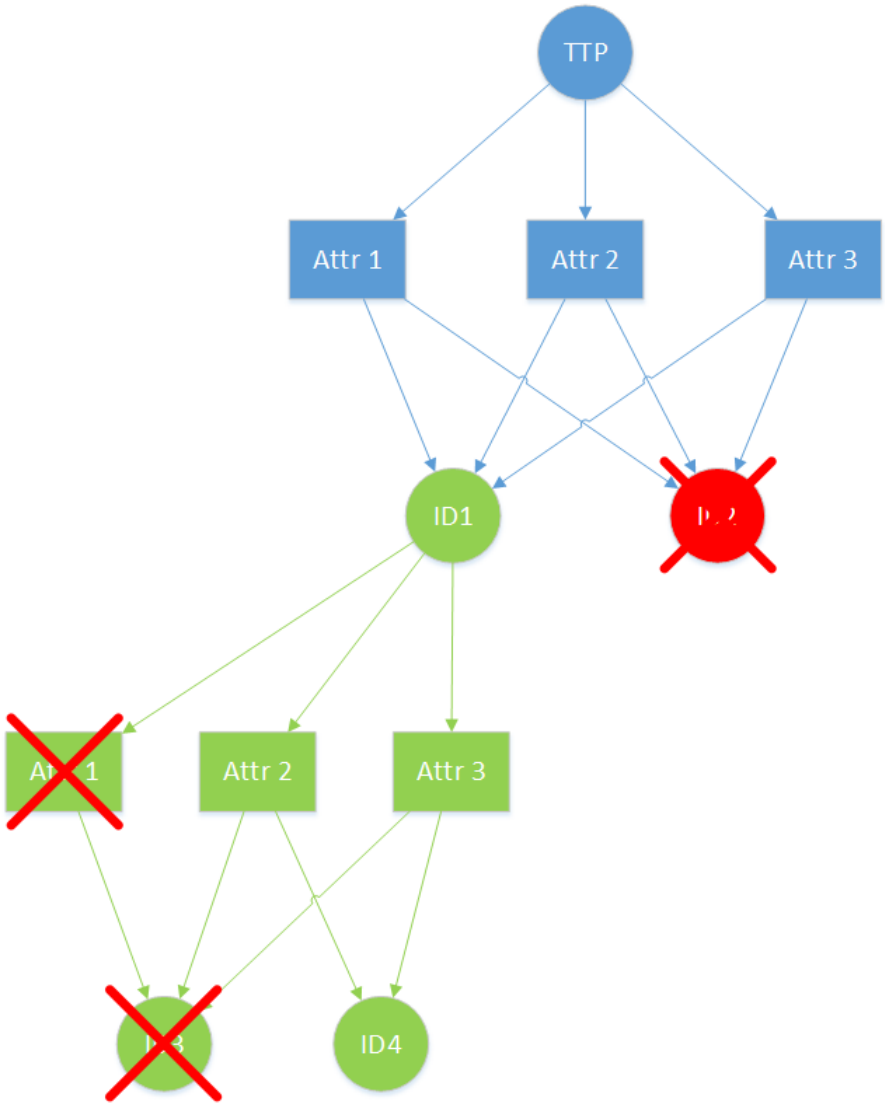

Figure 4.4: Proposed Delegation Solution Application Scenario

- Delegation Receiver: Using the proposed scheme, the TTP is also able to trace and block a certain user from getting an attribute key assigned by a delegation agent. This corresponds to the case of ruling out ID3 from the key delegation scenario in Figure 4.4.

Delegation Agent level corresponds to the need for an on-off switch in the delegation hierarchy in order to enforce restricted delegation capability as discussed before. It is the basis for other delegation restriction forms.

Delegation attribute is a new level that allows the TTP to define if a certain attribute is suitable for delegation purpose. For different attributes, their corresponding meanings and sensitivities in real world vary a lot. For sensitive attributes, it is often preferred to be managed and assigned by the TTP only, in order to maintain a high level of security. The issue with delegation is that it involves more participants into the key management and assignment process, which incurs more risk to security breaches. The TTP is assumed to have the highest security level in most scenarios, while a delegation agent is not guaranteed for the same capability. Delegation attribute allows the TTP to keep the control of certain attributes to itself.

Delegation Receiver allows the TTP to rule out certain users from getting attribute keys from a delegation agent. Similar to the discussion in the previous paragraph, some users may not be able to directly acquire certain attribute keys from the TTP due to their lack of trust by the TTP. Therefore, a feasible approach for them is to get a key component from a certain delegation agent as long as he can obtain trust from that agent. From the TTP's perspective, such approach is harmful to the integrity of the entire cryptosystem. The Delegation Receiver feature allows the TTP to block such approach so that certain users are virtually excluded from the system.

4.5.2 Delegation Agent Federation

One issue that comes with the proposed schemes is that if a user gets attribute keys from two or more delegation agents, these keys cannot be used for decryption at the same time. This is because the Delegation ID embedded in these attribute keys are different from agent to agent. Another reason is that the random value t's selected by these agents are not guaranteed to be the same by default. Therefore, under traditional system assumptions, these keys from different agents are not compatible with each other.

However, if the assumption is changed to that the Delegation Agents are selected by the TTP carefully such that they can be trusted at the same level as the TTP, then such issue can be solved. As mentioned in the Delegate function sections of each scheme, the crucial parts that define a delegation key are ID_d and t. When all the Delegation Agents are trusted, there is no need to revoke a specific agent. Therefore, the value of ID_d can be set to a common value IDD. When an agent creates keys for a user, this value IDD is used as the delegation ID in the user's key. In this way, all the delegation ID values in the delegation keys are set to the same value. Therefore, the Delegation ID difference issue is solved.

The other issue that needs to be addressed is the random value t. One approach for such issue is secure multi-party computation. The goal is to compute a random value t by all the Delegation Agents. This approach is secure, but it requires a large amount of computation and communication cost among all the Delegation Agents. When applied in mobile environment, such requirements are not easy to be satisfied. Another approach is to make the user as the media for communications among all the agents. When the first agent gets a request for attributes from a user, it randomly chooses the value for t and use other agents' public keys to encrypt the random value. The results are sent back to the user so that next time when it requests attributes from another agent, it needs to present the ciphertext to the agent so that the value t can be acquired by the agent.

Following the above discussion, the proposed schemes can be extended to support federation among Delegation Agents. It is worthwhile to point out that the discussion presented above is a special application case of the proposed schemes.

4.5.3 Performance Complexity Analysis

From performance perspective, it is worthwhile to analyze the cost for the additional delegation features introduced in the proposed schemes. In this section, such cost is further divided into computation cost and storage cost. A comprehensive analysis on the proposed schemes is provided in terms of complexity with regard to the notations listed in Table 4.1. Among all the schemes, there are five functions: Setup(), KeyGen(), Encrypt(), Decrypt(), and Delegate(), respectively. The analysis is carried out corresponding to the same function in each scheme in terms of computation cost and storage cost.

Computation Complexity Analysis

When analyzing the computation complexity, it is necessary to find out the most time-consuming operations involved in all the schemes. In fact, there are mainly four types of operations that are time-consuming: Pairing, Exponentiation, Multiplication, and Inversion. According to Li *et al.* (2014), the most computation-intensive operations are Pairing and Exponentiation. Thus, in this section, the amount of pairing operations and that of exponentiation operations needed for each function are taken as metrics for computation complexity. The complexity of all the schemes in the two metrics are presented in Table 4.2 and Table 4.3 respectively. The schemes OIDR-CP-ABE and MIDR-CP-ABE are the base schemes for SID2LR-CP-ABE and MID2LR-CP-ABE, respectively. The analysis is presented as follows:

In the original CP-ABE scheme, the amount of pairing operations needed for Setup() function is 1. Pairing is only incurred in calculating for the value of $e(g, g)^{\alpha}$. For the other four schemes, the number of pairing needed in Setup() is also 1 for the same reason that only $e(g, g)^{\alpha}$ in the global key materials require a pairing operation.

Table 4.1: Notations.

Sym	Meaning
p	the prime order of the multiplicative cyclic group G
g	the generator of the multiplicative cyclic group G.
m	the number of attributes defined in the system.
$\mathcal{U}$	the attribute set defined in the system, $\|\mathcal{U}\| = m$.
n	the number of identities in the system.
$\mathcal{I}$	the identity set defined in the system, $\|\mathcal{I}\| = n$.
α	a random element chosen by TTP from G.
b	a random element chosen by TTP from G.
$\mathcal{M}$	a message to be encrypted by the system.
M	a $l \times n$ matrix
ρ	a function that associates rows of M to attributes.
A_x	the x-th row of A.
S	the set of attributes created for a specific user.
$\tilde{S}$	the set of attributes used for delegation purpose.
l	the number of attributes involved in the encryption process.
r	the number of identities involved in the encryption process .

Due to such fact, it is more important to focus on the amount of exponentiation operations involved in the five schemes. In CP-ABE, the amount of exponentiation needed in Setup() function is 3. However, in both OIDR-CP-ABE and MIDR-CP-ABE, the amount of such operations required is $m + 3$, where m is the amount of attributes defined globally as illustrated in Table 4.1. Such difference comes from the fact that each attribute is defined as a value h_x in CP-ABE, while it is defined as ${h_x}^b$ in both OIDR-CP-ABE and MIDR-CP-ABE. For SID2LR-CP-ABE and

Table 4.2: Computation Complexity Comparison in Pairing Operations

Function	CP-ABE	OIDR-CP-ABE	MIDR-CP-ABE	SID2LR-CP-ABE	MID2LR-CP-ABE
Setup()	1	1	1	1	1
KeyGen()	0	0	0	0	0
Encrypt()	0	0	0	0	0
Decrypt()	$2\|I\|+1$	$2\|I\|+1$	$2\|I\|r+1$	$4\|I\|+1$	$4\|I\|r+1$
Delegate()	0	0	0	0	0

Table 4.3: Computation Complexity Comparison in Exponentiation Operations

Function	CP-ABE	OIDR-CP-ABE	MIDR-CP-ABE	SID2LR-CP-ABE	MID2LR-CP-ABE
Setup()	3	$m+3$	$m+3$	$m+4$	$m+4$
KeyGen()	$\|S\|+2$	$\|S\|+3$	$\|S\|+3$	$2\|S\|+4$	$2\|S\|+4$
Encrypt()	$3l+2$	$3l+2$	$3lr+2$	$5l+2$	$5lr+2$
Decrypt()	$\|I\|+1$	$\|I\|+1$	$\|I\|r+1$	$\|I\|+1$	$\|I\|r+1$
Delegate()	$2\|\tilde{S}\|+1$	$2\|\tilde{S}\|+1$	$2\|\tilde{S}\|+1$	$4\|\tilde{S}\|+1$	$4\|\tilde{S}\|+1$

MID2LR-CP-ABE, the additional computation cost is one more exponentiation operation, compared to OIDR-CP-ABE and MIDR-CP-ABE respectively. Such cost difference is relatively small in real-world application scenarios.

In all the five schemes, there is no need for any pairing operation in preparing for the keys of any user. Comparatively, exponentiation operation is the key contributer to the time cost in this function for all the three schemes. In CP-ABE, the number of exponentiations needed is $|S|+2$. In both OIDR-CP-ABE and MIDR-CP-ABE, such number is increased to $|S|+3$, which is a small increase. For SID2LR-CP-ABE and MID2LR-CP-ABE, such number is doubled due to the additional component for each attribute. It is clear from the table that the time consumption is linear to the

number of attributes assigned for a user if the major time-consuming operation is exponentiation.

When it comes to the Encrypt() function, the computation cost in terms of pairing operations is zero for all the five schemes. The amount of exponentiation operations is more useful in differentiating the computation cost among these schemes. In both CP-ABE and OIDR-CP-ABE, it takes $3l + 2$ exponentiations for encryption process. Here l is the number of attributes involved in the encryption process. It can be seen that SID2LR-CP-ABE and MID2LR-CP-ABE need about 2/3 more computation cost than OIDR-CP-ABE and MIDR-CP-ABE, respectively, in terms of exponentiation operations. The increased cost is still of the same order of significance as OIDR-CP-ABE and MIDR-CP-ABE. Such increase mainly comes from the more time needed in encryption with the additional delegation keys.

The Decrypt() function is the most time-consuming one among the five functions as it incurs more pairing operations than the other three functions. In CP-ABE, the number of pairings needed is $2|I| + 1$, where I is the set of attributes involved in the decryption process. It requires the same amount of pairings in OIDR-CP-ABE. SID2LR-CP-ABE and MID2LR-CP-ABE almost double the amount of pairings as in OIDR-CP-ABE and MIDR-CP-ABE respectively. In the decryption process, the contribution from exponentiation is much less because the amount of pairings is significant in Decrypt() function. The numbers of exponentiations in these five schemes are $|I| + 1$, $|I| + 1$, $|I|r + 1$, $|I| + 1$, and $|I|r + 1$, respectively.

For Delegate() function, the costs for pairing operations are zero for all the five schemes. In terms of exponentiation operations, the cost for original CP-ABE, OIDR-CP-ABE, and MIDR-CP-ABE are all $2|\tilde{S}| + 1$. For SID2LR-CP-ABE and MID2LR-CP-ABE, such numbers are almost doubled to $4|\tilde{S}| + 1$ in both cases. The major increase comes from the fact that the key components for each attribute are doubled

in SID2LR-CP-ABE and MID2LR-CP-ABE schemes due to the additional cost for delegation ID related computations. The same reason also contributes to the changes in KeyGen() function.

Storage Complexity Analysis

Table 4.4: Storage Cost Comparison

Function	CP-ABE	OIDR-CP-ABE	MIDR-CP-ABE	SID2LR-CP-ABE	MID2LR-CP-ABE
Setup()	$m+5$	$m+6$	$m+6$	$m+8$	$m+8$
KeyGen()	$\|S\|+2$	$\|S\|+2$	$\|S\|+2$	$2\|S\|+2$	$2\|S\|+2$
Encrypt()	$2l+2$	$2l+2$	$2lr+2$	$3l+2$	$3lr+2$
Delegate()	$\|S'\|+2$	$\|S'\|+2$	$\|S'\|+2$	$2\|S'\|+2$	$2\|S'\|+2$

When it comes to storage cost, the case is different in that there is no additional storage cost for Decrypt() function as its results are directly used as the final plaintext. For storage cost, only the storage space for final results of each function is considered. In other words, the storage for temporary variables that are normally used in computer memories are not considered. Each element used in the functions is a member from one of the four groups: G_1, G_2, G_T, Z_r, which is stored as an *element_t* data structure in PBC library Lynn (2014). Therefore, the number of such elements is used as a metric for storage cost analysis. Table 4.4 summarizes the overall cost corresponding to each function in all the five schemes.

In Setup() function, the resulting storage cost is used for storing PK and MSK. It takes $m+4$ for storing PK in the first three schemes. The same cost for the latter two schemes is $m+5$, due to the addition of $e(g,g)^{\beta}$. The storage cost for MSK is 1 in CP-ABE and 2 in both OIDR-CP-ABE and MIDR-CP-ABE. The same cost for SID2LR-CP-ABE and MID2LR-CP-ABE is 3. Overall, the additional storage

cost, compared with the former three schemes, is small and constant, which is not a significant issue in applications.

In KeyGen(), the cost for the first three schemes is $|S| + 2$. However, the same cost in SID2LR-CP-ABE and MID2LR-CP-ABE is almost twice as that in the first three. This comes from the need for two components corresponding to each attribute value: one for delegation ID and the other for user ID. As mentioned before, this is the approach how the proposed two schemes achieve their designed functionalities. Thus, such additional cost is necessary.

The storage cost for Encrypt() function, which equals to the size of the ciphertext generated in this phase, is $2l + 2$ in both CP-ABE and OIDR-CP-ABE. The size of ciphertext in MIDR-CP-ABE is larger than the former two schemes, which is $2lr + 2$. For SID2LR-CP-ABE and MID2LR-CP-ABE, the same cost is almost 1.5 times of that for OIDR-CP-ABE and MIDR-CP-ABE, respectively. Such difference also comes from the fact that an additional pair of key components need to be generated for each delegation ID.

Based on the analysis presented above, it can be seen that the overall costs in both computation and storage is no more than twice as those in the corresponding baseline schemes. Between the two proposed schemes, the SID2LR-CP-ABE scheme performs better than MID2LR-CP-ABE in both computation cost and storage cost. This is straightforward as SID2LR-CP-ABE is functionally less powerful than MID2LR-CP-ABE. The overall costs for both schemes are practical for real-world applications.

4.5.4 Real-world Implementation

To further evaluate the real-world time consumption, the proposed schemes are implemented in C language using PBC library Lynn (2014) on Ubuntu 14.04 64bit operating system. The hardware configuration for the machine that runs the exper-

iment is: Intel i7 Quad-core CPU at 2.60GHz; 8GB memory. To test the relations between the amount of attributes involved and the time consumption, the number of IDs revoked is fixed to 1 and the number of attributes that are involved in each of the four functions, *Setup()*, *KeyGen()*, *Encrypt()*, and *Decrypt()*, is gradually increased. In other words, the first test is run for the SID2LR-CP-ABE scheme. The time consumption of these functions are tested separately.

For each attribute setting, the experiments mentioned above are run for ten times and the average values, in milliseconds, are taken as the results, which are presented in Figure 4.5.

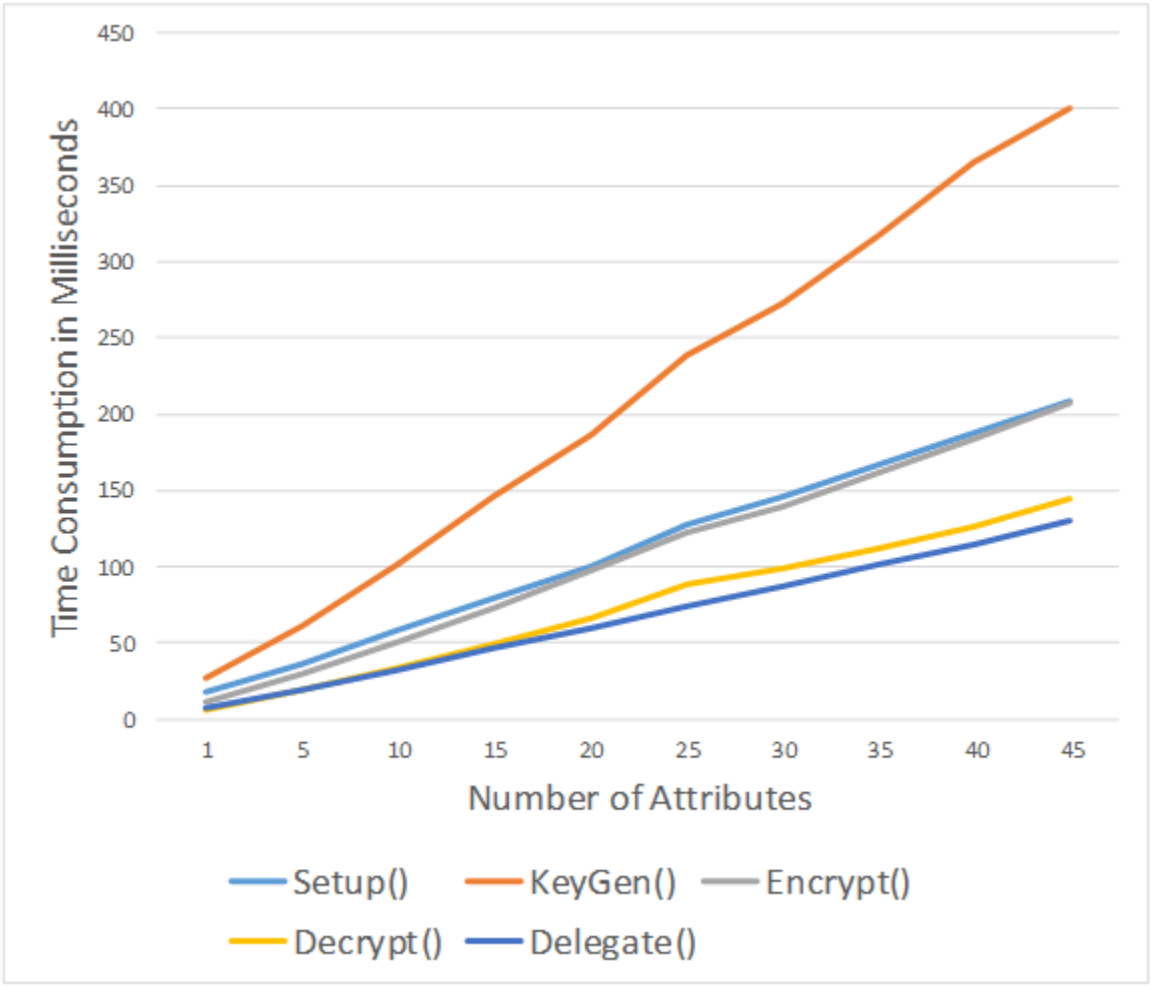

Figure 4.5: Relations between the Amount of Attributes and Time Consumption

As can be seen in the figure, the time consumption for all the four functions are generally linear to the number of involved attributes. The cost for KeyGen() is much larger than that for the other three functions. This is because the KeyGen() function requires a process for random element generation and element setting in implementation. Such process takes much more time than pairing or exponentiation operations. It happens once for each attribute. Therefore, this function takes the largest time cost. As in real applications, KeyGen() is run once for each user. It is relatively less frequent than Encrypt() and Decrypt(). Thus, such cost is acceptable. For Setup() function, it is only executed once by the TTP when the cryptosystem is established. The cost for Encrypt() and Decrypt() functions are more important to applications as they are the most frequently used functions. With 45 attributes involved, the cost for Encrypt() is right over 200 milliseconds. The cost for Decrypt() under the same setting is less than 150 milliseconds.

To further explore the influence on the time consumption from the number of IDs revoked, a second experiment is conducted with a fixed number of attributes and changing numbers of revoked IDs. This experiment is an implementation of MID2LR-CP-ABE scheme. In this experiment, the number of attributes is set to 20 and the number of revoked IDs is gradually increased from 1 to 10. The evaluation result is shown in Figure 4.6. It can be seen that the time consumption of Setup() and KeyGen() are not sensitive to the number of IDs revoked. This is because both functions do not have the revoked ID list involved in their operations. Both Encrypt() and Decrypt() follow a linear trend in Figure 4.6. The Encrypt() function is the most sensitive one to the number of revoked IDs. When the number of revoked IDs is greater than 6, with 20 attributes involved, the overall time cost for *Encrypt()* increases to more than 1 second. The Decrypt() function is relatively less sensitive to the increase of revoked ID amount.

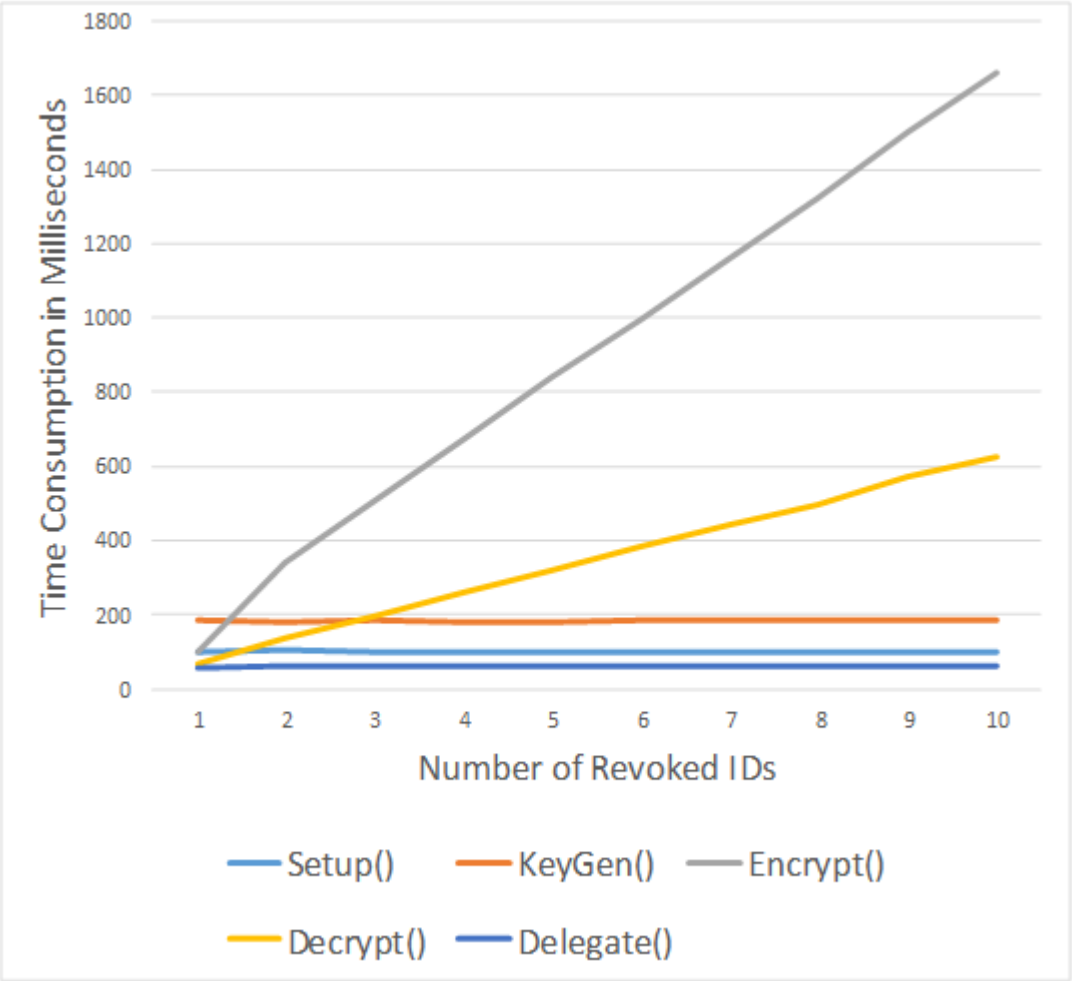

Figure 4.6: Relations between the Amount of Revoked IDs and Time Consumption.

When comparing Figure 4.5 with Figure 4.6, it can be seen that the time cost is more sensitive to the number of revoked IDs than that of attributes as the curves for Encrypt() and Decrypt() in Figure 4.6 increases much faster than those in Figure 4.5. Therefore, from performance perspective, it is recommended that a user should increase the number of revoked IDs only when it's necessary.

4.6 Conclusion

In this chapter, the idea of restricting the delegation capability in CP-ABE schemes is proposed and discussed with details. Different feasible directions for realizing such

capability are analyzed. Based on such analysis, two schemes that correspond to the two ID-revocable baseline schemes are proposed. Such schemes allow users to enforce restricted delegation capabilities in three different levels: Delegation Agent, Delegation Attribute, and Delegation Receiver. Performance analysis on these schemes are proposed in terms of their computation cost and storage cost. Here, the storage cost is also referred to as communication cost in networking scenarios. Based on the performance analysis, the proposed schemes provide restricted delegation features with practical cost for computation and storage.

Chapter 5

FUTURE RESEARCH DIRECTIONS

In this chapter, future research directions based on the work presented in previous chapters are discussed. The discussion is organized as two parts: one on anonymous communications in mobile environment and the other on further development for ABE schemes.

5.1 Anonymous Communications in Mobile Environment

The proposed scheme in Chapter 2 aims to solve the anonymity issue in MANETs by relaxing the strict system model assumptions so that it is more realistic for real-world application scenarios. The scheme is able to handle the influence from localization errors that are inherent with all localization techniques. It also helps reduce the scale of the entire network. Other research directions that meet the real-world requirements are worthwhile for further investigation. For instance, the relationship between the amount of information leak and the ratio of monitored network. In the proposed scheme in Chapter 2, the monitoring system is assumed to be able to monitor the entire traffic in the target network. It is not clear that if the capability of the monitoring system is reduced and only covers a small portion of the target network, how the analysis result accuracy is deteriorated.

Anonymity issues with mobile networks have been an important topic for quite a while. A main threat to anonymity in traditional MANETs is the link-based communication patterns. For link-less approaches, ICN networking as a example, it is worthwhile to further investigate on improving performance under certain security guarantees. The proposed scheme in Chapter 3 works fine for enforcing access con-

trol on contents. However, using ABE related scheme has a drawback that the time consumption for encryption and decryption increases high when attribute policies are large and complex. It is necessary to further explore more efficient and powerful schemes for such purpose, whether it be ABE-based or not.

5.2 Further Development for ABE Schemes

Many of existing works on ABE schemes are focused on extending functional capabilities. Such features include hidden policy, constant-length ciphertext, multi-authority, comparable attribute values, ID revocation, and attribute revocation.

Hidden policy refers to the case that the attribute policy is protected in certain ways so that anyone who obtains a copy of the ciphertext cannot directly read the content of the policy. The constant-length ciphertext feature aims to break the relation between the size of the attribute policy and that of the ciphertext as in the original CP-ABE scheme. One direct benefit from this is that an attacker is not able to correlate the size of the ciphertext to any information regarding the ciphertext when all of them are of the same size. Multi-authority is a feature that supports more than one single TTP in the entire cryptosystem. The feature of comparable attribute values extends the meaning of attribute definition. In original ABE scheme, each attribute value is defined as a nominal value. The comparable attribute feature allows comparison between attribute values. Thresholds and intervals can be enforced with such feature. ID-revocation allows a user to combine user IDs with attributes when identifying a group of users through a policy. Under certain circumstances, it allows to identify a group of users that cannot be directly specified through attribute combinations only. It also helps reduce the size of policies, therefore also reduces the computation and storage cost, in certain scenarios. The need for attribute revocation comes from the observation that it is impossible in some ABE schemes or would

be very complex in other schemes to enforce a policy which specifies that a certain attribute owners should definitely be excluded from decryption. Such issue lies in the fact that one such attribute owner can simply avoid using that specific attribute in the decryption process in the original ABE scheme and he will still succeed. The original ABE scheme does not have such a mechanism to force a decryptor get all his attributes involved in the decryption process. Therefore, such unwanted attributes can be "hidden" by the decryptor without raising any error or alerts.

All the above mentioned features are very useful and desirable in real application scenarios. They are of great value for future research work on ABE schemes. However, each additional feature comes with a certain cost. The most straightforward cost is in terms of additional computation powers needed and extra storage space/communication cost incurred. Another direct and difficult challenge to further extensions is the feasibility issue. In other words, the more features get incorporated, the less likely a feasible solution can be designed because each feature requires a specially designed scheme. Different special schemes are not compatible in most cases. Even if it is feasible, the more features get involved, the more complicated the resulting scheme will be. This directly incurs more additional performance cost. It is always a great challenge to balance between functionality, cost, and security in designing new ABE schemes. Security evaluation is another challenging direction that comes with such extension works. Under different security assumptions, the corresponding evaluation models and approaches are different. In many cases, such works are unprecedented.

5.3 Restrict Delegation Capability in ABE Schemes

In Chapter 4, a pioneer work on restricting the delegation function in ABE schemes is discussed, with a specific scheme designed for ID-revocable solutions as a concrete

research outcome. Restricted delegation capability is worthwhile to be further explored in a broad scope for all the ABE schemes. Specifically, different levels of such work, including those discussed in Section 4.3, need further investigation.

The majority of Chapter 4 solved the issue with breadth-first restriction. From utility perspective, it is very useful to allow each delegation agent in a delegation tree to specify his own delegation restriction rules. In a real world application scenario, U.S. Department of Education (DoED) may play the role as the TTP. It defines global parameters and creates attributes for all the education participants. Corresponding attributes are assigned to education organizations to simplify the management overhead on TTP. However, for different such organizations, DoED may have different restrictions on how an organization can further delegate different attributes. For instance, Arizona State University (ASU) may have a less rigid restriction than Mesa Community College (MCC) due to its much bigger size. ASU also needs to further allow delegations on some attributes at each of its schools due to its large populations. ASU itself as a university may have different rules for different schools. Each department under the same school also needs to further extend the delegation relation on some attributes to further specify their management policies.

To realize more powerful delegation restriction capabilities, it is necessary to start at the simple scenarios. The breadth-first solution serves as such a lead to future solutions. Following the discussion in Section 4.3, the next step is to realize depth-first restriction. This helps extend the control of a certain node in the delegation tree to more than his direct children nodes. As in the above example, with such capability, the DoED is able to control the delegation capability of School of Computing, Informatics, and Decision Systems Engineering (CIDSE) at ASU, or even the Computer Science department at ASU. Such capability can be realized by embedding secret into delegation keys that are passed along multiple generations within a delegation tree.

Both the breadth-first and the depth-first approach solve the delegation restriction issue in an all-or-nothing way. A parent node in the delegation tree is only able to enable or disable the delegation capability of a certain child node. It is not able to provide fine-grained controls. For instance, a parent node cannot specify the amount of children a certain node can create in the delegation tree. As shown in Figure 5.1, the TTP cannot control the value of n, i.e. the amount of users that A can generate delegation attribute keys for, with either breadth-first or depth-first approach.

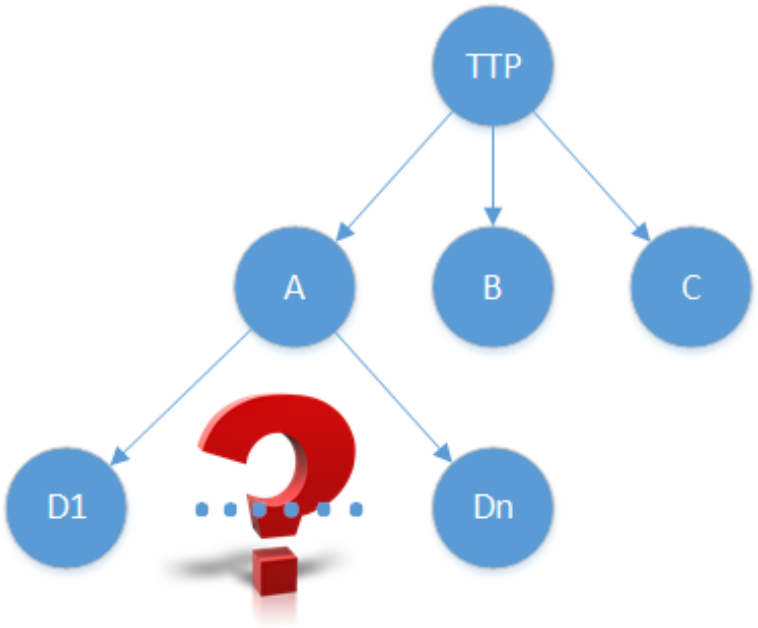

Figure 5.1: Fine-Grained Restriction.

Such fine-grained solution is very useful in practice. As in the above example, the DoED is able to put a limit on the number of schools ASU can create under its administration. For some sensitive attributes, it is desirable for the TTP to be able to track the number of assignees through the entire system, including those who obtain the attributes via delegation.

Based on the work presented in Chapter 4, the depth-first restriction and the fine-grained solution are two of the most important and interesting topics to work on in attribute delegation. Other aspects that need further investigation include perfor-

mance optimizations and security evaluations. As shown in Chapter 4, the delegation restriction schemes proposed require additional computation cost and storage cost. For all the ABE schemes, performance concerns are always a significant factor that determines the wide application of such schemes. For security evaluations, delegation function is entirely a different scope from the evaluation of a typical ABE scheme. It faces different types of attacks, security assumptions, and security models. Due to the novelty of restricted delegations, no previous work has been found on the security evaluation of such topic. Thus, future work in this direction is of great importance.

Chapter 6

SUMMARY

In Chapter 2, 3, and 4, one specific topic is discussed with proposed solutions for the problems targeted respectively. All the work is summarized in this chapter following the same sequence as they are presented.

The first topic of this dissertation is on the anonymity issue with MANETs. The proposed work analyzes the issue from an attacking party's perspective. If a powerful monitoring system is deployed in the same area as the targeted MANET to passively collect traffic of the MANET, what approaches can the attacker employ and how much useful information can he get from such system. A clear view on such problem provides insights on how much useful information is leaked in the wireless channel from the MANET. More importantly, it provides guidance for proposing counter-attack strategies on such issue.

Following this idea, an obstacle to applying the monitoring system in real world scenarios is the localization errors that come with every system. This is the focus of Chapter 1. The proposed scheme for super nodes is able to handle the influence from localization errors. With such approach, the above-mentioned monitoring system for anonymity analysis can be carried out in real life. This is the major goal of this part of work.

Following the work on anonymity in Chapter 2, one conclusion that is made from it is that the link-based communication pattern is a major contributor to the anonymity issue. Therefore, a direct alternative is to use link-less communications. One such solution is Information Centric Networking (ICN). ICN breaks the direct connections between a message sender and corresponding receivers. Instead, messages are indi-

rectly transmitted from one side to the other in the form of contents. In this way, the two parties are decoupled in their communication behaviors.

However, the indirect communication pattern also introduces challenges in maintaining a global content management mechanism, specifically in access control. For this purpose, a new naming scheme in ICN based on Attribute-Based Encryption (ABE) algorithms is proposed in Chapter 3. This scheme embeds the message encrypting key in content names, which are used for routing purpose in ICN. In this way, a potential receiver is able to determine his eligibility in decrypting the content through the content name at hand. Only permitted users are able to successfully decrypt the ciphertext.

The proposed ABE-based scheme also achieves features such as multi-authority. Such capability allows easy management on attributes among a number of trusted attribute authorities. For attribute management, attribute delegation is another approach. It helps reduce the workload on the centralized TTP. It allows a normal user to create and assign attribute keys to other users. However, it also introduces security concerns. Unrestrained delegation capability is harmful to the entire cryptosystem.

To properly handle such an issue, restricted delegation functions are proposed in Chapter 4. It is designed for an ID-revocable ABE scheme, but can also be extended to other ABE schemes. With such capability, a TTP is able to determine which attributes a specific user can further delegate. This serves as the first step of the road-map to delegation restriction schemes, which is discussed in Chapter 4.

The overall arrangement of this dissertation follows the thread presented above. For each of the three aspects, there is a performance evaluation to assess the overall cost for the proposed solutions. Based on the discussions and the evaluation results, future paths for continuing work in these aspects are presented in Chapter 5, which is expected to serve as a reference for interested readers in these areas.

REFERENCES

"Earthquake rocks internet in nepal", URL `http://tinyurl.com/zmm4voa` (2015a).

"Nepal communications hit by power outage, last-mile issues", URL `http://tinyurl.com/hcr3pc3` (2015b).

"Omnet++ network simulation framework", URL `http://www.omnetpp.org/` (Accessed Sep. 2013).

Angius, F., M. Gerla and G. Pau, "Bloogo: Bloom filter based gossip algorithm for wireless ndn", in "Proceedings of the ACM Workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications", NoM (2012).

Arianfar, S., T. Koponen, B. Raghavan and S. Shenker, "On preserving privacy in content-oriented networks", in "Proceedings of the ACM SIGCOMM workshop on Information-centric networking", ICN (2011).

Bethencourt, J., A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption", in "Proceedings of the IEEE Symposium on Security and Privacy", SP (2007).

Biswas, R., K. Chowdhury and D. Agrawal, "Attribute allocation and retrieval scheme for large-scale sensor networks", International Journal of Wireless Information Networks (2006).

Carzaniga, A., M. Rutherford and A. Wolf, "A routing scheme for content-based networking", in "Proceedings of the IEEE International Conference on Computer Communications", INFOCOM (2004).

Cheung, L. and C. Newport, "Provably secure ciphertext policy abe", in "Proceedings of the ACM conference on Computer and Communications Security", CCS (2007).

Cisco, "Cisco visual networking index: forecast and methodology, 2014-2019", URL `http://tinyurl.com/mev32z8` (2015).

Dannewitz, C., J. Golic, B. Ohlman and B. Ahlgren, "Secure naming for a network of information", in "Proceedings of the IEEE International Conference on Computer Communications", INFOCOM (2010).

El Defrawy, K. and G. Tsudik, "Privacy-preserving location-based on-demand routing in manets", IEEE Journal on Selected Areas in Communications **29**, 10, 1926–1934 (2011).

Faria, D. B. and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints", in "Proceedings of the ACM Workshop on Wireless Security", pp. 43–52 (2006).

Fotiou, N., G. F. Marias and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures", in "Proceedings of the ICN workshop on Information-centric networking", ICN (2012a).

Fotiou, N., P. Nikander, D. Trossen and G. Polyzos, "Developing information networking further: From psirp to pursuit", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (2012b).

Huang, D., "Unlinkability measure for ieee 802.11 based manets", IEEE Transactions on Wireless Communications **7**, 3, 1025–1034 (2008).

Huang, D., Z. Zhou and Z. Yan, "Gradual identity exposure using attribute-based encryption", in "Proceedings of the IEEE International Conference on Social Computing", SocialCom (2010).

Johnson, D. B., "Efficient algorithms for shortest paths in sparse networks", Journal of the ACM **24**, 1, 1–13 (1977).

Kohno, T., A. Broido and K. C. Claffy, "Remote physical device fingerprinting", IEEE Transactions on Dependable and Secure Computing **2**, 2, 93–108 (2005).

Koponen, T., M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker and I. Stoica, "A data-oriented (and beyond) network architecture", in "Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications", (2007).

Lewko, A. and B. Waters, "Decentralizing attribute-based encryption", in "Proceedings of the Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology", (2011).

Li, B., A. P. Verleker, D. Huang, Z. Wang and Y. Zhu, "Attribute-based access control for icn naming scheme", in "Proceedings of the IEEE Conference on Communications and Network Security", CNS, pp. 391–399 (2014).

Li, B., Z. Wang and D. Huang, "An efficient and anonymous attribute-based group setup scheme", in "Proceedings of the IEEE Global Communications Conference", GLOBECOM (2013).

Lynn, B., "Type a internals", URL `http://tinyurl.com/gs9s8y9` (2006).

Lynn, B., "Pbc library the pairing-based cryptography library", in "http://crypto.stanford.edu/pbc/", (Accessed March 2014).

named data, "Named data networking", URL `http://named-data.net` (2015).

Nishide, T., K. Yoneyama and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures", in "Proceedings of the International Conference on Applied Cryptography and Network Security", (2008).

Patwari, N. and S. K. Kasera, "Robust location distinction using temporal link signatures", in "Proceedings of the Annual ACM International Conference on Mobile Computing and Networking", pp. 111–122 (2007).

Psaras, I., W. K. Chai and G. Pavlou, "Probabilistic in-network caching for information-centric networks", in "Proceedings of the ICN workshop on Information-centric networking", ICN (2012).

Qin, Y., D. Huang and B. Li, "Stars: A statistical traffic pattern discovery system for manets", IEEE Transactions on Dependable and Secure Computing (2014).

Rasmussen, K. B. and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks", in "Proceedings of IEEE EAI International Conference on Security and Privacy in Communication Networks", SecureComm (2007).

Sharma, S., "Analysis of 802.11b mac: A qos, fairness, and performance perspective", URL `http://www.ecsl.cs.sunysb.edu/tr/wlanrpe.pdf` (2003).

Shoup, V., "Lower bounds for discrete logarithms and related problems", in "Proceedings of International Conference on the Theory and Application of Cryptographic Techniques", EUROCRYPT (1997).

Singh, S., "A trust based approach for secure access control in information centric network", International Journal of Information and Network Security (2012).

Sun, Y., S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar and S. Uhlig, "Trace-driven analysis of icn caching algorithms on video-on-demand workloads", in "Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies", (2014).

Wang, Z., D. Huang, Y. Zhu, B. Li and C.-J. Chung, "Efficient attribute-based comparable data access control", IEEE Transactions on Computers (2015).

Yu, S., K. Ren and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity", in "Proceedings of the International Conference on Security and Privacy in Communication Netowrks", (2008).

Yu-Ting, Y., C. Tandiono, X. Li, Y. Lu, M. Sanadidi and M. Gerla, "Ican: Information-centric context-aware ad-hoc network", in "Proceedings of the International Conference on Computing, Networking and Communications", ICNC (2014).

Zhang, Y., W. Liu, W. Lou and Y. Fang, "Mask: anonymous on-demand routing in mobile ad hoc networks", IEEE Transactions on Wireless Communications **5**, 9, 2376–2385 (2006).

Zhu, Y., H. Hu, G.-J. Ahn, M. Yu and H. Zhao, "Comparison-based encryption for fine-grained access control in clouds", in "Proceedings of the ACM Conference on Data and Application Security and Privacy", CODASPY (2012).