Context-Aware Attribute-Based Techniques for Data Security and Access Control in Mobile Cloud Environment

A Thesis Submitted to
City University London, School of Engineering and Mathematical Sciences
In Fulfillment of the Requirements for the Degree
Doctor of Philosophy in
Information Engineering

By

FEI LI

April, 2015

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis arose after years of research that has been done since I came to City University London. First and foremost, I would like to express my gratitude to my supervisor Professor Muttukrishnan Rajarajan, for his supervision, advice and guidance from the starting of my research. With his extraordinary experience and admirable insights, he has taught me about science, life, and encouragement. It is my pleasure to work with you.

I also would like to acknowledge the research collaboration with Dr. Yogachandran Rahulamathavan. Thanks for the invaluable advice and patience. I am grateful to my colleagues and everyone for their support and help on my research work.

By the completion of this thesis, I am approaching the end of a 20 years long life for officially enrolled as a student. Looking back at these years I will remember with affection the people I have met and the moments with joy and happy along the way.

I owe my final thanks to my beloved parents, with their unconditional support, love, and trust. This thesis is dedicated to them.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning. I hereby grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

The explosive growth of mobile applications and Cloud computing has enabled smart mobile devices to host various Cloud-based services such as Google apps, Instagram, and Facebook. Recent developments in smart devices' hardware and software provide seamless interaction between the users and devices. As a result, in contrast to the traditional user, the mobile user in mobile Cloud environment generates a large volume of data which can be easily collected by mobile Cloud service providers. However, the users do not know the exact physical location of their personal data. Hence, the users cannot control over their data once it is stored in the Cloud. This thesis investigates security and privacy issues in such mobile Cloud environments and presents new user-centric access control techniques tailored for the mobile Cloud environments.

Most of the work to date has tried to address the data security issues on the Cloud server and only little attention has been given to protect the users' data privacy. One way to address the privacy issues is to deploy access control technique such as Extensible Access Control Markup Language (XACML) to control data access on users' data. XACML defines a standard of access control policies, rule obligations and conditions in data access control. XACML utilizes Extensible Markup Language (XML) schema to define attributes of data requesters, resources, and environment in order to evaluate access requests. A user-centric attribute-based access control model using XACML which enables users to define privacy access policies over the personal data based on their preferences is presented.

In order to integrate the data security and user's privacy in mobile Cloud environment, the thesis investigates attribute-based encryption (ABE) scheme. ABE scheme enables data owners to enforce access policies during the encryption. Context-related attributes such as requester's location and behavior are incorporated within ABE scheme to provide data security and user privacy. This will enable the mobile data owners to dynamically control the access to their data at runtime. In order to improve the performance, a solution that offloads the high-cost computational work and communications from the mobile device to the Cloud is proposed. Anonymisation techniques are applied in the key issuing protocol so that the users' identities are protected from being tracked by the service providers during transactions. The proposed schemes are secure from known attacks and hence suitable for mobile Cloud environment. Security of the proposed schemes is formally analyzed using standard methods.

Notations and Abbreviations

2PC	Two-Phase Commit Protocol
3G	Third Generation
4G	Forth Generation
AES	Advanced Encryption Standard
APPEL	A P3P Preference Exchange Language
CH	Context Handler
CIA	Confidentiality Integrity Availability
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DAC	Discretionary Access Control
DES	Data Encryption Scheme
DVLA	Driver and Vehicle Licensing Agency
EC2	Elastic Compute Cloud
EPAL	Enterprise Privacy Authorization Language
EPIC	Electronic Privacy Information Centre
GID	Global Identity
GPS	Global Positioning System
HIPAA	Health Insurance Probability and Accountability Act
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identity
IdP	Identity Provider
ID-FF	Liberty Identity Federation Framework

ID-WSF	Liberty Identity Web Services Framework
ID-SIS	Liberty Identity Services Interfaces Specification
IMEI	International Mobile Station Equipment Identity
J2EE	Jave 2 Platform Enterprise Edition
JAX-WS	Java API for Web Services
JSON	JavaScript Object Notation
JSP	JaveServer Pages
KP-ABE	Key-Policy Attribute-Based Encryption
LTE	Long-Term Evolution
MA-ABE	Multi-Authority Attribute-Based Encryption
MAC	Media Access Control
ManAC	Mandatory Access Control
NHS	National Health Service
NI	National Insurance
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standard
OECD	Organization for Economic Co-operation and Development
OMB	U.S. Office of Management and Budget
P3P	Platform for Privacy Preferences
PAP	Policy Administration Point
PDP	Policy Decision Point
PDS	Personal Data Service
PEP	Policy Enforcement Point
PEC	Policy Evaluation Component
PHI	Personal Health Information

PII	Personally identifiable Information
PIP	Policy Information Point
QoE	Quality of Experience
QoS	Quality of Service
RBAC	Role-Based Access Control
RG	Request Generator
RSA	Rivest-Shamir-Adleman
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SP	Service Provider
SPId	Service Provider Identity
SSL	Secure Sockets Layer
SSN	Social Security Number
SSO	Single-Sign-On
SSS	Secret-Sharing Schemes
STA	Semi-Trusted Authority
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator
VP	Validating Point
W3C	World Wide Web Consortium
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language

XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

Publications

The results of the research described in this thesis have been published in the following papers:

[1] Fei Li, DasunWeerasinghe, Dhiren Patel, Muttukrishnan Rajarajan, "An User-Centric Attribute Based Access Control Model for Ubiquitous Environment" *Mobile Computing, Applications, and Services.* Springer Berlin Heidelberg, 2012: 361-367.

[2] DasunWeerasinghe, FEI LI, Muttukrishnan Rajarajan, "Novel Framework for Secure Mobile Banking", *In Proceedings of the 5th International Conference on Security for Information Technology and Communications*. 31 May – 1June 2012, Bucharest, Romania

[3] Fei Li, Yogachandran Rahulamathavan, Muttukrishnan Rajarajan, RCW Phan "Low Complexity Multi-authority Attribute Based Encryption Scheme for Mobile Cloud Computing" *Service Oriented System Engineering (SOSE). 2013 IEEE 7th International Symposium on*, pp.573,577, 25-28 March 2013, San Francisco Bay, USA

[4]Fei Li, Yogachandran Rahulamathavan, Mauro Conti, Muttukrishnan Rajarajan, “LSD-ABAC: Lightweight Static and Dynamic Attributes Based Access Control Scheme for Secure Data Access Control in Mobile Environment”*39th Annual IEEE Conference on Local Computer Networks,* 8-11 September 2014, Edmonton, Canada

[5]Fei Li, Yogachandran Rahulamathavan, Mauro Conti, Muttukrishnan Rajarajan, "Robust Access Control Framework for Unified Communications Network" *Special Issue on Security and Privacy in Unified Communications: Challenges and Solutions* (under review)

[6] Yogachandran Rahulamathavan, Suresh Veluru, Jinguang Han, Fei Li,Muttukrishnan Rajarajan, and Rongxing Lu, "User Collusion Avoidance Scheme for Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption", *IEEE Transactions on Information Forensics and Security*(under review)

1 Introduction

Mobile handsets have changed human life to a great extend during the last couple of decades, and it was one of the biggest inventions of the early 1980s.The recent technological advancements in mobile devices and wireless technology enabled users to migrate to portable computers such as smartphones and tablet computers from traditional desktop computers; hence, users can access online services via mobile handsets from anywhere and anytime. This transformation increased the user interaction with computers to a great extent. Various new online technologies and services are introduced to adopt the change in computer landscape. Mobile Cloud computing is one of the emerging technologies where the traditional Cloud computing is being transformed to improve the mobile Internet user experience. Mobile Cloud Computing is defined as the application of Cloud computing in combination with mobile devices over mobile networks [1]. The combination of Cloud computing and mobile technology improves the user experience and productivity, particularly in the working environment [2], e.g. location-based services such as Google Map services via mobile devices provide convenient navigation services to mobile Cloud users. All the necessary distance and route optimizations are done within the Cloud. Mobile devices are not involved into such

high-performance computations. However, this change creates challenges in terms of user data security and privacy. Within this context, this thesis discusses the security and privacy perspective of this trend and proposes novel algorithms to mitigate the risk in identity management and privacy protection. In the following, various components of mobile Cloud computing are briefly discussed followed by contributions and structure of this thesis.

1.1 Motivation

The recent technology advancements in mobile communication networks and the increasing penetration of smartphones are transforming the mobile Internet and empowering the users with rich mobile experience. Today's users prefer to access online data and services, such as using Dropbox for Cloud storage, and sharing photos with friends using Instagram via mobile devices [3]. This trend has fuelled the need to transform the traditional Cloud computing into mobile Cloud computing that can help to address some of the data processing challenges. However, the limitations of smartphones such as onboard computing power, battery time, and storage capabilities of mobile devices hamper their ability to support the increasing sophisticated application demanded by users [4]. Mobile Cloud computing is an emerging Cloud service model that combines the Cloud computing paradigm with traditional mobile networks framework. Researchers

embrace new paradigm of mobile Cloud computing such as mobile storage as a new way to extend the capabilities of mobile devices and mobile platforms, which has the potential to impact the business environment and provide more convenience to individual's daily life.

When users are accessing Cloud based services through their mobile devices, the Cloud service providers may collect data from the users for service delivery. Most of today's Cloud service providers work in a collaboration environment. Consider the following scenario, an Instagram user can either share his photos with other preferred Instagram users, or publish the content on his Facebook page. In order to do so, he has to authorize Instagram with certain access rights on his Facebook account. There are two security concerns in this scenario:

(1) ***Identity leakage***: Users' Facebook identity data will be collected by Instagram. Service providers collect user's attributes or identity related data during the service period. In mobile Cloud computing environment, the user's real identity can be disclosed without permission by a collusion attack from a number of service providers, which is considered as a new security threat from mobile Cloud computing.

(2) ***Data and Privacy Protection***: Users use the Cloud for online data storage. The data managed by Cloud service providers consists of different types of information, such as identity information, location data, and medical records. This information is closely associated with user privacy; hence, the access privileges to access this sensitive information should be strictly controlled. However, existing security approaches are not suitable for protecting the users' privacy in a mobile Cloud computing environment.

The above security and privacy issues are new challenges in mobile Cloud computing environment. Hence new security solutions are required to seamlessly secure the data and online services delivered over the mobile Cloud platform, as well as users' privacy.

1.1.1 Problems with the Current Technologies

Existing security mechanisms of current information systems are becoming increasingly inadequate for today's complex mobile Cloud environments. The majority of current security implementations are adopted from the traditional

approaches, which are not suitable for mobile Cloud environment [3, 5-7]. The major shortcomings of the existing security solutions are summarised below:

Lack of control for users to protect their data: Service providers often hold and process user data. If such data are disclosed or abused by unauthorized service providers, the mobile Cloud users may lose their competitive advantage or even go out of business. Most of the security solutions implemented by the Cloud providers do not allow the data owners to define the necessary access control policies to restrict access for the personal data. Hence, tailored access control techniques are required in order to restrict data access in Cloud computing in order to protect user's privacy.

Lack of protection mechanisms against user identity theft: In general, mobile users obtain different services through various Cloud service providers, e.g. when a Facebook user launches a third-party app in the Facebook App store, the app providers requests the users' identities and data access consent (i.e. to read the friendship details, location information or publish data on the personal home page etc.) in order to deliver the requested service. Personal data such as friendships and location information can be considered as privacy information. Hence, each app provider knows the users' partial identity and privacy information from their Facebook profile.

Today most of the social media apps are accessed from smartphones. Hence it is possible to obtain real-time contextual attributes (i.e. location, network connectivity, app usage)which are available in mobile Cloud environments. If two or more service providers corporate together to combine a user's information that they hold, the user's sensitive privacy information may be illegally shared without the user's consent. In order to protect the privacy of the individual, untraceable identities are required in mobile Cloud computing. The identity management solutions should satisfy the requirements that user's real identity should be protected and be hidden from the service providers.

The centralized architecture is inadequate: In real life, a number of authorities hold different types of attributes for an individual. Higher education institutions have people's educational records. Hospitals store their patients' medical records. These organizations can be considered as attribute authorities who maintain a set of attributes for the user. However, most of the current implementations have only one trusted authority, which acts as the administrator of the whole system, maintains identity management and issues decryption keys. It will become a bottleneck of the system if there are a large number of users. Storing all users' information at one central repository is not an ideal solution too. If a Cloud service provider is attacked then all sensitive information of the users will be compromised.

This thesis focuses on addressing the problems stated above, and proposes robust solutions to overcome the current security vulnerabilities in the mobile Cloud computing environment.

1.2 Contributions of the Thesis

This thesis contributes to the development of a framework for identification, authorization, and data access control in mobile Cloud computing environments. The current security technologies in Cloud environment are examined and a context-aware Multi-Authority Attribute-Based Encryption (MA-ABE) scheme to address the above security issues in mobile Cloud environment is proposed. The MA-ABE scheme enables data owners to define access policies during encryption and is flexible and practical for mobile Cloud computing environment. A number of attribute authorities can work together in a collaboration environment, and issue secret keys to the data owner to secure his personal data that is stored in the Cloud. By capturing real-time context-related attributes via users' mobile devices, a security solution can be achieved for authentication and authorization.

In order to restrict access for users' online data, an Attribute-Based Access Control (ABAC) Model using Extensible Access Control Markup Language (XACML) for mobile Cloud environment is presented. XACML is a privacy access policy language which performs access control based on attributes. Using XACML allows

access rules to be defined in a policy-oriented fashion, with policies being easily updated as rules change and is suitable for mobile Cloud environment.

The following are the major contributions of this thesis:

- **Multi-Authority Attribute-Based Encryption in mobile Cloud computing**: To protect the individuals data and privacy in mobile Cloud environments, the MA-ABE scheme is used to enable data owners to define access control policy while storing encrypted data in the Cloud. A data requestor can only decrypt the data based on his/her attributes. In this scheme, a number of attribute authorities maintain different sets of attributes and issue decryption keys for the user respectively. Users obtain decryption keys from each authority based on their attributes. The model has removed the trusted authority which is responsible for monitoring communications and issuing decryption keys in existing MA-ABE systems. Thus, it improves the security and system efficiency.
- **Incorporation of context-related attributes to conventional ABE shceme:** The available contextual information in mobile Cloud environment is used by data owners during the encryption and decryption process, which strengthens the security and privacy

protection in a more flexible, user-centric, and fine-grained manner. Light-weight cryptographic algorithms are also designed for mobile Cloud computing environment. By utilizing the Cloud infrastructure, the light-weight cryptographic algorithm offloads the high-cost computation and communication overheads to the Cloud.

- **Anonymisation key issuing protocol**: Anonymisation techniques are involved in the key issuing process. An anonymous algorithm is used to mathematically hide users' real identity. Thus, identities are protected from collusion attacks (see in section 1.1.1) in mobile Cloud environments.

- **XACML-based access control model:** A user-centric access control model for mobile Cloud environment is proposed to address user's data privacy issues in section 1.1.1. The model is based on the eXtensible Access Control Markup Language (XACML) and enables users to define their own access control policies to secure their data from unauthorized access.

1.3 Outline of the Thesis

The thesis consists of six Chapters in total. Chapter 2 gives an introduction and related definitions for mobile Cloud computing, identity, privacy and security related technologies. Then main technological building blocks for protecting

identity and data security such as cryptographic schemes, security protocols are discussed in this Chapter. The existing identity management systems such as .NET Passport, The Liberty Alliance, OpenID, Higgins and OAuth are examined in detail in Chapter 3. In addition to this the comparisons of the widely used identity management systems are also given in Chapter 3.

Chapter 4 discusses the guidelines of privacy protection, and discusses the current approaches in privacy protection, e.g. solutions for privacy, data access control, and then proceeds by explaining the main building blocks required for data access control. Several widely used access control models are discussed followed by a brief discussion of privacy protection languages. The current schemes of attribute-based encryption are also examined. Finally, the requirements and challenges for privacy-preserving and data access control for mobile Cloud environment are discussed.

Chapter 5 presents a new user-centric attribute-based access control model using XACML for mobile Cloud environments. The proposed framework enables users to define their own access control polices for their personal data. Real-time attributes are used to secure the data transactions. An attribute authority stores and manages all the attributes used in this study. The proof of concept of the proposed model is given at the end of the chapter.

Chapter 6 proposes three different attribute-based encryption schemes for mobile Cloud environments: 1) context-aware single authority attribute-based encryption scheme 2) context-aware multi-authority attribute-based encryption scheme and 3) low-complexity multi-authority attribute-based encryption scheme. In the context-aware single authority ABE scheme, any context-related attributes that can be captured from user's mobile devices are used during the encryption. In addition to this a context-aware multi-authority attribute-based encryption scheme is proposed in order to address issues of the single authority scenario. Finally, a low complexity multi-authority attribute-based encryption scheme is presented to offload heavy computation and communication tasks from the mobile device to the Cloud. The Cloud infrastructure is utilized to improve the efficiency of multi-authority schemes. The algorithm is designed as a lightweight solution which can secure the data confidentiality and preserve the user's privacy.

Chapter 7 concludes by summarizing the contribution of the thesis and highlights the possible future research directions.

2 Identity, Privacy, and Security in Mobile Cloud Environment

This chapter reviews the concept of mobile Cloud computing, identity, privacy and security as well as outlines important protocols used in this thesis. In particular, new issues rose in identity management, privacy and security technologies in mobile Cloud computing environment are discussed.

This chapter is organized as follows: the Section 2.1 describes the concept of mobile Cloud computing followed by a case study of identity, privacy and security technologies in mobile Cloud environment in Section 2.2. Several security services and mechanisms which are currently implemented in mobile Cloud environments are outlined in Section 2.3. The conclusions are drawn in Section 2.4.

2.1 Mobile Cloud Computing

2.1.1 Concept of Mobile Cloud Computing

Mobile devices now provide online data and services to users from anywhere at any time. Most of the UK mobile operators have established 3G mobile networks and are now moving forward to 4G LTE networks which can provide much better performance compared to the 3G networks [8]. The term mobile Internet refers to the combination of the mobile communication and Internet [9]. Hence, mobile users can enjoy much better mobile data services via mobile Internet. The number of smartphone subscribers within the UK is more than 50% of standard mobile users in 2013 and it is predicted to reach up to 75% by 2016 [10]. Due to the development of wireless communication technology and Web application technology, mobile Internet provides endless space for mobile network and a platform for developing new Internet-based technologies.

Online services for mobile devices are designed using the end-to-end principle. Data and services transactions are taken place between the service providers and end-users. It is coupled with Internet-based technologies through certain standards, such as Extensible Markup language (XML) [11], JaveScript Object Notation (JSON) [12], and Simple Object Access Protocol (SOAP) [13] etc. In today's market, the majority of "smart devices" are mostly based on Apple iOS [14] and Google Android mobile Operating Systems (OS) [15]. They provide powerful solutions for mobile platform to satisfy the growing requirements of mobile users. Developers can freely design innovative mobile applications for mobile users. Apple announced that there are over 75 billion downloads from the AppStore by

June 2014 [16], and over 1.3 million apps available for iOS devices by September 2014 [17]. Meanwhile, Google Play reached 25 billion downloads with over 675,000 apps by 26th September 2012 [18].Google did not release further data on the number of app downloads in the Play market. However, the Play market beats AppStore with over 1 million apps in 2013 [19]. The AppBrain provides real-time number of available apps in Google Play market [20], and there are 1432174 applications available in the Google Play market as of 17th December 2014. These mobile apps give great convenience to people's life. Users are increasing the demand of any sophisticated applications to do more work on their mobile devices. The latest advances in mobile communication networks and the increasing penetration of smartphones are transforming the mobile Internet and are empowering end users with rich mobile experience [4]. However, the limited onboard computing, energy and storage capabilities of mobile devices are hampering the ability of smart devices to satisfy such demands. For instance, it is impossible for a hotel reservation app to run at the smartphones as stand-alone software. Because a user has to download a huge size database which stores all related information, and the cost of operations such as search, update and reserve from such apps are heavy. In order to address these limitations, Cloud computing is identified as a possible solution [21].

Cloud computing has been widely recognized as the next generation's computing infrastructure. It offers some advantages by allowing users to use infrastructure, platforms, and software provided by Cloud servers at low cost [22, 23]. It represents a new paradigm shift in Internet-based service that delivers highly scalable distributed computing platforms in which computational resources are

offered "as a service". In addition, Cloud computing enables users to elastically utilize resources in on-demand fashion. As a result, mobile applications can be rapidly provisioned and released with minimal management efforts or service provider's interactions. With the explosion of mobile applications and the support of Cloud computing for a variety of services for mobile users, mobile Cloud computing is introduced as an integration of Cloud computing into the mobile environment. Mobile Cloud computing brings new types of services and facilities for mobile users to take full advantage of Cloud computing.

In Cloud environment, applications are delivered as services over the Internet. The hardware and systems software in the Cloud data centres provide those services. When a Cloud is made available in a pay-as-you-go manner to the public, it is called a public Cloud; the service being sold is utility computing [24]. Currently available public Clouds in the market include Amazon Elastic Compute Cloud (EC2), Google AppEngine, and Microsoft Azure. The term of private Cloud refers to internal data centres of a business entity or other organization that are not made available to the public. Community Cloud is the Cloud infrastructure which is shared by several organizations and supports a specific community that has shared concerns. Hybrid Cloud is a composition of two or more Clouds (Private Cloud, Public Cloud, or Community Cloud) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.

The National Institute of Standards and Technology (NIST) defines fou4 types of Cloud deployment models and three service models of Cloud computing [25]:

Cloud deployment models:

- *Private Cloud*: In this model, the Cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers. It may be owned, managed, and operated by the organization.
- *Community Cloud*: The Cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns. It may be owned by one or more organizations in the community.
- *Public Cloud*: The Cloud infrastructure is provisioned for open use by general public.
- *Hybrid Cloud*: In a hybrid Cloud, the Cloud infrastructure is a composition of two or more distinct Cloud infrastructures (private, public, community) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.

Service models of Cloud computing

- *Cloud software as a Service* (*SaaS*). The capability provided to the consumer to run their application running on the service provider's

Cloud infrastructure. The consumers are not required to manage or control the Cloud infrastructure such as network, storage, or servers. Users can access the services from various client devices through a thin client interface such as a Web browser (web-based email). The applications such as Gmail and Facebook are the popular mobile Cloud computing SaaS applications.

- *Cloud Platform as a Service(PaaS)*. Cloud consumers are able to develop and deploy their own applications onto the Cloud infrastructure using programming languages and tools supported by the Cloud provider. Consumers do not required to manage the underlying Cloud infrastructure but have the control over the deployed applications and configurations of the application hosting environment. The Amazon AWS is one of the emerging PaaS product which could be used for mobile Cloud computing.

- *Cloud Infrastructure as a Service (IaaS)*: The Cloud consumer has the ability of processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run their software including operating systems and applications. The consumer does not manage the Cloud infrastructure but has control over operating systems, storage, deployed applications and possibly limited control of select networking components.

Mobile Cloud computing aims at using Cloud computing techniques for storage and processing of data on mobile devices to reduce their limitations. It is the combination of mobile Internet and Cloud computing [26]. An illustrative example of mobile Cloud computing is how a smartphone can best utilize the Cloud resources to reduce its energy consumption. A computing task can be either executed on the mobile devices or outsourced to the Cloud. Where to compute the task is dependent on the overhead trade-offs between computation and communication while considering the requirements of applications' Quality of Service (QoS) and users' Quality of Experience (QoE) [27], where the QoS refers to the overall performance seen by a customer [28] and the QoE refers to the difference between expectations and perceptions for a service from a customer [29].

Zhang and Yan proposed a QoS framework to manage QoS assurance in mobile Cloud computing [30]. They used specific factors for mobile Cloud services such as signal strength, mobile operating systems, hardware etc based on the core factors such as reliability, assurance, tangibles, empathy, and responsiveness. Researchers in [31] proposed a solution which is called evidence-based method to measure QoE. They use six factors to provide the evidences, which are review design, quality of review, consistency across services, directness to services and business users of services, numbers of reviews, and other related modifying factors.

2.1.2 Advantages of Mobile Cloud Computing

Mobile Cloud computing improves the performance of many applications on mobile devices. By using mobile Cloud computing, the following limitations of mobile devices can be mitigated:

(1) *Data storage capacity and processing power*: Even though the hardware specification of mobile devices is much improved, storage and processing power are still the major constraints of the mobile devices. For instance, the Instagram [32] service enables mobile users to upload photos to Cloud immediately after capturing the images. Users can browse all the photos from any supported devices. With mobile Cloud computing, users can save energy and storage space of their mobile devices. Such services are today successfully implemented by Flickr [33] and Facebook [34] too.

(2) *Battery life*: Battery life is one of the most important concerns for smartphones. With the modern hardware, mobile devices are equipped with much more powerful processors, and more memory space to process complex computations. However, these new hardware result in an increase of energy cost and may not be feasible for all mobile devices. Hence offloading complex computations to the Cloud can reduce the operational cost of the mobile transactions. In addition it can also help to conserve the battery life-time.

(3) *Reliability*: Cloud can provide more reliable and effective environment to store data and hosting applications compared to the mobile devices.

Factors such as the mobile operating system, battery, hardware etc. can influence the reliability. Furthermore, the damage or theft of mobile devices can cause unexpected loss if data are stored locally. By using Cloud, such loss can be minimal since the data are stored and processed on the Cloud.

(4) *Flexibility and Scalability*: Cloud resources can be consumed in an on-demand, fine-grained, and self-service manner. It is more flexible for service providers and mobile users to run applications without reservation of resources. The deployment of applications can be scaled to satisfy the changes for demands of users, and service providers can easily update their services without considering resource usages.

(5) *Ease of integration*: Mobile users can consume services from different service providers since it is easy to integrate multiple services in the Cloud. An example of multiple services integration is that of Instagram users who can share their photos on Facebook or Flickr if they authorize Instagram with access to their personal data and page.

Due to these advantages, the mobile Cloud computing is largely adopted in Mobile Commerce, Mobile Health, Mobile Education and Mobile Gaming applications. Such mobile applications are developed in a global mobile market. Gartner's report says that by 2016, nearly 40% of mobile application development project will leverage Cloud mobile back-end services [35]. However, there are several drawbacks in mobile Cloud computing that needs immediate attention

before it can take off in commercial arena. The drawbacks associated with mobile Cloud computing are discussed in the next section.

2.1.3 Issues of Mobile Cloud Computing

With the advantages discussed above, different types of mobile Cloud computing services greatly increase convenience, however, new issues are introduced which may affect the QoS and QoE of mobile Cloud computing.

(1) *Network bandwidth and latency*: Bandwidth and communication latency are big issues in mobile Cloud computing since the radio resource for wireless networks is much scarce as compared with traditional wired networks. Bandwidth for 3G cellular systems may be limited by cell tower bandwidth in some areas with low power signal receptions which in turn leads to lower bandwidth and higher latency [26]. The incoming 4G wireless network can address this issue. Wi-Fi is also considered as a solution, but the performance of Wi-Fi depends on the number of mobile users. If there are a large number of users, the bandwidth will also be decreased.

(2) *Network availability and intermittency*: Constant and speedy Internet connection must be ensured in mobile Cloud Computing. Mobile users may not be able to connect to the Cloud to obtain service due to traffic congestions, network failures and loss of signal. In [36], authors proposed a Wi-Fi based multi-hop networking system to address this problem. A list of neighbour nodes is selected by a user, and if there is a loss of connection, the nodes with highest

weight value will continue to receive content. The new Hyper Text Markup Language (HTML) 5 comes with a function of data caching through mobile devices. It is also possible for mobile Cloud application to address this issue.

(3) Resource poverty of mobile devices: The limited onboard computing power and storage spaces were the major issues of mobile devices. With the development of hardware, such problems can be resolved so that the mobile devices are able to display as much information as required to obtain maximum convenience.

(4) *Security concerns*: The advancement in technology has also brought new security challenges. Everyone wants to protect their personal data online. Protecting user privacy and data secrecy from an adversary is a key to establish trust in mobile Cloud environments. The following two aspects of security related issues in mobile Cloud environment should be considered:

a. *Security for mobile user*: Mobile users utilize Clouds for computing and storage resources. Client applications may be required to be installed on the mobile devices. The security of the running environment (i.e. operating system) of client applications should be guaranteed and protected from threats, such as malware, viruses, and Trojans. Meanwhile, the design of client application should authenticate the user and protect it from being used by unauthorized users. On the

other hand, mobile Cloud applications will request user's real-time context information, such as location-based services. Mobile users face privacy issues as service providers may use the important personal information for marketing purposes without prior permission.

b. *Security at Cloud*: Mobile users benefit from storing data and hosting applications on the Cloud. Data integrity and confidentiality on the Cloud is one of the major concerns. It is important that no unauthorized users can reveal the data. Secondly, it is necessary to have access control over all of the user data on the Cloud. Thus, a user can decide which data requester can obtain access and know the purpose of the data usage. Thirdly, different service providers may holds partial identity information of a user. Avoiding collusion attack in mobile Cloud environment is also an important issue.

The above issues are main concerns of the mobile Cloud environments. The main focus of this thesis is on the data security and privacy issues in the mobile Cloud environment and the protection of users' sensitive personal information such as identities and identity related attributes during electronic transactions. In the following sections, a case study for the investigation of current relationships within identity, privacy, and security technologies in presented in the context of mobile

Cloud Computing in Section 2.2. The current security primitives to protect the user identity and privacy in Section 2.3.

2.2 Case Study

Recall the example used in Chapter 1. If an Instagram user wants to share his photos with his friends on Facebook. He needs to authorize Instagram using his personal Facebook account details. Figure 2.1 shows the default authorization settings of Instagram at a Facebook account. Instagram will ask for the permission to contents on Facebook page and access to data at any time. As a result, the user's identity information and attributes recorded at Facebook are revealed to the Instagram service. This is a typical identity information authorization scenario in a mobile Cloud environment.

A mobile Cloud user registers at service provider with a unique identity. For example, a user can use his email address as a unique identity to register at Facebook. The email identity may consist of a set of attributes such as surname, date of birth and place of birth to describe the user. These attributes can be either static or real-time. Attributes such as username, home address, date of birth, and device IMEI number can be considered as static attributes which will not be changed frequently. The identity in mobile Cloud environment normally includes real-time contextual attributes such as users' location records, time, and nearby devices. These contextual attributes can describe privacy information of a user. It is a key challenge to protect privacy while securing users' digital identities in the mobile Cloud environments [37].

It raises an identity security issue that if two such service providers collude together, then they can reveal all the user's attributes. Existing identity management systems, such as OpenID [38], Higgins [39] etc. are designed for traditional PC-based Cloud environment, and do not work with all their features in mobile Cloud environments. Such identity management systems will be discussed in detail in Chapter 3.

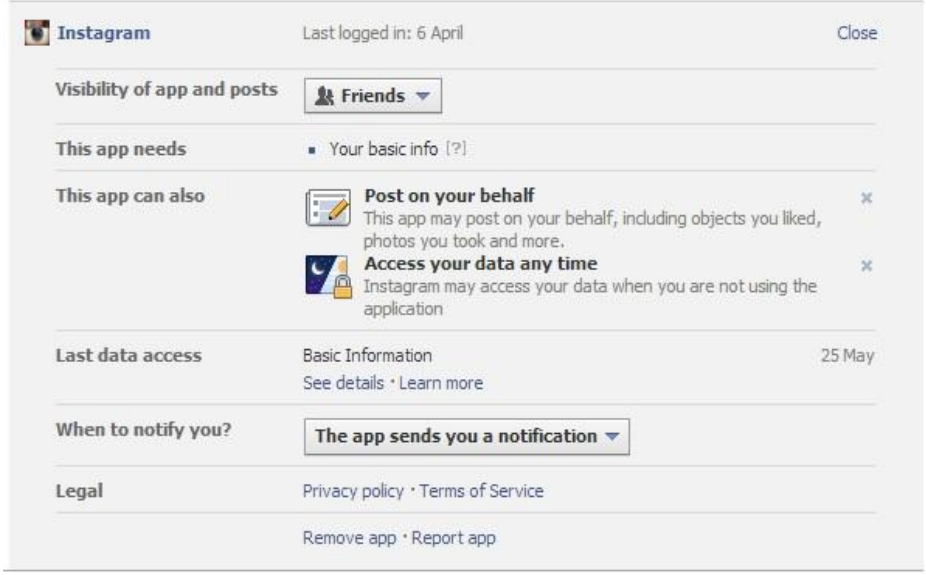

Figure 2.1Authorization Information of Instagram at a Facebook Account.

In today's mobile Cloud environment, mobile apps deliver convenient services to the user with good experience. This trend increasingly affects the user privacy. Let us take the location-based service as an example. Location-based applications are typical mobile Cloud applications. A user wants to book a table at a restaurant with best reviews around his current location. He uses a mobile app which has a large number of restaurants information stored in the Cloud database, such as Tripadvisor [40]. User's mobile device captures the current location and sends it to

the Cloud service provider hosting the mobile app. The app analyses the location and returns the requested results. In order to complete the booking, the user needs to give more details about him to the app. During this process, user does not know whether such personal details are collected by the app or some other third party entities. Furthermore, with a long time usage, the app may have a huge number of data based on the user's behaviours. The user will not know how his personal information will be used and even does not know his data is collected by the mobile Cloud service provider.

Most of the service providers claim that they will only collect anonymous data. However, when the service provider has a large number of users, it may sell the users' sensitive data to a third party who can mine the data to build business models for profits [41-43]. Mobile apps can also be used to inject the malwares to users' phones. Some malwares can even monitor user's daily activities [44]. Researchers have investigated this issue and proposed solutions in order to prevent collusion attacks. Bugiel *et al*. presented a practical security framework for Android platform which addresses the problem of collusion attacks [45]. Their framework can monitor application communication channels in Android's middleware. A system-centric security policy-based mechanism is deployed to enforce access control in Android platform. They also improved the security framework with a system kernel module [46], which enables the framework to provide security protection at system level.

The problem of identity security and privacy is to find a balance between user convenience and computer security. The best way is not to give any information in

order to protect privacy. However, in many cases users are often asked to give some personal information during system authentication process to gain access to data and services. Researchers have also put efforts to address issues raised above. Various access control techniques such as privacy-preserving languages, access control models, and attribute-based encryption are deployed to protect users' data. Chapter 3 and Chapter 4 details those technologies and gives a literature review for related work.

2.2.1 Existing Data Protection Laws

Privacy protection must control what information about an individual is disclosed over the Internet. It includes two aspects: 1) the users should have the control of giving data access consents about their personal information to any authorized entities (i.e. it determines who can access the data) and 2) the control of how the data can be used (i.e. it determines the purpose of data usage).

The Organisation for Economic Co-operation and Development (OECD) also published guidelines on the protection of privacy and trans border flows of personal data [47]. The guidelines have been used to derive privacy laws governing the use of information systems in a number of countries. It introduces eight basic principles for data privacy protection which are listed below.

- ***Collection Limitation Principle***: The collection of personal data and any such data should have limits. And the collection means should be lawful and fair, and, where appropriate, with the knowledge or consent of the data subject.

- ***Data Quality Principle***: Personal data should be relevant to the purpose of use, and the extent for those purposes should be accurate, complete and kept up-to-date.
- ***Purpose Specification Principle***: The purpose for personal data collection should be specified earlier than the time of data collection and the subsequent use limited to the fulfilment of those purpose or such others as are not incompatible with those purpose and as are specified on each occasion of change of purpose.
- ***Use Limitation Principle***: Personal data should not be disclosed, made available or otherwise used for the purpose other than those specified at the collection time, except with the consent of the data subject or by the authority of law.
- ***Security Safeguards Principle***: Reasonable security safeguards should be deployed to protect personal data against such risks as loss or unauthorised access, destruction, use, modification or disclosure of data.
- ***Openness Principle***: There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available for establishing the existence and nature of personal data, and the main purpose of their user, as well as the identity and usual residence of the data controller.
- ***Individual Participation Principle***: An individual should have the right:

a) To obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him;

b) To have communicated to him, data relating to him

I. within a reasonable time;

II. at a charge, if any, that is not excessive;

III. in a reasonable manner;

IV. in a form that is readily intelligible to them;

c) To be given reasons if a request made under subparagraph (a) and (b) is denied, and to be able to challenge such denial;

d) To challenge data relating to him and, if the challenge is successful to have the data erased, rectified, completed or amended.

- ***Accountability Principle***: A data controller should be accountable for complying with measures which give effect to the principles stated above.

The USA government published the Health Insurance Portability and Accountability Act (HIPAA) in 1996 [48]. The act addresses the security and privacy issues of health data. The European Union (EU) published the Data Protection Directive which regulates the processing of personal data within the EU. It defines that personal data can only be gathered legally under strict conditions for a legitimate purpose. Organizations who collect and manage user's personal information must protect it from misuse and must respect certain rights of the data

owners which are given in the EU law [49, 50]. In 2013, the European Commission proposed a major reform of the EU legal framework on the protection of personal data which is aimed at strengthening individual rights and tackling the challenges of globalisation and new technologies.

2.3 Security Concepts, Technologies and Mechanisms

This section examines the current technology building blocks and standards for developing security frameworks in identity management. In mobile Cloud environment, communications and data exchanges of mobile devices are taking place via wireless medium. Similar to Ethernet, three key aspects of information security referred to as *CIA triad: Confidentiality*, *Integrity* and *Availability* [51] must be followed in wireless communication. Additional aspects to security such as *Non-repudiation*, *Accountability* are also referred to traditional *CIA triad* [52]. The key aspects are described below.

Confidentiality refers to the data concealment, preventing the disclosure of information to unauthorized entities.

Integrity protects information resources form modification and deletion by unauthorized parties. It ensures the accuracy and consistency of information resources over the life-cycle.

Availability serves the purpose that the information resources must be accessible when it is required. It ensures that the system work promptly and the services are not denied to authorized users.

Non-repudiation implies the occurrence of an event or an action and the participant. A party cannot deny receiving a request and the other party cannot deny that it is the sender of the request.

Accountability implies the actions or decisions which are made by an entity are accountable.

The concepts listed above are necessary requirements for any security frameworks. These concepts help to protect the information systems from attacks. Security mechanisms are normally designed based on those concepts.

2.3.1 Security Technologies and Mechanisms

Information systems implement security mechanisms to detect, prevent or recover from a malicious attack. It consists of series of security services such as authentication, authorization and access control etc. Various security techniques are used for these services such as cryptography, digital signatures [53], Transport Layer Security(TLS) [54] protocol, and Security Assertion Markup Language (SAML) [13].

Cryptography is the mathematical approach to secure communication in the presence of adversaries. It will satisfy confidentiality of messages. The information should not be revealed to any other parties except the intended recipient. Prior to the modern age, cryptography is synonymous with encryption. The encryption is one of the important areas in cryptographic technologies. The message which contains the fully readable information is called plaintext. The encryption process is used for

scrambling or disguising the plaintext by applying cryptographic algorithms. A ciphertext is generated based on the plaintext after encryption process. The decryption process converts the ciphertext to plaintext. It is the reverse of the encryption process. A key is required for both encryption and decryption to perform cryptographic algorithms. It is a secret value which is sensitive and should not be revealed to any other third party. The key normally is a special numerical value which is long and large enough to prevent from attacks.

The current cryptography technologies can be classified by two main categories: the secret key technology and the public key technology. In the secret key cryptography, the sender and receiver use the same key for encryption and decryption. Therefore, the key is shared as a secret by the two ends but is blind to the outer world. Different from the secret key cryptography, the public key cryptography has a pair of keys, a private key and a public key. The public key is open to any one while the private key is only known to the key holder. Either key of the key pair can be applied for encryption, but only the other key of the same key pair can do the decryption. Based on these two main technologies, several known cryptographic mechanisms are described in the following.

2.3.1.1 Symmetric Cryptography

Symmetric cryptography is the oldest and best-known cryptographic technique. It also refers to the secret key cryptography. The secret key can be a number, a word or a string of random letters. The information system takes as input a secret key to the encryption algorithm with the plaintext to generate the ciphertext. Figure 2.2

depicts the work flow of the symmetric cryptographic mechanism. There are two commonly used types of symmetric encryption algorithms that are listed below:

Block ciphers takes as input a block of data and encrypt them as a single unit with a key. Blocks of 64 or 128 bits have been widely used. Such algorithms are implemented in a variety of ways. For example, the Advanced Encryption Standard (AES) [55] and the triple-DES (Data Encryption Standard) [56].

Stream ciphers encrypt the digits (typically bytes) of a message one at a time. The stream cipher algorithm proposed in [57] is a widely used model.

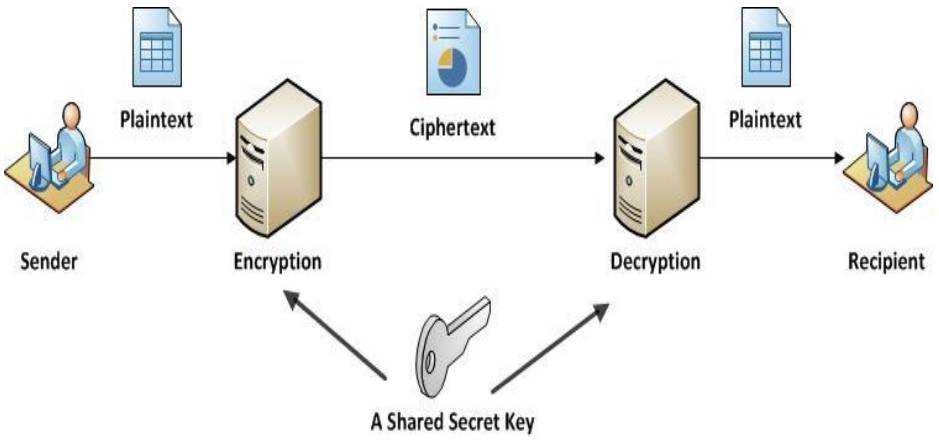

Figure 2.2Symmetric En/Decryption Process

2.3.1.2 Asymmetric Cryptography

In symmetric cryptography, securely exchanging the secret keys over the Internet or public domains is a critical issue. Users have to prevent from sending them to wrong recipients. Asymmetric cryptography is a solution. It is also known to as

public key cryptography. Instead of using pre-shared secret keys, every user must have a key pair. The key pair consists of two keys: a private key and a public key.

Both the private and public keys are generated or acquired by owner of the key pair. The public key is publicly available and can be used for encryption, verification of a digital signature (digital signature will be discussed in the next sub section). The private key is secured at the owner and is kept as a secret. In a public key scenario, a sender encrypts a message with the recipient's public key. The receiver who holds the corresponding private key can decrypt the message. Figure 2.3 describes work flow of the asymmetric cryptographic mechanism.

There are several existing asymmetric encryption algorithms, the EI Gamal [58] and Rivest-Shamir-Adleman (RSA) [59] are the most widely used public key schemes.

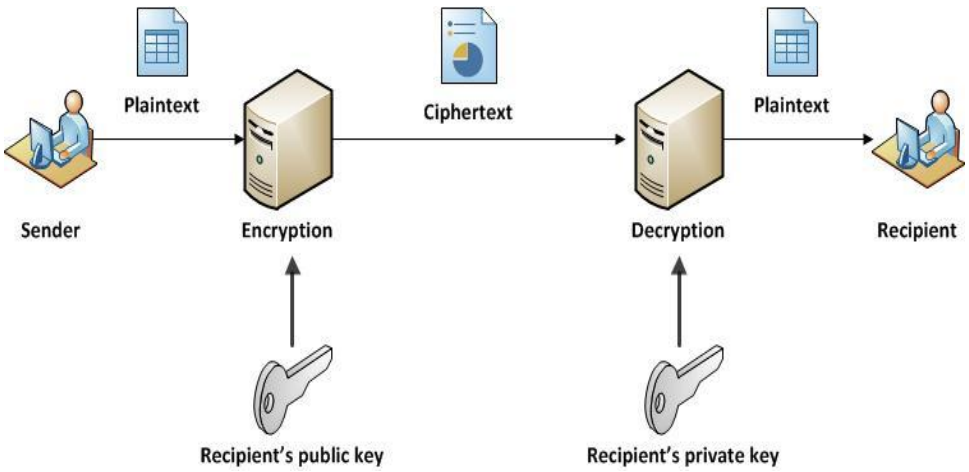

Figure 2.3Asymmetric En/Decryption process

2.3.1.3 Digital Signature

Digital signature supports the non-repudiation aspect of the information exchange. In digital signature scheme, a message is signed by the sender using the private key. The recipient can verify the signature using the sender's public key. The digital signature scheme gives an evidence for a recipient to believe that the message was sent by a known sender. The sender cannot deny the fact of having sent a message. It is used to preserve non-repudiation and authentication. The Digital Signature Algorithms [60] (DSA) and RSA [53]are commonly used.

2.3.1.4 Hash Function

Hash function is used to provide integrity of messages. A hash function is a public function which maps any length of data input into a fixed length output. Given a message ***M*** of arbitrary length, a hash function ***H*** will produce a fixed-sized hash value ***h*** in a form of ***h = H(M)***. ***h*** is a hash value of message digest. The properties of a hash function [61] are listed as follows:

One-way property: *H(x)* is easy to compute *h* with any given input *x*, however it is computationally infeasible to compute such that *H(x) = h*.

Weak collision resistance: For any given x, it is computationally infeasible to find $y \neq x$ with *H(y) = H(x)*.

Strong collision resistance: It is hard to find any pair (*x, y*) to compute such that *H(x) = H(y)*.

By applying a hash function, the secret parameters such as password, secret keys can be verified based on the hash values of them. It greatly helps for data

validation with data privacy. Secure Hash Algorithm (SHA) is a hash algorithm and it was developed in 1993. The cryptography hash algorithm SHA-1 was published in 1995 with a maximum input length 2^{64} bits and outputs a 160 bit message digest[62]. However, a successful attack on SHA-1 was reported in 2005 [63] and in order to avoid these problems, more advanced algorithms such as SHA-2 and SHA-3 in SHA family are used in modern day secure communication.

2.3.2 Protocols and Standards

2.3.2.1 Transport Layer Security (TLS)

The Transport Layer Security and its predecessor, Secure Sockets Layer (SSL), are both cryptographic protocols which provide a channel communication security over the Internet [64]. It secures the communication channels above the IP layer and below the application layer. In a TLS/SSL scenario, asymmetric cryptography is used for authentication of key exchange. Symmetric cryptography is implemented for data exchange.

SSL was first developed by Netscape [65], it enables a client-server application (normally is a Web browser and a Web server) to exchange data over public domains. Data confidentiality and integrity are ensured. The latest version of TLS is published in 2008 [54].

2.3.2.2 eXtensibleMarkup Language (XML):

eXtensibleMarkup Language (XML) is a self-descriptive, platform-independent markup language. It is free, open standard and resembles a general-purpose version of HTML. XML is designed to represent, transport and/or store a range of

structured data. It enables users to define their own tags in XML document and is widely used for business data share and exchanging in different domains. XML parser is a software program that checks the syntax of the XML document and makes the XML document's data readable to applications. XML schema is a method of describing the semantic rules for a XML document to be validated. A XML parser firstly checks whether the document is well formed, the term of well-formed means satisfy all XML syntax rules. Then the parser will check whether the document is Valid, which means satisfied semantic rules that are in XML schemas. W3C published the XML specification in 2008 [48].

2.3.2.3 Simple Object Access Protocol (SOAP)

SOAP is a standard lightweight transport protocol in Web services used for exchanging structured information in a decentralized, distributed environments over HTTP protocol. SOAP specification defines, using XML technologies, an extensible messaging framework containing a message construct that can be exchanged over a variety of underlying protocols [13]. In SOAP message systems, different applications can interoperate by negotiating both parties into a common data transfer protocol. Typically, a simple SOAP message is enclosed inside an SOAP envelope. The envelope contains two main regions such as the SOAP Header and SOAP Body. The SOAP Header contains information about the SOAP message and the SOAP Body contains the message payload.

2.3.2.4 Security Assertion Markup Language (SAML)

SAML is an XML-based framework that allows the exchange of authentication and authorization information between two Web service domains. There are four key elements covered by the SAML specification.

SAML Assertions: a pack of XML-based data includes one or more statements made by a SAML authority.

SAML Protocols: Defines data structures of request and response messages on how to obtain assertions.

SAML Bindings: Map SAML protocol messages onto standard communication protocols, e.g. HTTP.

SAML Profiles: Describe particular combination of SAML assertions, protocols and binding for a particular user case.

SAML 1.0 was first adopted as a Web standard by OASIS in 2002 [66]. After several revisions, in 2005, SAML 2.0 was announced [67]. It has been widely adopted by government, higher education and commercial enterprises.

SAML protocol messages are transported within the SOAP body and SAML response has SAML status and one or more assertions. The assertion consists of one or more statements which contains a single authentication statement. SAML is widely used in federated environment because it enables the functionality that users can authenticate themselves in one Web service domain without re-authentication in another domain.

2.4 Summary

The mobile Cloud technology is a combination of Cloud computing and mobile Internet technologies. In general, when a user interacts with a service provider, some static user attributes may be gathered by the service provider. However, in mobile Cloud environment, not only static attributes but also dynamic user attributes such as location records and user behavior are collected by the service providers. Hence, this new notion adds various new threats to the user privacy. Identity management systems manage users' identities; it is also necessary to modify the security concepts to cope with the emerging mobile application trend. In particular, the current security concepts used in identity management is powerful enough for securing data and service transactions. Even though, smart mobile devices can use these security features, these is a necessary to develop new techniques to protect the privacy of real-time dynamic attributes in mobile Cloud environments.

Due to the nature of mobile Cloud environments, service providers are now collecting much more sensitive information about the user with or without their consents. Users should have full control over their own data. Security technologies such as access control, and cryptography should enable the users to disclose personal data based on their preferences. The existing technologies are able to secure data transaction and confidentiality, but they cannot cover such new privacy

issues in today's mobile Cloud environment. Besides security, privacy protection has become a critical issue in modern day mobile applications.

3 Identity Management Systems

As the number of mobile Cloud services requiring user authentication continues to grow, so does the number of digital identities. To help protect digital identities, a range of identity management systems have been proposed to address security issues of identity protection and identity management. These proposed systems are designed to simplify the management of a large number of identities, meanwhile, protect identities against cyber-attacks. This chapter briefly surveys various identity management systems in the literature.

This chapter is organised as follows. The concept of Single-Sign-On (SSO) and federation which are implemented today by most of the identity management systems is introduced in Section 3.1. In Sections 3.2-3.6 describe .NET Passport, the Liberty Alliance, OpenID, Higgins, and OAuth respectively. A discussion of the above identity management systems and literature review is given in Section 3.7. Finally, the Chapter is concluded in Section 3.8.

3.1 Single-Sign-On(SSO) and Federation

Different online services may be hosted by different service providers. Users have to do the authentication to each service provider separately in order to access the different the Cloud services. SSO is a technology that lets users authenticate to a single authentication authority once so that they can access all authentication protected resources and services without re-authentication within the federated environment. Therefore, one login action enables user access to all the permitted resources in a complex system without entering multiple usernames and passwords. SSO provides convenience to both the user and service provide. Users do not have to memorize many usernames and passwords and at the same time, it also reduces the number of authentication requests to the service providers.

Federation, or Federation of identity, describes the technologies, standards which serve to enable identities to obtain access across autonomous security domains. A user of one domain can securely access data or services seamlessly from networks of other domains. SSO is a subset of the federated identity management framework. The user authenticates to the authentication server by submitting credentials such as username, X.509 certificate and identity related attributes etc. The credential includes the user identity and security information. The user identity is known as a federated identity. It is used in order to permit the user to gain access across different domains which are covered by the same authentication server using the same identity. The authentication server is named as

an *identity provider*. It takes charge of the authentication and authorization and maps the federated identity to other user identity credentials.

SSO not only provides convenience to traditional PC users in the past, but also can continue to play an important role in today's mobile environment. The simplification of identity management delivers excellent user experience in mobile environment. Identity management systems satisfy requirements for security and user experience in different mobile environment. There are several identity management systems in today's market. In the following sections, these identity management systems are briefly described in detail.

3.2 Microsoft .NET Passport:

In 1999 .NET Passport was introduced by Microsoft [68]. In a .NET Passport system, the users can access multiple websites using a single set of credentials. Microsoft presents .NET Passport system as their web-based SSO solution. A Passport user gives permission to the Passport server to the Passport account credentials. The credentials consist of username, password, email address, first name, last name, date of birth, etc. The sites deploying Passport services are called participating sites. Microsoft provides Passport services as a centralized authentication server.

.NET Passport frees the user from multiple registrations and memorizing passwords. The following figure explains how the Passport system works.

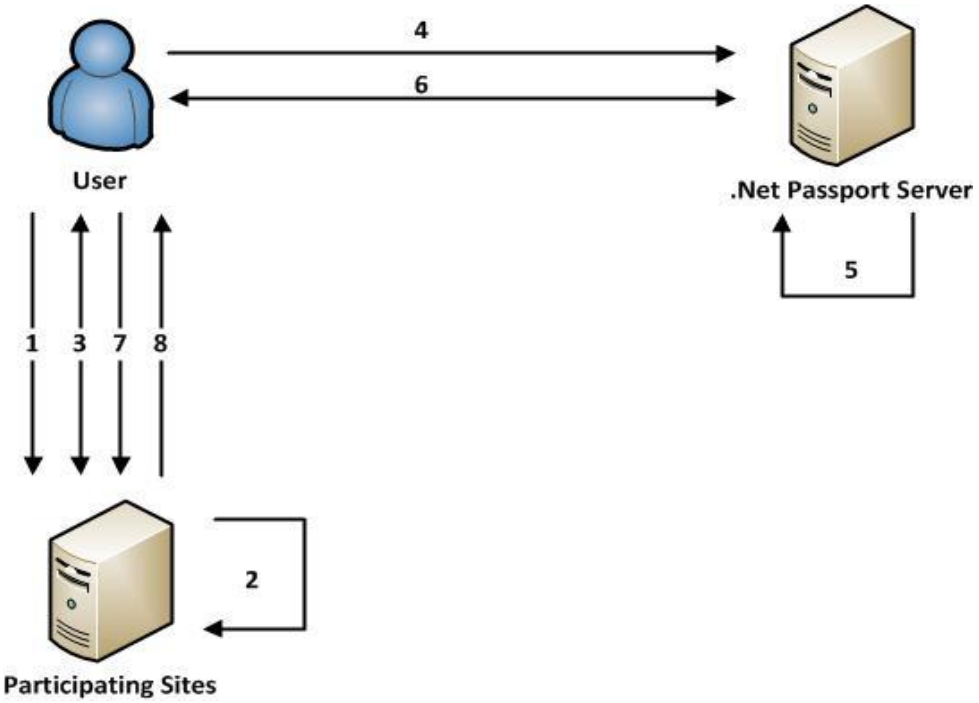

Figure 3.1.NET Passport Authentication Process

(1) User requests resources from a Passport participating site via the browser

(2) The participating site generates a Passport object and forward to the browser to check whether there are any valid passport cookies

(3) If there are any existing cookies, the page loads with a sign-out link and consider the user as an authenticated user. Otherwise, the page loads with a sign-in link.

(4) The user clicks the sign-in link to start the authentication process. The page then redirects to the .NET Passport sign-in page. The user's browser sends the Passport authentication request to the .NET passport server. The request contains the Passport Manager Object, the

participating site ID, the return URL to the participating site after authentication

(5) .NET Passport Server checks the validity of the participating site ID and returns the URL and the Passport sign-in page

(6) If the user is successfully authenticated by the .NET Passport server. The server generates encrypted cookies based on user's Passport User ID and the Passport profile and then forwards these data to the user's browser with the redirection URL. Cookies are encrypted by the participating site's public key. The communication is secured by TLS/SSL channel encryption

(7) The browser redirects the page to the participating site with the encrypted cookies

(8) The Passport Manager authenticate the cookies and grant access to the user for accessing the site

.NET Passport is a SSO mechanism for systems that are connected over the Internet. User authentication is implemented via web forms. Although today's mobile browsers, such as Safari and Chrome are designed with many important functionality to execute operations as a traditional PC browser, the .NET Passport is not a suitable for mobile Cloud environment because of its browser-based feature and lack of participation from users.

In 2008, Microsoft announced that .NET Passport becomes an identity provider of the OpenID framework [69]. The users of .NET Passport system are allowed to access any web sites that are covered by the OpenID authentication.

After this planned implementation in 2009, there is no further update for .NET Passport [70].

3.3 The Liberty Alliance(Kantara)

The Liberty Alliance is a large consortium which is established in 2001 by approximately 30 organizations. Now it has over 150 diverse member companies and organizations representing leaders in IT, government, and finance from across globe to build open, standards based specifications for federated identity management. The vision of the Liberty alliance is to establish privacy and security best practices as well as implementation guidelines. It also collaborates with other standards bodies with an eye toward adopting or extending other specifications [71]. From the mid of 2009, the work of the Liberty Alliance project has been to contribute to the Kantara Initiative [72]. Figure 3.2 depict the abstract view of the Liberty architecture.

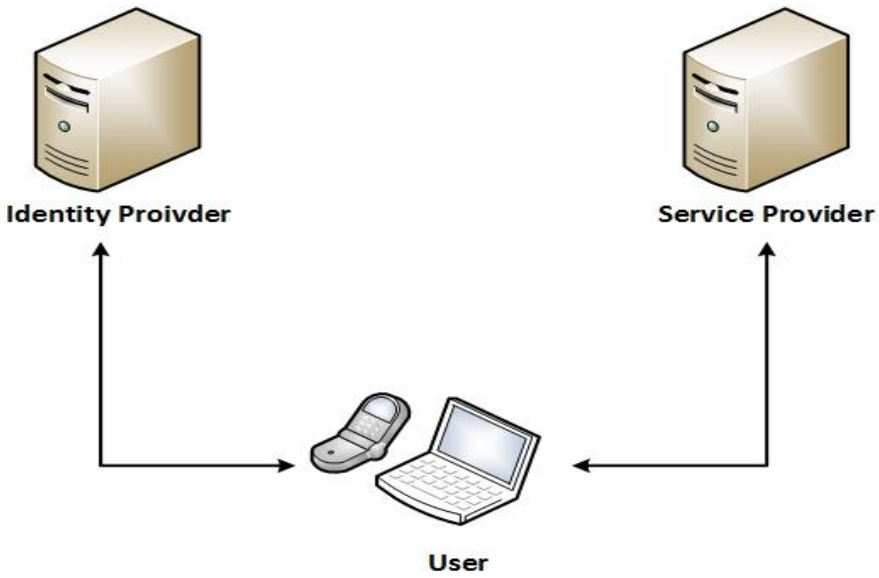

Figure 3.2 The Abstract View of Liberty Architecture

The Liberty Alliance project consists of a number of modules. Each module aims to solve different problems in identity management. The following section will discusses them in detail.

Liberty Identity Federation Framework (ID-FF): The Id-FF enables identity federation and management. A Liberty Federation allows users to authenticate and sign-on to a domain once from any device and then access to the partner Web sites without re-authenticate. It is independent of platform, network devices, computing and mobile devices [73]. It provides approaches for implementing federation and SSO.

Liberty Identity Web Services Framework (ID-WSF)[74]: The ID-WSF is an open framework for deploying and managing a variety of identity-based Web services. It contains a set of specifications for creating, using and updating various aspects of identities. The ID-WSF also presents security functions for privacy protection.

Liberty Identity Services Interfaces Specification (ID-SIS): The ID-SIS is an open framework which contains a collection of specifications. These specifications are designed for the interoperability of different services [75].

Liberty Alliance is based on the notation of 'trust circles' which are formed by a trusted authentication server and set of service providers. The trust relationship of the authentication server and service providers are supported by contractual agreements and it is outside the scope of Liberty Alliance specifications. According to the specification, the user authenticates to the authentication server and it expresses the authentication assertions to the relying

service providers. The assertion consists of the name identifier of the user for service providers to identify the user. The authentication server uses distinct identifiers for each user with different service providers. Figure 3.3 describes the Liberty notation of Trust.

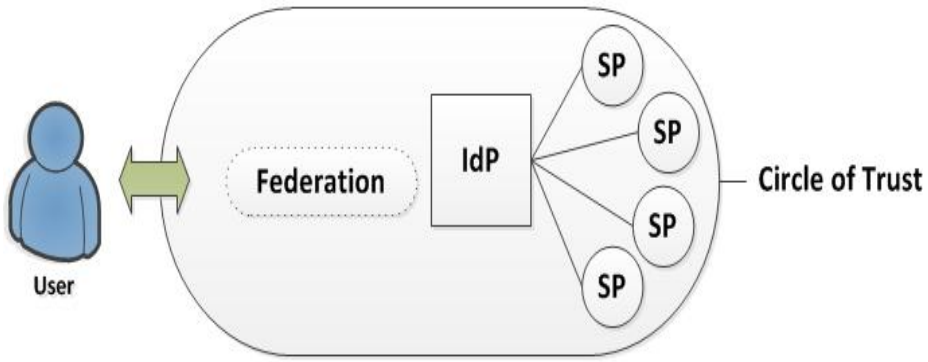

Figure 3.3The Liberty Notation of Trust

The Liberty Alliance supports SSO and use SAML token for authentication. Using SAML also enable it to bring attributes within the communication messages. However, it does not support attribute exchange which is critical in mobile Cloud environments.

3.4 OpenID

OpenID is an open source and decentralized identity management system that supports Web SSO to multiple sites using a single digital identity [76]. The OpenID framework was announced in 2005. An OpenID user can create an account with a preferred OpenID identity provider and then use the account for signing into any sites which accepts OpenID authentication. There are three main actors under the

OpenID framework: the user, the Relying Party (RP), and the Identity Provider (IdP). The RP is always known as the service provider from which the user requests services.

In the OpenID framework, three parties are involved in the authentication process. They are theOpenID provider (OP), which stands for the identity provider; the Relying Party (RP) which is also called service provider and the user. Assuming that the OP and the RP has an existing trust relationship. OP has a trusted list of RPs. In the OpenID model, the users only need to have the identity and password details. Figure 3.4 describes the workflow of OpenID framework.

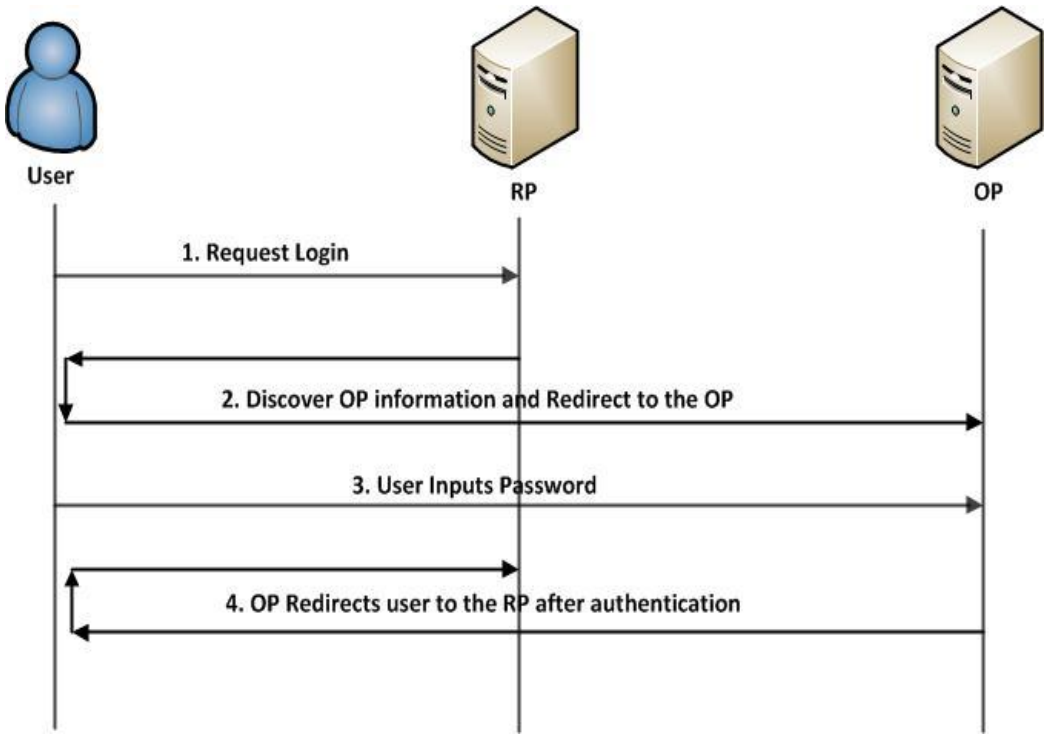

Figure 3.4OpenID Authentication Process

(1) User requests to login into the RP and submits the credential

(2) RP discovers the OP information from the user's credential and redirects the OP to the user

(3) User enters the password into the field on the OP's page

(4) If the user's credential is verified, OP redirects RP's page to the user

OpenID users can choose a trustworthy OpenID server to register their OpenID account. They are identified as a URL: http://yourname.openidserver.com. The RP parses the URL and will get the OP's address "openidserver.com" and the user's identity "yourname". Then the RP can redirect the OP to the user for authentication. The authentication process will make use of the identity "yourname" and the user will inputs the password. If the authentication is successful, OP will redirect RP's page to the user. In the whole process, user is not required to reveal the password to the RP which protects the privacy [77]. However, limitations are discussed in [78] that OP only uses one password to authenticate the user which is not secure enough. OpenID is a symmetric cryptography based framework, which is simple to use but more vulnerable than asymmetric encryption. Moreover, it is a web browser based mechanism and it utilizes the cookies which is not suitable for the mobile Cloud environment.

There are two main versions of OpenID specification: OpenID 1.1 [79] and OpenID 2.0 [38]. The 2.0 version is compatible with the 1.1 version. OpenID has been largely adopted with one billion accounts and nine million sites enabling

OpenID authentication services [80]. IT enterprises like Google, Yahoo, and Microsoft are acting as the OpenID Identity Providers.

3.5 Higgins

Higgins [39] launched in 2003, an open-source Internet identity framework designed to integrate identity, profile, and social relationship information across multiple sites, applications and devices. Higgins is InfoCard-based and involves an active client.

Higgins provides a Personal Data Service (PDS) that let each user control how their personal data is shared with friends and organizations they trust. A PDS is a Cloud-based service that works on behalf of a user, the individual. It gives the user a central point of control for his personal information. The user's interests, contact information, address, profiles, affiliations, friends etc. can be considered as user personal data. A PDS is a place where a user establishes bi-directional data flows between external businesses and his PDS, or between a trust recipient's PDS and his PDS. Figure 3.5 depicts the architecture of Higgins 2.0.

Higgins uses the concept of cards to store information between the user and the other entity. The entity can be either a service provider or a person. In Figure 3.5, the cards shown in green represent a relationship between the user and an external site or business. The relationship includes a bi-directional data connection that shares and synchronizes a set of attributes between the site and the user's PDS.

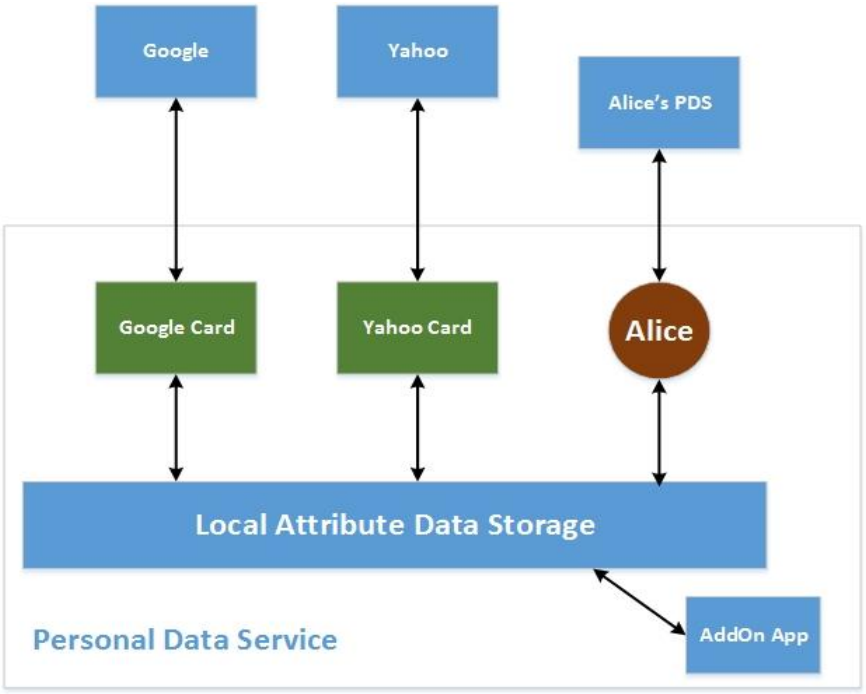

Figure 3.5Architecture of Higgins

The profile shown in brown circle represents a relationship between the user and another person. The relationship includes a bi-directional data connection that shares and synchronizes a set of attributes representing a friend.

The blue box shown as *AddOn App* is a built-in app or extension app which has access to the local storage and update, add value to what's available irrespective of any particular relationship connections.

The local storage for data storage holds user's data using the vocabularies of the Higgins' Persona Data Model 2.0. It exposes these data to the Portal or a browser extension via an HTTP/Comet messaging interface.

Higgins provides online identity management services in a profile/card based manner. A card is established for the user to control the data release to a preferred recipient. However, there are some limitations of Higgins. Although the client

apps are design as browser-based or a stand-alone client app, they cannot run perfectly on mobile devices at the moment. This is a major drawback in mobile Cloud environment.

3.6 OAuth

OAuth is an emerging, open, identity management standard. It describes a system which is designed to enable an end user to grant an Internet application controlled access to personal information (e.g. user attributes, photos, location records) stored at a third party site, without divulging long-term credentials such as passwords. Briefly, it allows a user, to grant access to your private resources on one site to another site. While OpenID in Section 2.6.2 is about using a single identity to sign into many sites, OAuth is about giving access to your stuff without sharing your identity at all [81].

There are four entities involved in the OAuth protocol:

(1) *Resource Owner*: refers to an end user of a hosting site.

(2) *Client*: refers to the application (requesting site) to user's resources.

(3) *Resource Server*: refers to the hosting site of user's resources

(4) *Authorization Server*: refers to the server that issues an access token to a client after successfully authenticating the user and obtaining its authorization.

OAuth 2.0 [82] was published in 2011 and many IT leading enterprises such as Facebook, Google, LinkedIn are on the list of OAuth service providers. Recall the Instagram scenario, the OAuth protocol enables an Instagram user authorizing on Instagram access to personal data of his Facebook account. It is easy to use and saves plenty of time for the user. An OAuth user can freely choose which service provider can obtain the authorization to his personal content from the hosting site, but also define which part of data and time period that the requesting service provider can access. Your identity credentials such as password and username will not be revealed to the requesting services party. Hence user's privacy is partially protected.

The following Figure 3.6 depicts the work flow of the OAuth protocol.

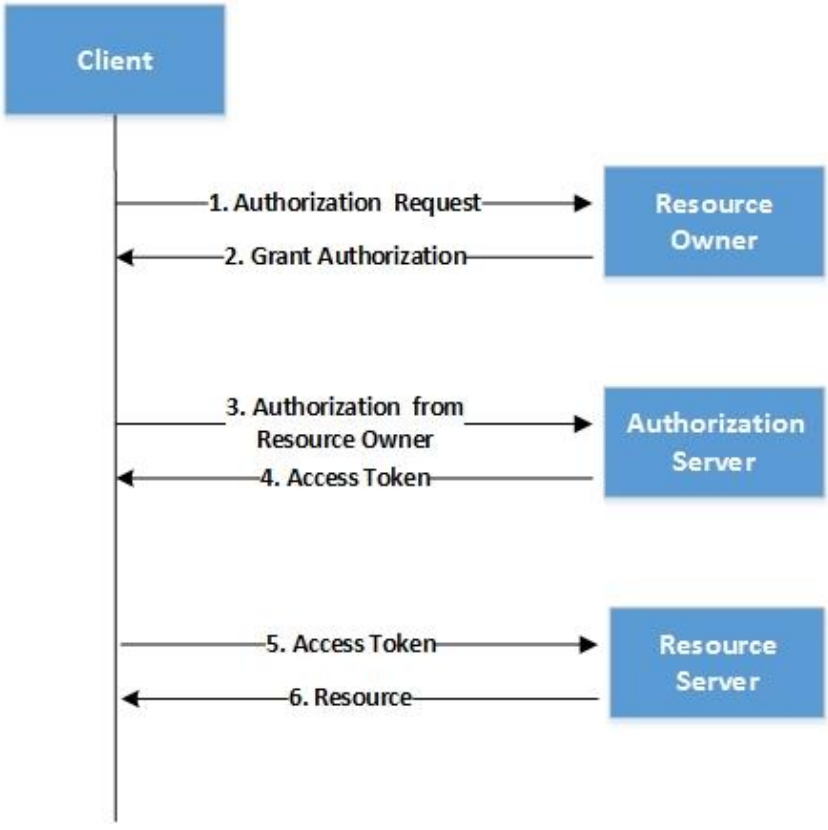

Figure 3.6Workflow of OAuth Protocol

Although, OAuth limits new sites from obtaining all of user data and selling it against your will, it cannot prevent abuse of user's personal data. One of the largest OAuth implementation is Facebook which shares user's data with business partners. Those third-party applications in the App Store of Facebook have the rights to request access to personal data such as a user's friendship details, email address, education history, hometown etc. These settings make the user's privacy under threat. If one or more service providers combine their data together, the user will be face with the possibility of collusion attack.

3.7 Comparison and Literature Review

A generic comparison of the above identity management systems is given. Table 3.1 gives the results of comparison, where √refers to supported and ×refers to not supported.

	Type	SSO	Attribute Exchange	SAML Tokens
.NET Passport	Redirect-based	√	√	×
The Liberty Alliance	Redirect-based	√	×	√
OpenID	Redirect-based	√	√	×

Higgins	Client-based	√	√	√
OAuth	Redirect-based	×	√	×

Table 3.1 Comparison of the Different Identity Management Schemes

An identity management system should be able to protect private and sensitive information related to users and process [83]. The private and sensitive data not only contain identity information, but also contain privacy information. The traditional identity management technologies provided by existing identity management systems should be improved to support the emerging mobile Cloud technology. Researchers have done a lot of work in order to improve privacy protection of current identity management systems.

Sun and Yan proposed a generic and flexible solution based on Cloud computing technology [84].They built a Cloud-based trusted identity management system. The system carries out authentication and authorization among heterogeneous mobile networks. They solved the WLAN/Cellular integration issues using Cloud-based technologies. However, they did not consider the privacy protection. Authors [85] presented a framework for mobile identity management. They used strong authentication methods to provide solutions encompass user access, signing and verification of users and transactions. However, strong authentication requires users to do further actions based on the requests, and this framework did not consider the user data protection.

For a smartphone user, the GPS sensor can capture the real-time location data, the camera can be used to recognize a person's face etc. Thus, lots of contextual data can be used to identity a user [83]. Paruchuri and Chellappan [86] proposed an architecture using the sensor data from the smartphone to generate the context and identify the user to the outside world. Various context information such as location, phone call history, Web browser history etc. are used to identify the user. However, their architecture is built and ran on a mobile device, thus it cannot work in a collaborated environment. Kim *et al.* [87] presented a context-aware platform for user authentication in Cloud computing. There are three components in the proposed platform: agent, middleware and Cloud service providers. The agent works on smart devices. It collects the context information, stores those information as a user profile. The profile also contains user's preferences. Thus, a user can initially control the data and specify the object that he wants to reveal his personal information. Middleware is responsible for connecting the agent and Cloud service providers. The proposed platform protects users' identity when authentication with a Cloud service provider, but it does not protect user's online data.

Biometric authentication methods are also proposed as solutions for mobile environment. Witte *et al.*[88] proposed a context-aware mobile biometric authentication system based on support vector machines. They combine context information from the embedded sensors on a mobile device and biometric information together to achieve a higher level security. For instance, a dark environment may reduce the reliability of face recognition. Thus, context-aware

information is employed to improve the performance of the biometric systems. However, their work did not help the user to protect online personal data.

Since the context information is collected from the mobile devices based on the users' behavior, user-centric model is investigated to solve the security and privacy issues for mobile users. The user-centric model enables user control their own data and activities with strong privacy and security protection [89]. Cloud resources should be collected and allocated according to mobile applications and customized for each individual user. Boger*et al.*[90] proposed a user-centric management approach using smartcard. In their approach, users' attributes are stored in the smartcard, thus, a user can control the release of his attributes. However, a user has to carry the smartcard when he needs to identify himself which cause inconvenience and increase the possibility of identity theft if the smartcard is lost. Seo *et al.*[91] proposed an architecture for context management over pervasive networks. They set up a centralized context manager which is able to understanding complex interactions among different contextual entities. Thus, the system can process and deal with different contexts obtained from different domains around a user. Their work strengthens the user-centric model by managing contexts from different providers for the user. Authors in [92] presented an approach empowering end-users to tailor their mobile applications according to their privacy needs. They implemented a mobile application prototype, which can collaborate with different mobile Cloud applications. However, their evaluation results are based on specific application they developed. The compatibility with existing mobile Cloud apps is unsolved.

Based on the research work on context-aware solution and user-centric model, researchers investigated a new approach to identify the user based on the user behaviors. The user's context data on a smartphone such as call history, location records, app usage etc. indicates the user's normal activities. By profiling such activities, frauds can be detected [93].Lee and Song proposed a model to analyze user behavior of privacy management on online social network [94].They first studied the core issues of privacy and then tested conventional theories for the context in online social network. They contributed to describe the privacy issues for online social network applications and their proposed model can serve as a reference for system designers to understand managing privacy on online social networks. Their research was an ongoing project, and did not consider the privacy issues for mobile users. Paraskevopoulos *et al.*[95] developed methods for enabling the extraction and characterization of normal behavior patterns. By studying call activity and mobility patterns, they classified the behaviors that exhibit similar characteristics. The data recorded by cell-towers were also used to analyze the mobility of mobile phone users. Their experimental results showed that the proposed methods can be used to predict events and actions that are possible to happen if some specific circumstances exist. Their research moves one step forward for detecting users' behaviors. Such evidences can be used to identify a mobile user. Lathia*et al.*[96] presented a platform which is called *UBhave*, the *UBhave* is a project that aims to investigate the power and challenge of using mobile phones for digital behavior change interventions. Data is collected from smart mobile devices and user's online social networks activity. By developing tools to visualize the sequential use of the various interventions over time, the

overview of user's behaviors can be provided. The analysis of large data set from the user, the system can be used to promote positive behavior changes. However, the platform still has issues for the energy constrains at the smart mobile devices. Furthermore, the privacy concerns when collection data from a mobile user is still unsolved.

3.8 Conclusion

The .NET Passport, Liberty Alliance, OpenID, and Higgins identity management systems provide the SSO architecture for users' convenience when accessing multiple Web sites in a Web browser. However, such identity management systems do not have a proper client in mobile platforms which is not suitable for mobile users. To a certain degree of protecting user's online privacy, OAuth is a flexible and powerful privacy protection scheme that users have the ability to choose the recipient of his personal data. But it cannot protect the data from abuse by the hosting service provider. The requirements of mobile Cloud user should not only focus on identity management, but also on the privacy protection.

Based on the study of the above identity management system and recent research works, in mobile Cloud environment, users should establish an online storage which stores all his personal data. A privacy-preserving mechanism should be deployed to protect users' Personally Identifiable Information (PII). The existing identity management systems can protect user's identity from potential attacks, but naturally they do not consider the privacy protection and thus lack mechanisms to protect users' data. It is necessary to setup an access control system to restrict access on user's online data and using novel cryptographic technologies to satisfy the requirements in mobile Cloud environment. The behaviour-profiling techniques can be explored to improve the identification of a mobile user to a more fine-grained way.

4 Access Control Technologies

The exponential growth of Internet usage let user's personal data to be leaked to various unknown service providers. Users do not have control over the data collected by the service providers and this may eventually breach the user privacy. Responding to this requirement raises the emerging need of solutions supporting proper information security governance, allowing users managing their information to enforce restrictions on information acquisition as well as its processing and secondary use. Hence, it is necessary to set up a data access control mechanism so that users' data can be protected from being abused. There are three main solutions that can be explored to protect the privacy of the user's data: (1) privacy-preserving techniques to protect user data, (2) deploying access control technique to control usage of user data and (3) storing data in the encrypted form. This chapter reviews the current main access control models, privacy protection languages, and encryption schemes, and then discusses the possible solutions to address the related issues.

This chapter is organized as follows: Section 4.1 discusses background information about privacy-protection. Section 4.2 reviews the access control models and the privacy-preserving languages are examined in Section 4.3. Section 4.4reviews the attribute-based encryption schemes and discusses the challenges in existing approaches. Section 4.5 concludes the chapter.

4.1 Access Control Models

Most of the time, users want to safeguard the information that may be harmful or embarrassing. Thus, the notion of privacy and the notion of control fit together[97]. Access control is a suitable approach for privacy protection in electronic information systems. Access control defines the selective restriction of access to a place or other resources. In the information engineering, the access may mean certain operation (such as Read, Write, Share and Delete) on the resources, or consuming services. Permission to access a resource is called authorization. With the years of research carried out over the past decades, several access control models have been proposed and applied in the industrial applications.

In any access control model, the entities that can perform actions in the system are called *subjects*, while the entities which are representing the resources to which access may be controlled are called *objects*. In a computer information system, both subjects and objects should be considered as software entities rather than human users. Human users may only perform actions on the system via the software entities which are controlled by them. In most of the current information systems, access is approved based on successful authentications. The system will make a decision to grant or reject an access request from an already authenticated subject.

The models that are used by the existing systems are classified into two types, the capability based model and the access control lists (ACL) based model[98]. In a capability based model, the subjects will be granted access to any objects based on an unforgettable reference or capability to the objects. For instances, the possession of your car key grants you access to your car. Such a capability will be transmitted

to another party in order to convey the access. The Plessey System 250 was built in 1970 [99]. It is the first operational computer system to implementation capability based model. The mechanism is different in the ACL based model. The system granting access to a subject depends on whether the subject's identity is on the permission list which is associated with the objects. For example, the security staff of a high-level conference will check a guest's ID to see if his/her name is on the invitation list. Access is conveyed by editing the list. For computer networking, an access control list refers to rules that are applied to port numbers that are available on a host, each with a list of hosts that are permitted to use the services.

The capability based model and the ACL based model have the features to grant access to group members. Such as the doctors in the same department will have the same access privileges. In a higher level of control, users are assigned with a user ID. All the processes started by the same user ID have the privileges corresponding to the user ID. This level of control is still not fine-grained enough. A doctor has the access to modify any patient's records in a department. However, to a history of records, the system should define the access policy so that they won't be easily tampered. There should be a more fine-grained access control scheme to achieve a more flexible privacy protection mechanism. In the following subsections, four main access control models which are categorized as either discretionary or non-discretionary are discussed [100].

4.1.1 Discretionary Access Control

The Discretionary Access Control (DAC) [101] criteria was defined as follows:

"*Means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject*."

The DAC mechanism restricts access to objects based solely on the identity of subjects who are trying to access them. The user may also be restricted to a subset of the possible access types (e.g. read, write, execute) available for the protected resources. Typically, a particular user or set of users have the authority to distribute and revoke the access to an object, which means a particular user with certain access permission can pass the permission on to any other subject without notifying the administrator or the access control system.

The first general model of DAC was proposed by Lampson in 1972 [102]. With several years of research, the DAC mechanism is widely used in variety of implementations, especially in the commercial and industrial environments because of its flexibility. In most of the operating systems, such as Windows, Linux, and Macintosh are based on DAC model[103]. If a user creates a file in one of these operating systems, he/she can decide the access privileges he/she wants to give to other users. Then the operating system will make access decisions based on the access privileges define by the user. However, the DAC mechanism has drawbacks. For the issue of distributing permission, the particular user cannot provide real assurance on the information flow in a system. A user who has the access to read certain data may pass the permission to another user which are not cognizant by the data owner [100]. The data owner loses the control of the usage of data after

sending it to the user. Most of the proposed designs of DAC [104-107]are not lightweight so that it cannot efficiently deployed in mobile Cloud environment.

4.1.2 Mandatory Access Control

The Mandatory Access Control (ManAC) is commonly discussed in contrast to DAC as sometimes it is termed as non-discretionary access control. In ManAC model, a user will be granted access to a resource only if rules exist that allows the user to access the resource. Subjects and objects have a set of security attributes. When a subject requests access to an object, an authorization rule will be enforced by the system, and examines the provided security attributes. A decision will be made based on the attributes and rules.

ManAC model was investigated over decades. Thomas [108]presented a ManAC mechanism for the UNIX file system. In this mechanism, all files have a set of attributes. File attributes consist of name, owner, owning group, access permissions and modification times of a file, and etc. The operating system uses some of these attributes to enforce DAC. DAC allows the owners of files to determine who can access their files by defining DAC access policies. Rjaibi and Bird [109] presented a multi-purpose implementation of ManAC in relational database management systems. Their implementation allows a database administrator to define labels and to set up a database table such that access to a row in that table is based upon the label associated with that row and the label associated with the user accessing that row. ManAC are mostly deployed in centralized systems such as operating systems, and database systems. Li *et al.*

presented a fine-grained ManAC model for XML documents [110]. They extended XML document model to include label information and define new rules to satisfy this extended model. An extra XML file contains security labels and permitted actions are defined. Each request has to satisfy the requirements in the extra XML files. Authors in [111] presented a ManAC mechanism for mobile devices based on virtualization. Their proposed mechanism was deployed inside the virtual machine monitor. The virtual machine monitor is considered as a trusted computing base. All requests from each subject to objects are enforced by the access control mechanism based on policies. Thus, mobile devices were not required to be involved in the access control process. This model moved a step forward for access control in mobile environment at that time. However, it is out of date to support mobile Cloud environment.

In practice, a subject is usually a process or thread; objects are constructs such as files, directories, shared memory segments, etc. For instance, in a database system, ManAC can also be applied. The objects are tables, views, procedures, etc [112] and the subject is the user. The root user (the administrator) defines access rules for each user and the data consent for user is granted based on the rules. Typically, a centralized security policy administrator will control the security policies. Users do not have the ability to override the policy, which is contrast to the DAC mechanism that a subject has the ability to make access decisions and/or assign security attributes. In real world, almost all UNIX-like operating systems uses ManAC as a part of the operating system's kernel[113], such as FreeBSD[114].

4.1.3 Role-Based Access Control

The Role-Based Access Control (RBAC) is an approach to restrict system to authorized users [115, 116]. It is largely adopted in majority of the enterprises and organizations with more than 500 employees which require multi-level security [117].

Within an enterprise or organization, roles are created for various job functions, e.g. manager, director, researcher etc. In a RBAC model, permissions are associated with the roles. Users are made members of appropriate roles, and permissions to certain resources are acquired through the role assignments. The users do not have the access to the resources but only request permission through their roles. This greatly simplifies the management of permissions. There are three primary rules defined for RBAC:

Role Assignment: A subject must be assigned a role in order to exercise permission.

Role Authorization: A subject's active role must be authorized for the subject. This ensures that users can only use roles for which they are authorized.

Transaction Authorization: A subject can exercise permissions only if the permissions are authorized for the subject's active role. This rule ensures that users can only exercise permission for which they are authorized.

Roles are commonly combined in a hierarchy. High-level roles subsume permissions owned by sub-roles. Compared with DAC model, access permissions are controlled by the system not users. RBAC differs from ManAC that is the management of permissions. ManAC controls basic operation permission (e.g. read,

write) based on a user's security attributes. While in RBAC, the system controls complex operation in a transaction (such as an e-bank transaction), or may be as simple as read or write. In such a view, the management of permissions in a system is transferred to and can be considered as the management of roles. since a role in RBAC can be viewed as a set of permissions. Figure 4.1 depicts a standard RBAC model and Figure 4.2 shows an example of the RBAC role hierarchy.

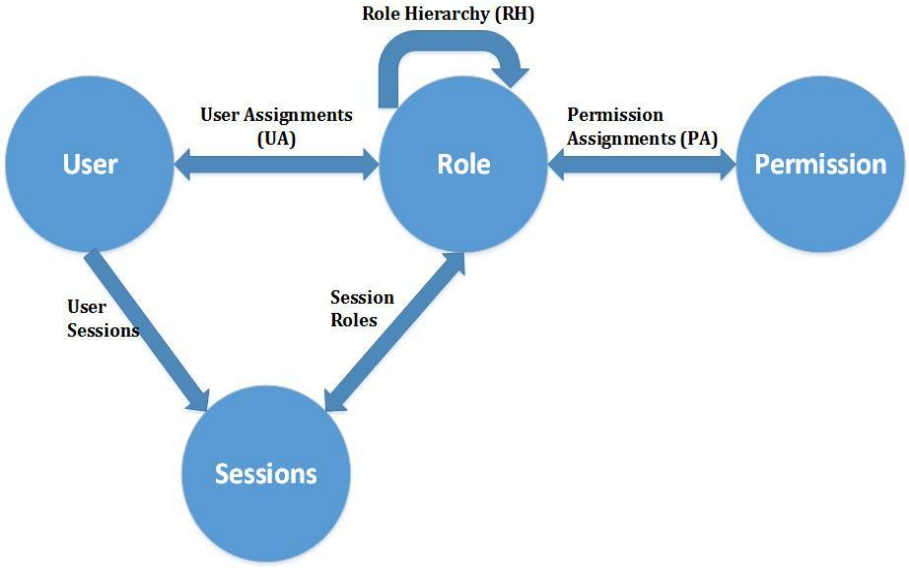

Figure 4.1Illustration of the Elements in a Standards RBAC Model

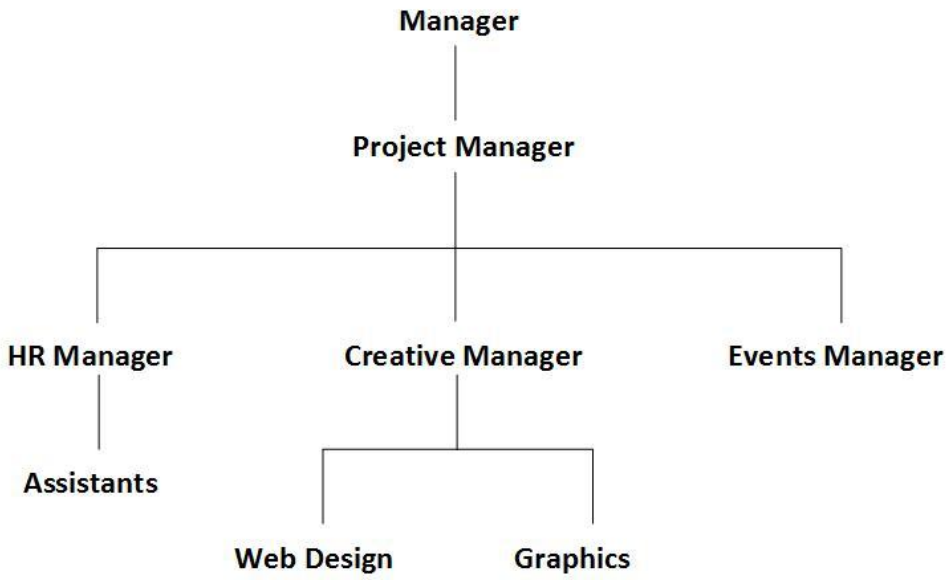

Figure 4.2Illustration of a Role Hierarchy

Researchers have done many works to improve the RBAC model. Elisa Bertino *et al.*[118] proposed the Temporal-RBAC (TRBAC) model which addresses temporal issues related to RBAC. Coyne and Weil proposed a model which is called the International Committee for Information Technology Standards (INCITs) Cyber Security 1.1. The model implemented the access control mapping between difference information systems [119]. The work improves the interoperability cross complex enterprises. Elisa and Catania *et al.*[120] presented a spatially aware RBAC model which involves location-based services. Their work extends the RBAC model to deal with spatial and location-based information. Sejong Oh and Seog Park [121] proposed a task-RBAC model which was founded on the concept of classification of tasks. The task model deals with each task differently according to its class, and supports task level access control and supervision role hierarchy. Strembeck and Neumann [122] presented an context

aware RBAC model. Their approach extended the RBAC services to the enforcement of context constrain. The new requirements for the RBAC can be found in [123-126]. Kulkarni and Tripathi proposed an context-aware RBAC model in pervasive computing [127]. They setup a programming framework to specify and enforce context information during an execution of a role, such as dynamically interfaced services, permission executions.

Traditional RBAC models lack the ability to specify fine-grained access control on individuals in certain roles and on individual objects. This is a big disadvantage when it comes to collaborative environments since their nature requires permissions based on object types. It suits a centralized environment which is different from today's distributed mobile environments.

4.1.4 Attribute Based Access Control

In Attribute Based Access Control (ABAC) model, access is granted based on attributes of the user, not based on the rights of the subject associated with a user after authentication. Unlike RBAC, the ABAC model can define permissions based on just about any security relevant characteristics, known as attributes [128]. Compared with ABAC, RBAC is outdated, expensive to implement and unable to accommodate real-time environmental states as access control parameters. ABAC is newer, simpler to implement and accommodates real-time environmental states as access control parameters [129].

ABAC fully encompasses the functionality of RBAC approaches. The ability to make use of additional resource attributes is suitable for mobile Cloud environment. Attributes such as location, time, date, IMEI number, OS version of a mobile device

can strengthen the security level of data transactions and provide more evidence to the system to make access decisions. Therefore, ABAC can provide a more flexible and complex access control solution. Due to the mobility of a mobile user, such contextual attributes keep changing. Traditional access control model cannot utilize such attributes in run-time. Different combinations of attributes from a subject, an object and environment make ABAC provide a more fine-grained way than RBAC.ABAC is a flexible, scalable, dynamic and fine-grained model, with special features providing great convenience to distributed environments. Moreover, DAC, ManAC and RBAC are initially designed for access control management within a single domain, which is not suitable for mobile Cloud environment because of the collaboration among different mobile Cloud service providers. ABAC is considered as a suitable access control model for mobile Cloud environments.

Attributes are the set of properties that may be associated with a given entity. An entity may be a subject, resource or environment that is considered related to the interaction between a user and an application. Authorization decisions are based on the related attributes, which are established by digitally signed credentials through which credential issuers assert their judgments about the attributes of entities. Because these digital credentials are signed, they can serve to introduce strangers to one another without on-line contract with attribute authorities [130]. Three types of attributes are considered for the access control purpose.

Subject Attributes: A subject is an entity which requests access on an object. The attributes associated with a subject define the identity and characteristics of the subjects. Such as a subject's identifier, name, organization, job, etc.

Resource Attributes: A resource is an entity that is requested by a subject. The attributes of resources can be leveraged to make access control decision with subjects' attributes. The online resources' metadata can be considered as relevant attributes for making decisions.

Environment Attributes: The environment attributes are used for describing the related operational, technical, and even situational environment or context in which the access request taken place. Such as time, location, data, operating system (OS) .etc. The environment attributes are not associated with a subject or an object.

Researchers [128, 130,131] have addressed the issues for deploying ABAC in the Web service architecture. In [84], Bo Lang and Ian Foster *et al.* presented an ABAC model to address privacy issues for Grid Computing. They also provided a toolkit that is called Globus Toolkit release 4 to implement the framework. Bobba proposed an attribute-based message system using ABAC to realize a real-time message system [132]. A location aware ABAC system is presented by Isabel *et al.* to extend ABAC with dynamic authorizations. Lanjing Wang and Baoyi Wang proposed an ABAC model for Web services in multi-domain environment [133]. They presented meta-attribute and meta-policy to describe the attributes and polices in local domain to collaborate with other domains. ABAC is an ideal access control model for mobile environments, and several solutions have been proposed to address the privacy issues in mobile environment [134-137].

4.2 Privacy-Preserving Languages

In order to address the issues in the privacy practices, different types of languages are available to represent the human readable policies in more precise and computer compatible formats [138]. Some languages are designed to help enterprises express their privacy policies in ways that are more amenable to policy enforcement and some languages are designed to help users define their privacy preferences.

Privacy policy languages can help with several of the stages involved in managing privacy policies (e.g. writing, testing, approving, analyzing, and withdrawing). They were designed to express the privacy controls that both organizations and users want to express. Most of the privacy policy languages were designed for specific purposes with specific features and characteristics and most of the initiatives for designing these languages have occurred in the last fifteen years. The W3C began the development of the Platform for Privacy Preferences (P3P) in 1997. CPExchange was developed in 2000 to facilitate business-to-business communication about privacy policies [139]. Then the requirements to express internal privacy polices within an organization from industry made IBM design the Enterprise Privacy Authorization Language (EPAL) in 2003[140]. And during the same period, the OASIS presented the eXtensible Access Control Language (XACML) for both privacy and security policies in a machine readable format [141].

The privacy policy languages are expected to be fairly simple and small. Therefore they have been designed as light-weight XML markup languages. They are not expected to perform high-level mathematical operations or complicated

flow controls. The next three sections will briefly discuss three main privacy policy languages.

4.2.1 The Platform for Privacy Preferences

The Platform for Privacy Preferences (P3P) is developed by W3C in 1997. It is designed to express website privacy policies in machine-readable format[142] and to grant users more control over their personal information when browsing the Web. The P3P user agent allow users to automatically be informed of site privacy practices and to automate decision-making based on the Web sites' privacy practices [143]. Thus, the P3P Preference Exchange Language (APPEL) was introduced to express user's preferences for making automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled sites [144].

A P3P-enabled website will have a set of policies, e.g. stating the uses of personal information that is gathered from the site visitors. With a P3P-enabled Web browser, a P3P user can also define a set of policies, e.g. what personal information can be revealed to the Web sites that they visited. Then when a user visits a site, P3P will compare what personal information the user is willing to release, and what personal information the server wants to get. If the two do not match, P3P will inform the user and ask if he/she is willing to proceed to site and risk giving up personal information.

Although P3P provides the support for privacy for Web sites, the Electronic Privacy Information Centre (EPIC) has criticised P3P, referred to the technology as a "Pretty Poor Policy" [145]. They claim that P3P software is not suitable for

average person to understand because of its complexity and difficulty. Many Internet users are likely to be unable to use the default P3P software. Furthermore, the P3P framework is not initially designed for supporting Web privacy. In the mobile Cloud environments, these disadvantages are critical.

4.2.2 Enterprise Privacy Authorization Language

The Enterprise Privacy Authorization Language (EPAL) is used to formalize privacy authorization for actual enforcement within an intra- or inter- enterprise for business-to-business privacy control [146]. EPAL services themselves are exchanging privacy policies and making privacy authorization decisions. In particular, EPAL concentrates on the privacy authorization by abstracting data models and user authentication from all deployment details.

EPAL has a rudimentary temporal nature that a request to perform an action might lead to allow or deny judgement and obligation [147]. In EPAL, an obligation is usually an action that some agent is required to perform in the future which controls the purpose of data usage after sending to the requesters. The EPAL framework does not consider the privacy for the Web application which is the limitation for mobile Cloud environments.

4.2.3 Extensible Access Control Markup Language

The Extensible Access Control Markup Language (XACML) was formed by the OASIS standards consortium. XACML defines a privacy policy language using the attributes of requestors, resources, and environment. It provides standard XML schema for expressing policies, rule obligations and conditions. Additionally, it specifies a request/response protocol for sending a request and having it approved.

XACML is primarily an attribute-based access control system, where attributes associated with an entity will be embedded into a XACML access request, while the access decision is made depending on whether the access requester satisfies the XACML policy.

The XACML access control framework contains four key main components: the Policy Administration Point (PAP), the Policy Decision Point (PDP), the Policy Enforcement Point (PEP), and the Policy Information Point (PIP).

- *PAP*: The system entity that creates a policy or policy set
- *PDP*: The component that evaluates the applicable policy and renders an authorization decision.
- *PEP*: The system entity that performs access control. It issues decision request to the PDP and enforces the access decision received from the PDP.
- *PIP*: The PIP is where the entire attribute are stored.

Additionally, in most of the systems, a context handler is required to convert access requests into the native request format of the XACML canonical form and convert the authorization decisions in the XACML canonical form to the native response format. Any required obligations should be performed by PEP with the enforcement of an authorization decision [148].

Figure 4.3 depicts the data flow of the standard XACML frameworks. The model operates based on the following steps.

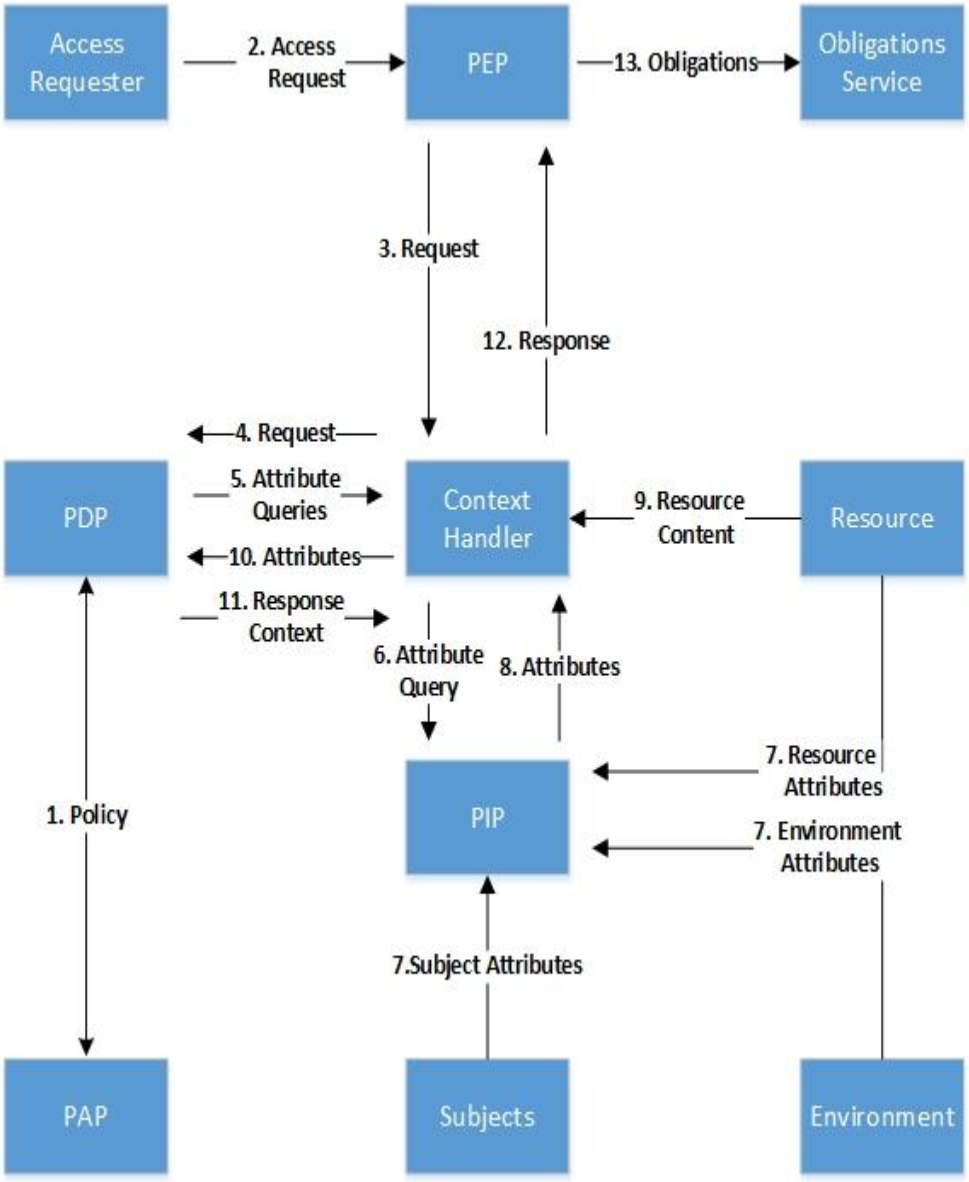

Figure 4.3 Data Flow of a Standard XACML Framework

(1) PAPs prepare and write the policies and policy sets to be available to the PDP. These policies and policy sets represent the complete policy for a specific target.

(2) The access requester sends a request to the PEP.

(3) The PEP sends the request for access to the context handler, in its original request format, including the attributes from all the stakeholders, such as subjects, resource, action and environment.

(4) The context handler constructs an XACML request context and sends it to the PDP.

(5) If additional attributes are required, the PDP will request them from the context handler.

(6) If the additional attributes request received from the PDP, the context handler requests them from a PIP.

(7) The PIP obtains the requested additional attributes from subjects, resource, action and environment.

(8) The PIP returns the requested attributes to the context handler.

(9) The context handler can decide whether to include the resource in the context.

(10) The PDP evaluate the policy after receiving the requested attributes and resource (optionally) from the context handler.

(11) The PDP sends the response context and access decision to the context handler.

(12) The context handler forwards the response context to the PEP after converting it to its original format.

(13) The PEP fulfils the obligations.

(14) If access is permitted, the PEP permits the access; otherwise, it denies access.

Based on the flow, the PDP has no control over the enforcement of the policy decision. This model for access control depends on having every request for protected resources go through PEP, which isresponsible for all enforcements. The implementation of PEP is usually application- or platform-specific, as it is usually built into the application or the platform on which the application is built.

Any policy must be stated in terms of a specific vocabulary of terms. In XACML, vocabulary terms are called attributes, and are defined by the particular applications or domains that use policies – they are domain-specific. The PDP does not need to understand the domain-specific meaning of each attribute used in a policy; it needs to understand only how to determine whether the values supplied for each attribute in an authorization decision request satisfy the conditions for access specified resource in the policy. In order to do this, the PDP needs to know only a unique identifier for the Attribute and the generic data type of the values for that Attribute. It is the responsibility of each application or domain that will be using policies to define an Attribute to correspond to each of its policy vocabulary items. This means assigning a unique identifier for the item, the data type of the values for that item, and the meaning of each value that might be used for that item.

The meaning of the values is not used by the PDP, however, and is significant only to the application or domain. Thus, the format (syntax) of an Attribute is domain-independent, even though the meaning (semantics) of the Attribute value is

domain-specific. The policy language may itself define some standard Attributes that are available for use in any policy.

XACML was largely adopted after being presented in 2002 [149]. Compared with EPAL, the functionality of XACML 2.0 is a superset of EPAL 1.2. The EPAL differences often result in less functionality than XACML has [150]. XACML is designed to support centralized or decentralized policy management. It also supports both general access control and privacy policies, allowing these closely related policy types to be integrated. XACML is both a more comprehensive access control policy language than EPAL, and a full- featured privacy policy language. XACML has already been an OASIS approved standard. With the support of many organizations, XACML are more open to the public requirements. Whenever an issue comes, the researchers can react quickly to address them. There is an active community of users and developers who are continuing to expand improve, and apply the language [151] which gives a powerful and long life to XACML.

Many scientific research works have been done to address issues in the Internet privacy protection field. Claudio extended the XACML architecture and modules for effective credential-based management and privacy support [149]. Anderson and Devaraj proposed a XACML-based Web services policy constraint language (WS-Policy Constraints) [152]. Vivying*et al.* extended the XACML with policy negotiation. They explore XACML's potential in privacy negotiation and introduced a policy negotiation point. Their work addressed issues in the access control negotiation process between Web services [153].

Since XACML provides the support for integrating SAML standard [154], the combination of XACML and SAML gives a more powerful solution for authentication and authorization. SAML can protect, transport and request XACML schema instances and other information needed by an XACML implementation. Ardagna *et al.* described extensions to the access control industry standards XACML and SAML to enable privacy-preserving and credential-based access control [155]. Their extensions allow the requester to learn which attributes have to be revealed and which conditions must be satisfied, thereby enabling to leverage the advantages of privacy-preserving technologies such as anonymous credentials. Hommel presented architecture for the integration of XACML and SAML, it established an access control framework in Federated Identity Management systems [156]. Nils and Vladimir describes a location aware access control model on top of Geographically XACML. It sketches how spatial separations on duty constraints (both static and dynamic) can be implemented. The work provides location dependent access control and security enhancing solutions on mobile devices [157].

XACML can also be used for mobile environments. Qing presented a XACML-based policy-driven access control for mobile environments [158]. They introduce a mapping function and assign a unique ID for the mobile users. An attribute authority takes charge of maintaining the attributes. An access request from a mobile user in an unknown domain can be involved in the local authorization process.

XACML is a powerful privacy access control language which can be deployed in a distributed system. Although in a XACML policy-based system, the message

flows must be formatted in the XACML format, it provides huge convenience for access control that the user can easily define polices to protect data. Such features can help mobile users to protect their privacy in mobile Cloud environments. In the next section, another approach to protect users' data in mobile Cloud environments is investigated.

4.3 Attribute Based Encryption

Besides establishing an access control system, another way to protect privacy in mobile Cloud environment is by storing the data in the encrypted form, thus minimizing the risk. However, data sharing is one of the most important features of mobile Cloud computing. Traditional encryption schemes are one-to-one solutions and cannot efficiently satisfy the user's online data sharing requirements.

Encrypting the data in the Cloud alleviates the above problem [159]. Thus, if the servers which store the data are compromised, the amount of lost information is limited because the adversary does not have the decryption key to obtain the data in the plain text. In a traditional public key encryption method, data is encrypted by a particular individual who has already established a public-private key pair. The traditional method in mobile Cloud environment limits the ability of users to selectively share their encrypted data at a fine-grained level, e.g. a patient wants to grant decryption access to doctors in a specific department of a hospital for his medical records. This can be done by either encrypting the file in a shared group public key, or encrypting the file for each doctor. None of these techniques is

particularly appealing. There is a requirement for expressing more complex access policies during the encryption.

Attribute-Based Encryption (ABE) was proposed by Sahai and Waters [160]. It is a type of public-key encryption scheme in which the key of a user and the ciphertext are dependent on attributes. In an attribute based encryption system, data owner encrypts the data by using several attributes. A decryptor can decrypt the data only if his/her attributes matches the required attributes. Figure 4.4 shows the architecture of Sahai and Waters' ABE scheme.

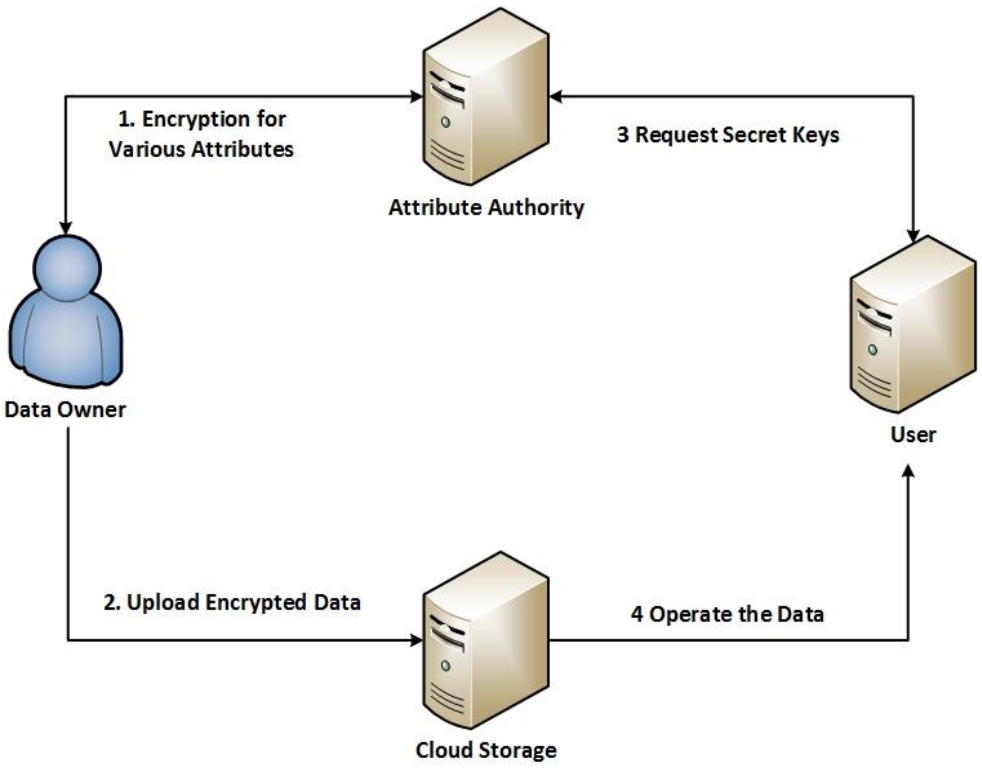

Figure 4.4Illustration of Attribute-Based Encryption Scheme

Sahai and Waters made some initial steps to solve this problem for expressing access polices during encryption. In their system, user's key and ciphertexts are

labelled with sets of descriptive attributes and a particular key can decrypt a particular ciphertext only if there is a match between the attributes of the ciphertext and the user's key. An authority with access to the master keys will issue different private keys to users, where a user's private key is associated with an access structure over attributes and reflects the access policy. The decryption algorithm allows users to decrypt data using their private keys as long as their access policy specified by their private key permits.

The original ABE construction of Sahai and Waters allows, the authority to issue the private key based on threshold access policy. In order to decrypt the data, at least k attributes should be overlapped between a ciphertext and a private key. This primitive feature limits the expressibility of the access policy. For instance, the attribute authority monitors and manages n number of attributes in total where $1 \leq k \leq n$. The data owner encrypts the file with k number of attributes enforced. When a user requests a decryption key for the file, the attribute authority issues a private key for the user with d_k number of attributes. Only if $k \leq d_k$, the user can decrypt the requested file. Sahai and Water's model cannot define more complex access policies, two main extensions of original ABE scheme were proposed afterwards: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE) to address the drawbacks of Sahai and Water's scheme and provide more functionality.

4.3.1 Key-Policy Attribute-Based Encryption

In 2007, Goyal and Pandey [159] developed a richer type of ABE cryptosystem and demonstrated its applications. In their system, each ciphertext is labelled by the encryptor with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. This method is called as Key-Policy Attribute-Based Encryption (KP-ABE), since the access structure is specified in the private key while the ciphertexts are simply labelled with a set of descriptive attributes. Their method supports fine-grained access control to facilitate granting differential access rights to a set of users and allow flexibility in specifying access rights of individual users. The data is stored on the server in an encrypted form; different users are allowed to decrypt the data based on the access policy.

In the ABE scheme, the access policy scheme is based on the secret-sharing scheme (SSS). In 1979 Shamir [161]proposed a construction methodology for secure-sharing scheme in 1979. In his scheme, the access structure is a threshold gate. If there are t or more parties take part in within N ($1 \leq t \leq N$) number of parities, they can reconstruct the secret by using their shares. If there are less than t number of parties, they cannot get any information about the secret. Benaloh [162] extended Shamir's idea to a tree-access structure. The tree consists of threshold and the interior nodes consist of AND and OR gates. The leafs consist of different parties. Any set of parties that satisfy the tree can reconstruct the secret together. Goyal and Pandey proposed a new secret sharing scheme that each private key is associated with a tree-access structure where the leafs are associated with attributes. A Secret-Sharing Scheme (SSS) is used to divide a secret among a number of

parties. The information given to a party is called a share of the secret. Every SSS realizes some access structures that define the sets of parties who should be able to reconstruct the secret by using their shares.

The KP-ABE scheme can be viewed as an extension of Sahai-Water's conventional ABE framework with the embedding of a secret-sharing scheme in the private key. The authority can specify a more general secret sharing scheme for monotonic access trees. They also suggested a new ABE scheme which is discussed in the next section. Figure 4.5 shows the work flow of KP-ABE scheme.

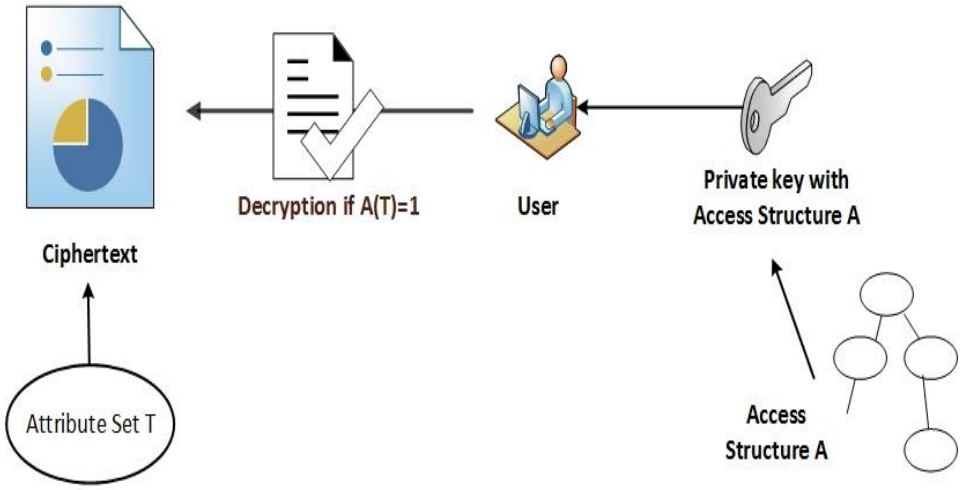

Figure 4.5Key-Policy Attribute-Based Encryption Scheme

4.3.2 Ciphertext-Policy Attribute-Based Encryption

Bethencourt *et al.*[163] presented a new ABE scheme which is called Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In CP-ABE scheme, user's private key will be associated with an arbitrary number of attributes expressed as strings. When a data owner encrypts a message, he/she specify an associated access structure over attributes. A user will only be able to decrypt the

ciphertext if that user's attributes pass through the ciphertext's access structure. The access structures are described by a monotonic access tress, which means the nodes of the access structure are composed of threshold values and the leaves describe attributes. They suggest the AND gates can be constructed as *n*-of-*n* threshold gates and OR gates as *1*-of-*n* threshold gates. Figure 4.6 depicts a working setup of the CP-ABE scheme.

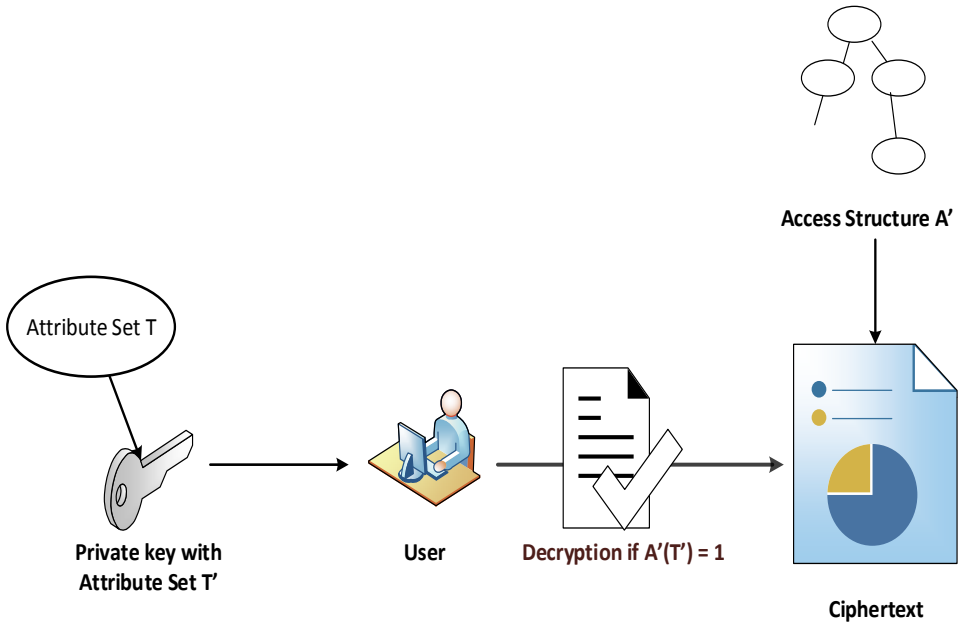

Figure 4.6Ciphertext-Policy Attribute-Based Encryption Scheme

Traditionally, access control is enforced by employing a trusted server to store data locally. The server is entrusted as a reference monitor that checks a user's credentials before allowing him to access data. Most existing public key encryption methods allow a party to encrypt data to a particular user, but are unable to efficiently handle more expressive types of encrypted access control scenarios. Both the KP-ABE and CP-ABE schemes can efficiently address this problem. A

data owner can exert control over who has access to the data he/she encrypts. All the users who satisfy the requirements are able to decrypt the ciphertext.

There is only one authority that issues the private keys and manages all the attributes in both the KP-ABE scheme and CP-ABE scheme making this scheme highly secure. The authority is normally named as central authority or global authority. It verifies all the attributes or credentials for each user in the system and issues private keys to the user. Therefore, it is powerful enough to get all the messages and communications of the user with or without permissions. Users' privacy is at risk if the central authority is compromised. This is the major drawback of the CP-ABE and KP-ABE scheme. Furthermore, since the central authority issues private keys for all the users, if the system is built as a large and global scale system, the authority will become a common bottleneck. If the system spreads the central authority's key over several machines to alleviate performance pressures, it will also raise the risk of key exposure [164].

4.3.3 Multi-Authority Attribute-Based Encryption

Researchers proposed Multi-Authority Attribute-Based Encryption (MA-ABE) scheme to address the issues of a single authority ABE scheme. In reality, different entities are responsible for maintaining different sets of information, such as Driver and Vehicle Licensing Agency (DVLA) managing all the UK driver and vehicle information, National Health Service (NHS) managing UK's patients' information. In a single authority scenario, in order to get medical and driver's information, there must be at least one fully trusted authority that monitors all the information. If the

fully trusted authority is compromised, then all the sensitive data that stored at DVLA and NHS is under risk. There is a requirement to construct an ABE model where more than one authority can operate simultaneously, each handing out secret keys for a different set of attributes.

Chase [165] firstly proposed an efficient MA-ABE scheme in 2007. In her scheme, the data owner has the ability to specify for each authority k a set of attributes monitored by that authority and a number d_k so that the message can be decrypted only by a user who has at least d_k number of the given attributes from every authority. The scheme allows any number of attribute authorities to be corrupted, and guarantee the security of encryption as long as the required attributes cannot be obtained exclusively from those authorities and the central authority remains honest. However the central authority still exists in the scheme but with less functionality. It is stilled fully trusted but handles no attributes and only issues the corresponding keys.

For a user of Chase's model, he will be assigned with a global identifier (GID). The GID has the following properties:

- No user can claim another user's GID
- All authorities can verify a user's GID

Thus, the GID can be an identifying string for which a user has provable credentials and it seems likely that such information would be presented when users' attributes are verified. If two users request the private keys from two attribute authorities, the GID enables the authorities to distinguish the two requests in order to prevent collusion.

Each user will send his GID to the central authority and receive a corresponding key. The central authority will not get any information about the user's attributes. The central authority only issues a setup key for the user's GID and holds the master secret of the system. Although the central authority does not monitor any attributes, it still be able to decrypt any messages with the master secret. Figure 4.7 illustrates the architecture of Chase and Chow's MA-ABE scheme.

Chase's work[165]made a small step forward than the single authority ABE scheme, the main drawback of CP-ABE and KP-ABE still remains with the existence of central authority. In 2009, Chase and Chow presented an improved MA-ABE scheme [166] based on Chase's previous work. In the latest work, the central authority was removed and it provides secure privacy-preserving techniques and better security. In the improved framework, GID is still used as the identifier of users with same properties discussed before. Each pair of attribute authorities would share a secret key. For each user, he/she will request secret keys from each authority and combine them to get the decryption key. The authorities are not allowed to communicate with each other to prevent collusion attacks. Chase and Chow used anonymous techniques to hide the identity of users which means the GID is now completely hidden to the attribute authorities. The authorities cannot track user's attributes from the requests to identify the user and then combine their keys together to get decryption keys of the user. It achieves the pseurandomity, anonymity and untracebility, therefore, the privacy of the user is protected. They also provide extensions to the basic framework, such as to support more complex access structure and variable thresholds gates across authorities.

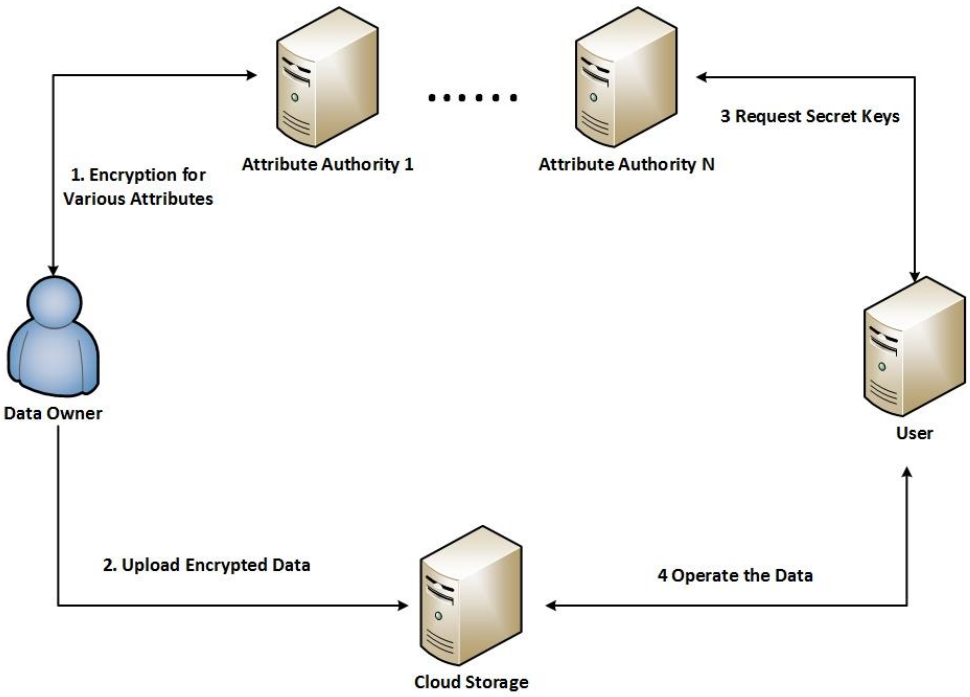

Figure 4.7Illustration of Multi-Authority Attribute-Based Encryption

The improved Chase and Chow's framework provides a more practical ABE scheme. However, the computational cost and communication overheads are high and complex. The user has to communicate with all the authorities to obtain the decryption key. In mobile Cloud environment, the processing power may not be strong enough and the mobile data network may not be reliable for completing all required communication.

Lewko and Waters [164] proposed a decentralizing ABE scheme. In their system, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. A party can act as an authority simply by creating a public key and

issuing private keys to different users that reflect their attributes. Each authority can function entirely independently and can join or leave freely. The central authority is removed in the scheme. It is secure against any collusion attacks and it can process the access policy expressed in any Boolean formula over attributes. However, their method is constructed on a mathematical model that incurs heavy computational cost.

Kan Yang and XiaohuaJia [167] designed an access control framework for multi-authority and proposed an efficient and secure MA-ABE scheme for Cloud storage. Their work is the first multi-authority scheme based on the CP-ABE scheme which does not require a central authority. However, they introduce a certificate authority which is responsible for assigning an unique identity for each user and an authority identity for each attribute authority. The certificate authority is also fully trusted.

Li *et al.* proposed a framework that achieved fine-grained, cryptographically enforced data access control model based on MA-ABE scheme for the medical records stored in the Cloud [168]. The patient uploads his encrypted PHI files to the Cloud storage using KP-ABE scheme; then the doctor or nurse requests the PHI file based on the attributes received from all the authorities. However, their proposed framework is based on a single central authority, which may be the bottleneck of the systems. Akinyele *et al*. presented a design of self-protecting electronic medical records using KP-ABE scheme on mobile devices [169]. Their system allows healthcare organizations to export medical records to locations outside of their trust boundary. Each encrypted record has its own access policy. They implemented an iPhone client application for storing and managing medical records offline.

However, their work is not suitable for the collaboration environment since it is a single authority based scheme.

In mobile Cloud environment, real-time context-related attributes can be captured to provide more complex policies to secure the data access control. These attributes can be time, current location, date, IMEI number and any other contextual attributes. Those can be derived from the local context information using the hardware embedded within smart devices. Enforcing these attributes into the encrypted data during the encryption can effectively reduce the risk of collusion attack. Several works are discussed in the following paragraphs. They have proposed approaches to incorporate such type of attributes.

Liao and Chao [170] proposed a location-based encryption scheme. They proposed a location-dependent approach to incorporate the coordinates while encrypting. The user can only decrypt the data when the coordinates acquired for GPS receiver are matched with the target coordinate. Considering the inaccuracy and inconsistency of the GPS signal, they involve a toleration distance to increase the practicality. However, privacy-preserving and data access control were not considered. A secure, selective group broadcast system in vehicular networks using dynamic attribute-based encryption is presented by Chen *et al*. in [171]. Chen *et al.* introduced a fading function to CP-ABE scheme in order to allow attribute level updates. The fading functions allow treating attributes in accordance with their rate of change so that attributes can be updated independently from the entire private keys. By choosing a proper fading rate, the balance of trade-offs between security and efficiency can be controlled to deliver a better broadcasting service or a better security services. Their work finds a solution for using context-related attributes

under a single domain. Xu and Martin proposed a dynamic user revocation and key refreshing for ABE in Cloud storage [172]. In Xu and Martin's model, the system can refresh the system keys or remove the access from a user without issuing new keys to other users or re-encrypting existing ciphertexts. Therefore, the mobile user can have a more flexible solution to protect their data and privacy in the mobile Cloud environment.

4.3.4 Challenges

In the ABE scheme, collusion attack is the most important challenge. For example, Alice is a doctor in hospital *A* and Bob is a student at City University London. Sender encrypts a message with two out of three of the following attributes: doctor of hospital A, student at City University London, and home address in London. If Bob and Alice combine their attributes then they have two out of three required attributes, but they should not be able to get the decryption keys to obtain the content of the message.

Sahai[160] addressed this issue by generating different random sharing of a secret for each user. So private keys issued for different users cannot be combined. In Chase's original MA-ABE scheme [165], GID is introduced as the identifier of each user and can be distinguished by each authority to prove the possession of attributes. In MA-ABE, the scenario is different from the single authority. Assuming that there are two authorities, $AA_1$and AA_2. Alice request keys for attribute set *S1* from $AA_1$and *S2* from AA_2. Then Bob also requests keys for attributes *S1* for AA_1 and *S2* from AA_2. If the authorities do not exchange information about the two requests and Alice and Bob are identified by nothing

beyond their attributes. So from the authorities' point of view, the two scenarios must be identical. By using the GID, the authorities can distinguish these two scenarios in order to prevent collusion attacks.

However, using the GID may cause the system to issue the decryption keys based on the identifier and not based on the attributes. So Chase still uses the central authority to issue a setup key for the user's GID. The central authority will not monitor any attributes; user only sends the GID to it and receives the key. Each authority computes his own independent secret by giving away only the GID. With the key obtained from the central authority, the user can always combine the results to obtain the secret. The secret given by each authority is a pseudorandom secret and only combined with the key from the central authority.

In Chase and Chow's improved MA-ABE scheme [166], without the adding of additional key of the central authority, each attribute authority generates a set of secret while issuing keys. When user receives these keys and reconstructs the secret, the computation will use these secrets to get the summation which is equal to 1 or -1. The innovative mathematical design successfully addressed these issues. MA-ABE is an ideal solution for data security in the mobile Cloud environment, since it combines the techniques of access control and cryptographic technologies..

4.4 Conclusion

A large amount of personal data is generated by mobile-based services; these user-generated data must be properly protected from unauthorized usage. Advances in technology and the rapid use of mobile Cloud computing systems created a necessity for protecting context sensitive information transferred through the system. The access control is not only for defining who is able to take actions on resources, but also to control the purpose for which the resources were accessed. In mobile Cloud environments, there are real-time contextual attributes which can be derived from the context information. From this aspect, the access control model for a mobile use case should be able to capture not only the static attributes of related stakeholders, but also dynamic attributes to improve the security level.

The DAC, ManAC and RBAC are designed for a traditional privacy-preserving scenario. They can still continue to contribute to suitable environments. In today's mobile Cloud environment, user's mobile devices can be considered as a node on the whole network. Those smart devices are always provided with Internet access through Wi-Fi or 3/4G data networks. They are powerful enough to execute any Web-based or Cloud-based services. Security and privacy-preserving framework is required to secure the communication between two parties and also to protect the collected personally identifiable information in the connected digital society. ABAC is a flexible, scalable, and fine-grained access control model which has shown enormous potential in the last couple of years to enhance the service provisioning from smart mobile handsets. Considering the

nature of attribute-based authorization, XACML is an ideal privacy policy language to work with ABAC. The combination of XACML and ABAC can address privacy-related issues in the mobile Cloud environment.

The MA-ABE scheme is a suitable solution for addressing issues of both security and privacy aspects in mobile Cloud environment. This chapter examined the privacy protection technologies of attribute-based encryption schemes for encryption and access control of personal data. Existing privacy protection techniques are not designed for mobile environments. The work proposed in this thesis will combine state of the art existing access control and privacy preserving techniques to provide a novel framework through which the personal data can be protected in a mobile Cloud environment.

5 User-Centric Attribute-Based Access Control Model Using XACML

The exponential growth of mobile applications let users to leave a substantial amount of online data via Cloud-based services such as Google services inthe Internet. As discussed in Chapter 3, leaving the data (personal) in online Cloud storage raises privacy issues. There is a need for user-centric approach whereby users can control the usage of their personal data in order to mitigate privacy breaches. Therefore, in today's mobile Cloud environment, privacy protections are important and technologies for protecting privacy are considered as vital features in mobile Cloud-based services. Hence, in this Chapter the author proposed a general, flexible approach to support the user-centric access control model based on XACML, where user controls the privacy of personal data in the mobile Cloud environment.

This chapter provides an implementation of a user-centric access control model using XACML. Section 5.1 describes the architecture of the model and Section 5.2 gives details of policy evaluation process. Section 5.3 gives the security evaluation of the proposed model followed by a case study. The proof of concept and implementation details are given in Section 5.4. The conclusion is given the chapter in Section 5.5.

5.1 Architecture of Policy-Based User-Centric Approach

In a mobile Cloud environment, the following contextual attributes can be exploited to secure the personal data: location, time, date, app usage etc. These attributes can be exploited to authenticate the user who requests permission for data access. In the proposed approach, authorization is a mutual process between a user and a service provider (SP). To allow a SP to access the personal data, the user authorizes the SP by pre-defined privacy access policies while the user is also challenged by the system in order to verify whether he/she is the actual user who is requesting services from the SP. An Attribute Authority (AA) which stores all the users' and SPs' attributes for authorization is introduced. The AA is a trusted party by both the users and SPs. By defining personal privacy access policies, users can control the disclosure of sensitive information which is stored online (e.g. Cloud based storage). Thus, the proposed model enables the mobile users to authorize the SP with the participation of AA. By using the available context-related attributes, the system also verifies the user to ensure it is the actual person who is requesting the services. Compared with the current identity management systems, the proposed model focuses on using context-related attributes to verify a mobile Cloud user at run-time which is more flexible and secure in mobile Cloud environment, and provides data access control for the mobile Cloud user to enforce access control on his/her online data.

In the proposed model, the users define XACML policies in order to control the access to their personal online data. XACML is a privacy access policy language which is a dialect for defining attribute-based access control policies [173]. The proposed user-centric approach has five main actors: the user, the SP, Policy Evaluation Component (PEC), the AA, and Identity Provider (IdP) as shown in Figure 5.1.

In the proposed model, AA is considered as a central authority. It maintains the attributes of the users and SPs. SP has to register with AA in order to provide a service to the mobile Cloud user. AA assigns a unique Service Provider Identity (SPId) to the SP and this identity is stored at the AA. Any attributes related to the SP is also stored at the AA. In most of the current identity management systems, an identity provider authenticates the users and SPs[174]. It is also introduced in the proposed model, and it is not only responsible for authentication, but also connects the users, SPs, and the PEC. In order to restrict access control on users' online data, PEC is introduced to enforce access control. PEC is modified based on the standard XACML structure. It extends the existing XACML standard [151]with real-time authentication for the user. By cooperating with AA, real-time attributes such as location, time are used to verify the user. Therefore, the proposed model enables users to protect their online data by personal preferences in real-time.

Details of each actor are briefly explained below:

- ***User***: A user is a person with a mobile device and the mobile device is connected to the Internet via the mobile data network or Wi-Fi.

- ***Attribute Authority***: The AA is the trusted party that stores attributes representing the SPs and the users.
- ***Policy Evaluation Component***: PEC is the core component that formats all the requests into XACML requests, evaluates the access requests and performs the access decisions.
- ***Identity Provider***: IdP is responsible for authentication of the user and forwarding the access requests to the PEC.
- ***Service Provider***: A SP provides online resources (such as Cloud services or online data) to the user and requests user's data based on the policies.

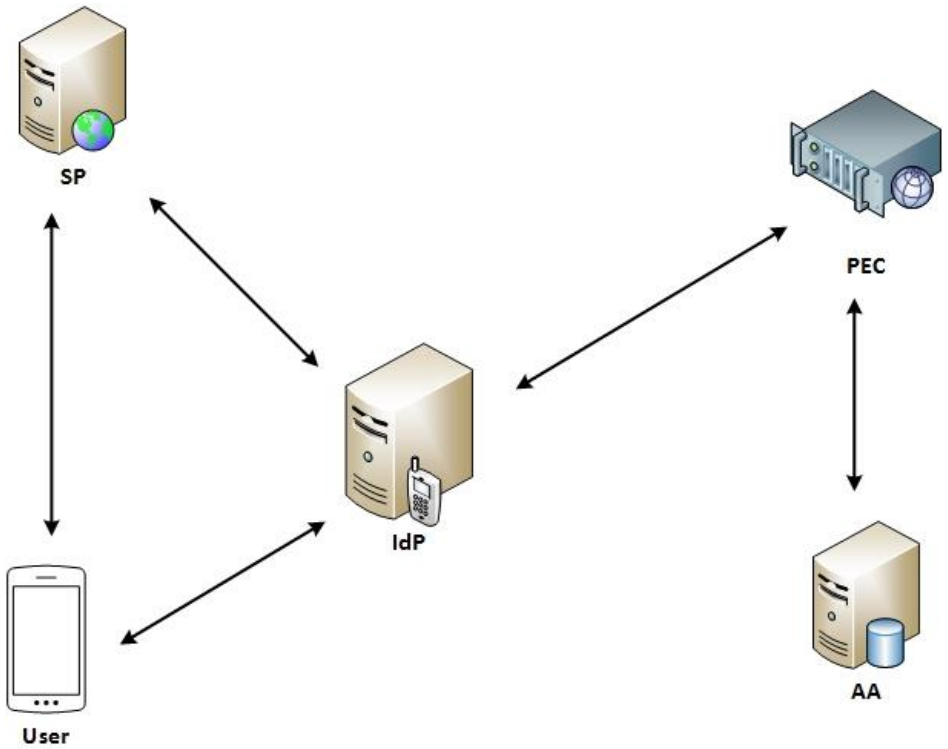

Figure 5.1 Framework of the proposed User-Centric Policy-Based Access Control Model Using XACML

The user's personally identifiable information was defined as PII in the previous chapter. In order to authorize the user, PII is used as key attributes for the user authorization process. The U.S. Office of Management and Budget suggests several examples of attributes related to the PII [175] as given below:

- *Name*: Full name, maiden name, mother's name, or alias.
- *Personal Identification number*: Social security number (SSN), passport number, driving licence number, taxpayer identification number, or financial account or credit card number.
- *Address Information*: Street address, home address, IP address, or email address.
- *Personal Characteristics*: Photographic image (especially face or other identifying characteristic), fingerprints, handwriting, or other biometric data.
- *Information about an individual that is linked or linkable to one of above*: date of birth, place of birth, race, religion, weight, geographical indicators, employment information, medical information, education information, and financial information.

In some countries, people are assigned with a unique number which can be used for identity verification. In the UK, driving licence number and National Insurance (NI) number can be considered as unique numbers. As studied in Chapter 4, the Global Identity (GID) is considered as a key attribute in the proposed model

which stands as an identity of a user. The user is registered using the GID and other attributes. Assuming that the GID is a secret and only the user knows the exact value of the GID.

The IdP and the PEC have an existing trust relationship. The user is assumed to request data and services from the SP. The IdP can also be a federated identity provider whereby the user can use the same identity to access any other partner sites. The user needs to register with IdP before requesting online resource from the SP. PEC is an online access control service, which enable users to define access control policies for their online data. PEC can be considered as a private Cloud that implements access control services on behalf of the user. AA is trusted by the user, SPs, and PEC.

5.1.1 System Initialization

In the proposed access control model, there are several steps that needs to be executed by both the user and the SPs in order to initialize the system.

Initially, a user needs to finish the registration process. The registration process can be done online. The registration process includes the following steps:

(1) Register with the IdP

(2) Register with the AA

(3) Define privacy access policies for his/her attributes at the PEC

After the registration, the user's details such as username and password will be stored in the IdP, the data which is related to users' privacy is stored at the AA. For

instance, the NI number and driving licence number are stored at the AA. With the pre-defined XACML policy, the PEC can evaluate any access request on user's PII.

For a SP, the registration process is explained below:

(1) Register with the AA and assign a SPId

(2) Register with the IdP

After the registration, the SP's attributes such as SPId, name, address etc. are stored at the AA. SPId is considered as an identifier of the SP.

5.1.2 Design of the Model

The main goal of proposed model is to enable mobile Cloud usersto define access control polices on their online PII. Thus, data requesters who satisfy the access policy can be granted access on these sensitive personal online data. Assuming all the actors of the proposed model and any services or data provided by the system must be available online. The workflow of the proposed model is depicted by the stages described below:

Stage one: for a registered user, the first step is to login into the system via IdP. After receiving the request, IdP starts the authentication process. If the user is successfully authenticated, he can then request any data or services from a SP. Once received the resource request, the SP would ask several attributes of the user in order to deliver the resource. In a mobile environment, the user should have full control over his online personal information. Thus, in order to protect user's privacy, an access control mechanism which restricts the access on user's personal

data is proposed. The SP must satisfy the privacy access policy defined by the user in order to gain the PII data. The policies stored at the PEC are used to control the access on user's data and the policy evaluation is carried out by the PEC.

Stage two: after requesting the data or services, the IdP generates a local security context based on the request. The security context contains information of the user, SP and the requested resources. The security context will be forwarded to the PEC by IdP.

Stage three: after receiving the security context, PEC extracts information about the user, SP and the requested resource from the security context. PEC executes the following three steps:

> *Step 1*: Retrieve the information of the SP, query the AA for the relevant attributes of the SP in order to check the authenticity of the SP.
>
> *Step 2*: Retrieve the information of the user, query the AA for the relevant attributes of the user to check the authenticity of the user.
>
> *Step 3*: Evaluate the request based on the policies stored in the Policy Repository.

In order to verify the user in Step two in Stage three, the PEC needs to generate a challenge message and send it to the user. User's mobile device generates a response message based on user's inputs. Then PEC starts the policy evaluation if the verifications of the SP and user are successful. If the result of policy evaluation in Step 3 is successful, then PEC sends the result to the user and SP via IdP.

From the three stages, the proposed model support SSO which is an important feature in today's identity management systems. The use of real-time attributes

enables the proposed model to dynamically authenticate the user, which cannot be provided by the current identity management system such as OpenID, Liberty.

Figure 5.2 depicts the workflow of the proposed model. The message flow is described below.

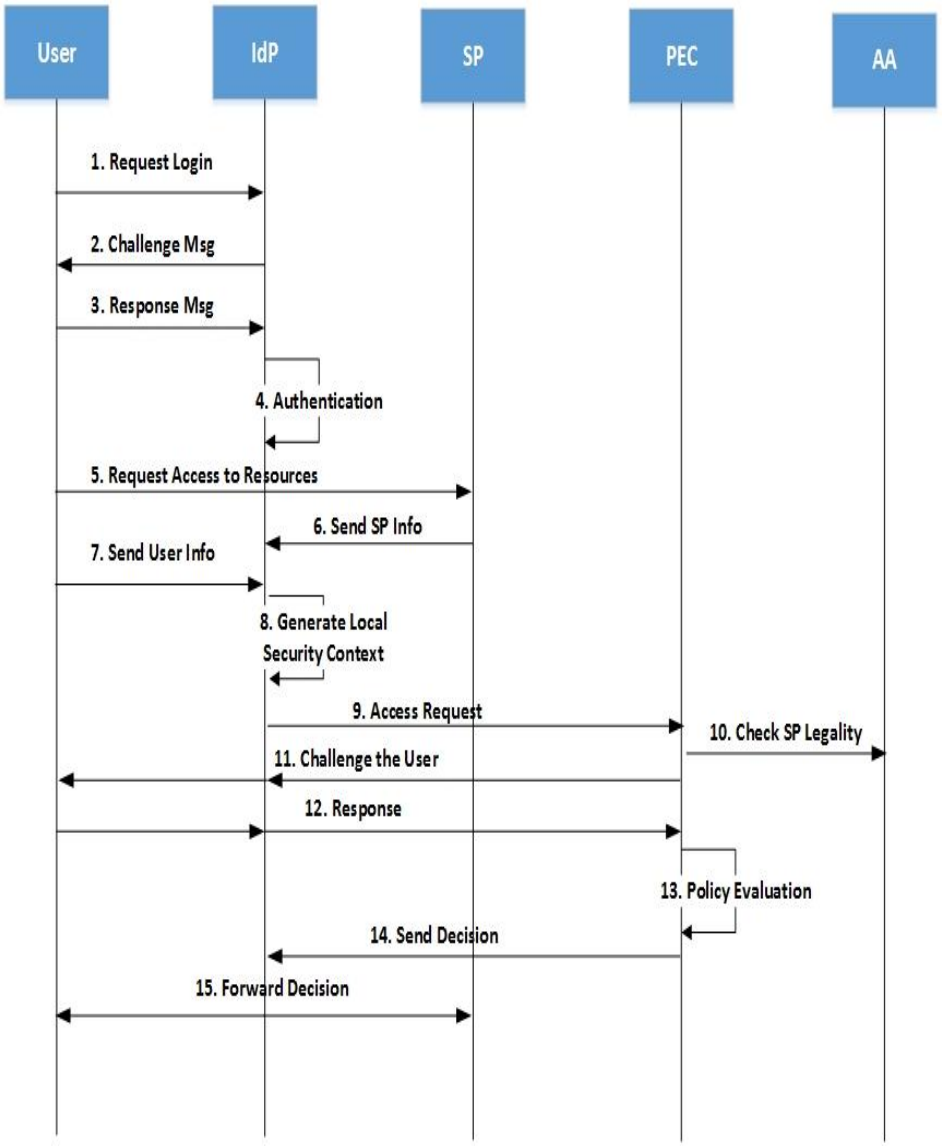

Figure 5.2Illustration of the Message Flow of Proposed User-Centric Policy-Based Access Control Model

(1)The user sends login requests to the IdP using the mobile device.

(2)The IdP sends challenge message to the user for authentication.

(3)After the user inputs username and password, the mobile device generates a response message.

(4)The IdP authenticates the user based on the response message.

(5)The user sends a request to SP to access data or services.

(6)The SP forwards the attribute of itself and the attributes of the requested resource to IdP

(7)User sends his attributes to the IdP

(8)The IdP generates a local security context based on the request. The context contains necessary attributes of the SP, the user and the requested resource.

(9)The IdP forwards the request to the PEC for further policy evaluations.

(10)The PEC receives the request and extracts any required information from the request. Then, the PEC queries the AA to check the validity of the SP.

(11)The PEC queries the AA for the GID of the user and generates a challenge request to the user based on the GID.

(12)After receiving the challenge, the user has to answer the challenge. The mobile device generates a response message which contains the user's answers and sends the response back to PEC.

(13) If the response is verified, then the PEC starts to evaluate the request.

(14) If the request is approved, the requestor will be granted permission. The decision will be sent to the IdP.

(15) The IdP forwards the decision to both user and service provider.

From Figure 5.2, it shows that once the mobile user requested the data or services from SPs, IdP starts to collect all the necessary information from the user and SP. The information from the user contains username, or location, time and any other dynamic real-time attributes. Meanwhile, the information also contains details of the SPs and the information about the requested resource. Such information consists of related attributes and the IdP generates a local security context based on these attributes. Thus, the security context consists of three parts:

(1) Attributes of the user

(2) Attributes of the SP

(3) Attributes of the requested resources

The PEC extracts necessary information from the security context and queries the AA for any information about SP. If the response from the AA shows that the SP is a legitimate provider, then the PEC checks whether the user has defined any access control policy for his personal data.

Based on the requested resource, the SP asks any PII attributes of the user in order to deliver the resource. During this step, the PEC challenges the user in order

to authenticate the user's identity. Policy evaluation protects user's privacy. Such operations that are carried out at the PEC are the necessary steps for policy evaluation process of the system. In the next section, more details about the architecture of PEC and the policy evaluation process will be presented.

5.2 Policy Evaluation Component (PEC)

Policy Evaluation is the core process of a privacy policy-based access control system. In our model, PEC is the key component of the architecture. It is responsible for XACML authentication and authorization. After receiving the local security context, PEC starts to the access evaluation for the access requests.

The PEC consists of the following main functional components:

CH: Context Handler, formats received access requests into XACML format

PEP: Policy Enforcement Point, issues access requests to the PDP and enforces access decisions

VP: Validating Point, verifies the SP

PDP: Policy Decision Point, evaluates the applicable policy and renders an authorization decision

RG: Request Generator, generates challenge requests to verify the user

PR: Policy Repository

Each component performs different functionalities and processes. Let us recall the local security context generated by IdP in the previous section. Once the PEC received the security context (see step 8 in Figure 5.2), the PEP extracts the attributes of the SP and user. Then PEC queries the AA for the attributes of the mobile user and SP. Figure 5.3shows the architecture of the PEC.

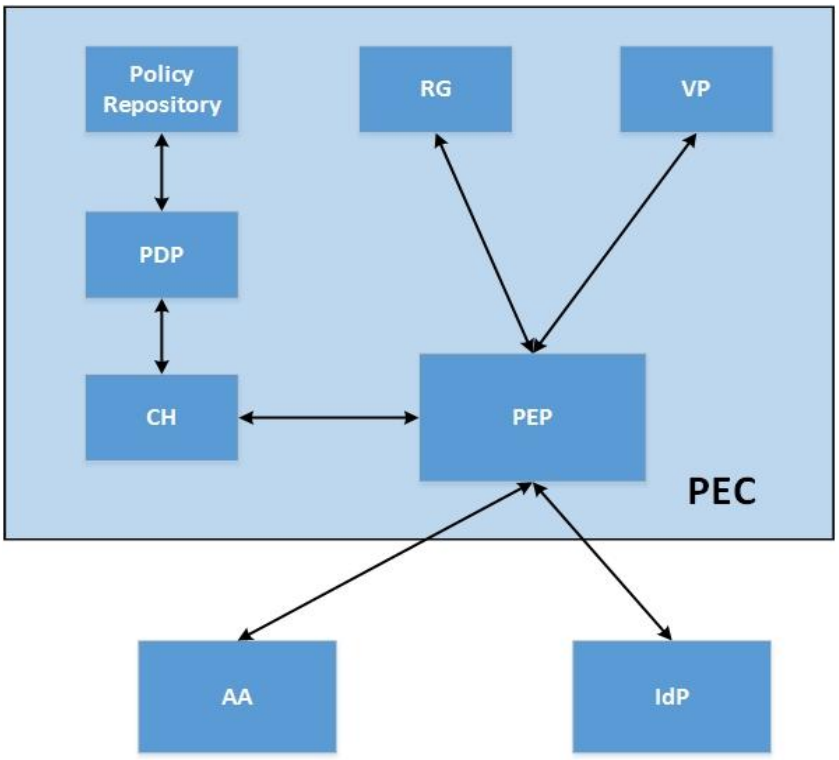

Figure 5.3Architecture ofthe PEC

Figure 5.4 depicts the authorization process of the model. For instance, if a mobile user wants to purchase an item from the SP, then the user needs to request the mobile banking services to complete the payment. Generally, the user is concerned about the authenticity of the SP. At this time, the user needs to check whether it is the real SP with whom he/she wants to communicate. The VP compares the attributes of the SP obtained from AA and the attributes extracted from the security context received from IdP. If the SP is verified, RG will generate a challenge message. PEP forwards the challenge message to the user via IdP. User's mobile device displays this challenge request and waiting for the inputs from the user. This step is to ensure that it is the actual account holder who is requesting the resource. A response message will be generated using the input and sent to the PEP. PEP forwards the response to VP. If VP successfully validates the user, CH will generate a XACML access request and send it to the PDP for policy evaluation. PDP makes the access decision based on the policies in the policy repository. If the access decision is "Deny", then user's personal data will not be revealed to the SP.

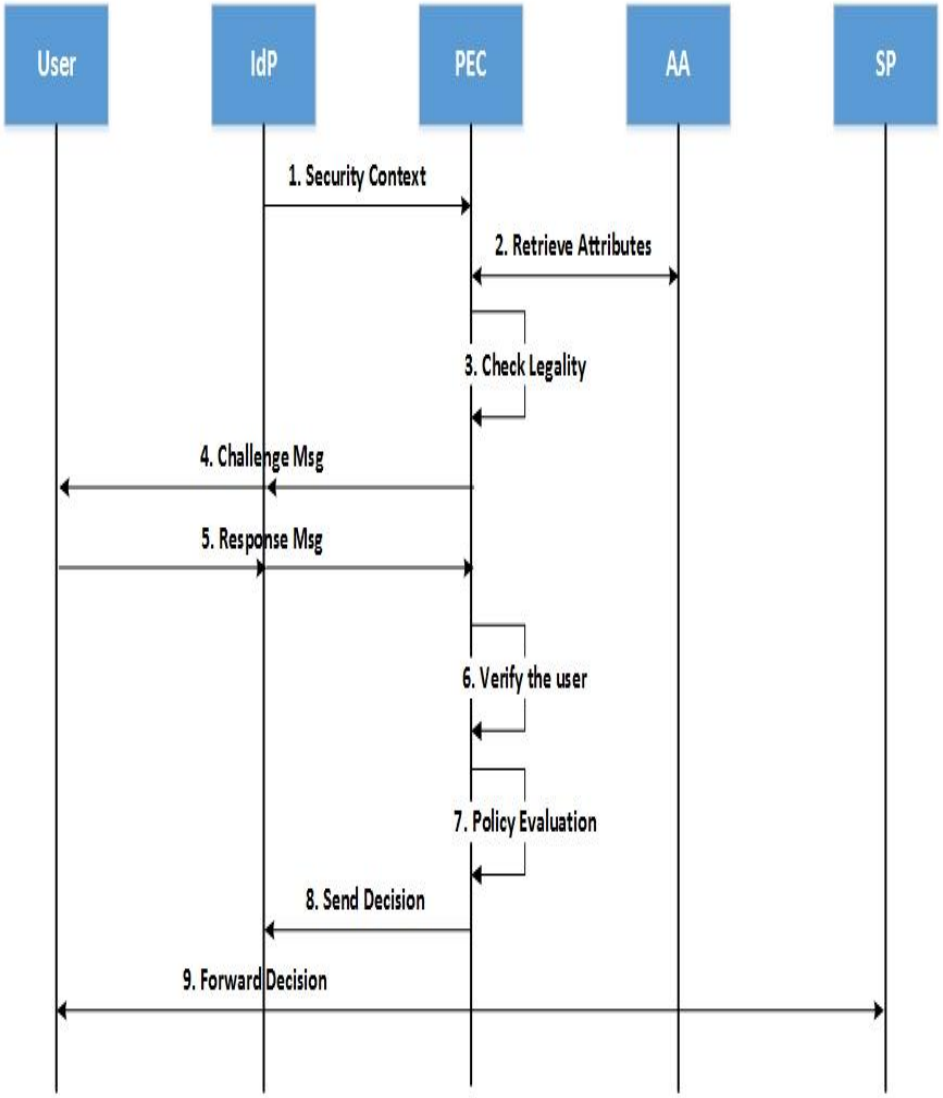

Figure 5.4Illustration of Work Flow of Policy Evaluation

The workflow of PEC is briefly depicted below:

(1) IdP generates the security context and sends it the PEC

(2) PEC retrieves attributes from AA

(3) PEC starts to check whether the SP is a legitimate service provider or not

(4) A challenge message is generated by PEC in order to verify the user, the challenge is forwarded by the IdP

(5) User's mobile device generates a response message based on user's input and sends it to the PEC via IdP

(6) PEC starts to verify the user based on the response message

(7) If the verification is successful, PEC then start the policy evaluation process

(8) PEC sends the access decision to IdP

(9) IdP forwards the decision to both the user and the SP

This approach enables user to define privacy access policies to protect their data. In order to deliver data and services, SP requires user's PII. Current identity management systems cannot provide flexible access control mechanisms to help a mobile user restrict access on his/her PII. The proposed model is designed and built on top of the standard XACML framework. Users can simply define privacy access polices to control access from different requests on his online PII. The thesis contributes the access control model for mobile Cloud users by modifying the standard XACML framework. The access control evaluation is given in next subsection.

5.3 Security Evaluation

5.3.1 Protocols on Authentication

5.3.1.1 Protocols on Authentication

Based on the message flow presented in Figure 5.2 and Figure 5.4, the protocols on authentication can be separated into the following stages.

- User (smart devices) authenticates to the IdP

  (1) The user will request access to the SP from the IdP.

  (2) IdP challenges the user with username and password.

  (3) IdP verifies the user and redirect the user to the SP

  (4) The user log into the SP

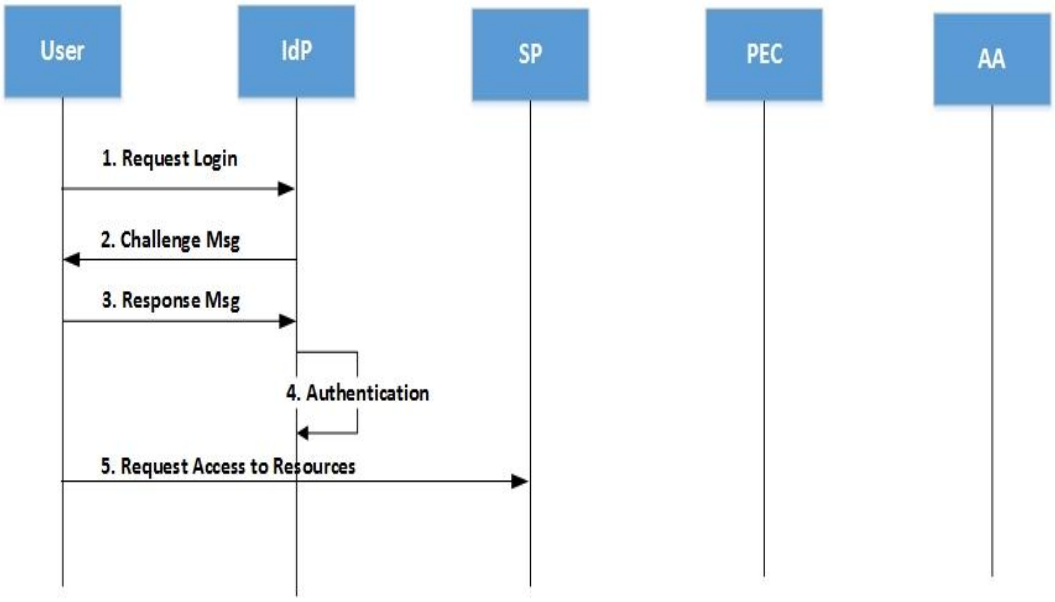

Figure 5.5Authentication between User and SP

- SP authenticates to the PEC

  (1) SP sends attributes to IdP

  (2) IdP generates local security context

(3) IdP sends local security context to PEC

(4) PEC verifies SP

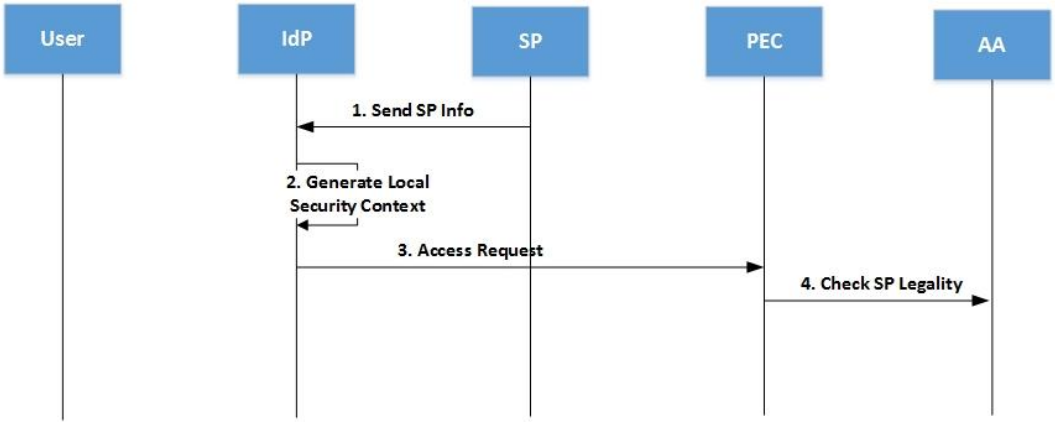

Figure 5.6Authenticaion for SP

- User (smart devices) authenticates the PEC

(1) User sends attributes to IdP

(2) IdP generates local security context

(3) IdP sends local security context to PEC

(4) PEC challenges user using GID

(5) User responses to the challenge

(6) PEC verifies the user

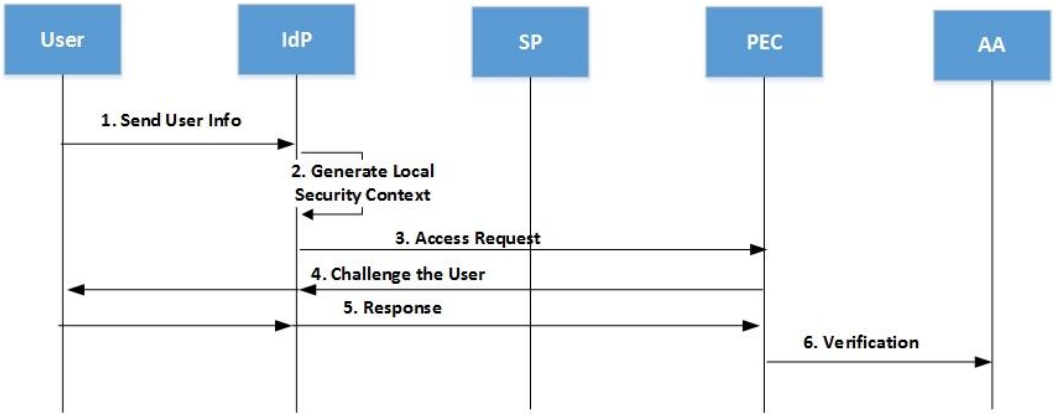

Figure 5.7Authentication for User

5.3.1.2 Security Parameters

- Local Security Context

  Local Security Context is a set of attributes from user, SP and the required resources. The PEC validates user and SP based these attributes. It is defined in XML format.

```
<SecurityContext>
    <user>
        <username>John</username>
        <time>201206121230</time>
        <organization>City University London</organization>
    </user>
    <SP>
        <SPId>200010000001</SPId>
        <SPName>Citi Bank</SPName>
        <time>201206121235</time>
    </SP>
</SecurityContext>
```

5.3.2 Security Analysis

The current identity management systems and existing technologies provide powerful protection to the data confidentiality and integrity. The proposed model uses existing cryptographic technologies to secure data communications between each active role. The main goal of the proposed model, is to utilize the existing

dynamic attributes in a mobile Cloud environment to protect users' online data based on pre-defined privacy access policies.

5.3.2.1 Access Evaluation

Let consider the following simple scenario: A customer who lives in the UK is travelling to Paris. During the travel, he wants to use his mobile device to access the online service from a SP in Paris and at the same time would like to reject all requests for online transactions from the UK based ISPs during this period. The security analyse is given in the following subsections.

Our proposed scheme is secure from spoofing attacks. Use the above scenario as a typical example and assume the user has defined an access policy that any access requests from Paris can be approved. The flow is based on Figure 5.2.

User: Login to the IdP in Paris.

Adversary: Registered at the IdP as a service provider. The user can request data or services from the adversary. Adversary want to collect users' online data without permission.

(1) User request a service from the adversary

(2) IdP collects attributes from the user and adversary

(3) IdP collects access request on user's data from adversary

(4) IdP generates local security context and sends to PEC

(5) PEC extracts attributes from the local security context

(6) PEC queries AA to get legitimate attributes for AA

(7) PEC verifies SP based on the attributes received from AA

(8) PEC verifies the user based on the GID

(9) The access request is evaluated by PEC

(10) PEC sends the results to IdP

(11) IdP processes the results

Initially the user is logged in to the IdP. IdP collects the necessary attributes from both the user and SP. Attributes from user are location, time, username, OS, IMEI etc. Attributes from the SP are SPId, name, time, IP address, location etc. IdP generates the local security context, which is described below:

```
<SecurityContext>
    <user>
        <username>user</username>
        <time>2013007291600</time>
        <location>Paris</location>
        <OS>iOS 7.0</OS>
    </user>
    <SP>
        <SPId>200010000001</SPId>
        <SPName>Thomas Cook</SPName>
        <time>2013007291600</time>
        <location>London</location>
        <IP>0.0.0.0</IP>
    </SP>
    <acessrequest>
        <subject>Thomas Cook</subject>
```

```
            <object>
                <name>John</name>
                <gender>Male</gender>
                <...>
                ......
                </...>
            </object>
        </accessrequest>
    </SecurityContext>
```

PEC extracts necessary attributes from the local security context. In order to verify the user and SP, it queries the AA for correct attributes to evaluate access request. Recall that the user has defined access policy on his personal online data. The adversary has to pass the following two steps.

- PEC verifies SPId based on the AA's records

  Adversary's SPId which is sent via the local security context needs to match the record from the AA.

- PEC evaluate the access request based on the user's pre-define policies.

  If the SP passes the first step, PEC evaluates the access request. In this scenario, the request is: "Thomas Cook requests John's online PII in UK'. The local security context proves user's current location is Pairs, thus the access policy is active. Based on the polices, the access decision is 'FALSE'.

The adversary can gain access to user's online data unless 1) it is a legitimate SP which has records in AA, and 2) its access requests satisfy user's privacy access

policy. Therefore, the proposed model provides extra protection on user's online data.

Following the verification of the AA, PEC verifies the user based on the GID. The challenge message is displayed on the smart devices and asking user to input the GID. The consideration for this step is unauthorized use of the smart device. The 'unauthorized use of the smart device' suggests that someone is trying to use the smart device without the user's permission. For instance, user lost the device in Pairs, someone got hold of the phone and ordered products on user's behalf. GID is sensitive information and only known to the user.

5.3.2.2 Security on Communications

Communication security is mainly focusing on the message confidentiality and integrity in the proposed mobile environment, which is evaluated with respect to the encryption or signature algorithm and the cryptographic keys.

Roles in the proposed model such as SP, IdP, PEC and AA can be considered as online services providers. The communications between these entities are normally carried out using the public key infrastructure. Each entity has a private key and public key pair. The sender use its own private key to sign message and use the receiver's public key to encrypt the message. Thus, the receiver uses sender's public key to verify sender digital signature and decrypts the message using its own private key.

With the development of mobile platforms, smart devices can provide more secure and flexible ways to secure communication. Android and iOS supports mobile apps using the SSL to secure the communications [176]. Hu *et al.* proposed a 3 factor

(PIN code, SIM card authentication and facial biometric) authentication method to protect the message security on Android platform [177].In the proposed model, the mobile user uses these existing technologies to protect message communications, however, provides more data access restrictions on his online data usage.

5.3.3 User-Centric Approach

In the proposed model, the IdP authenticates the users. If the user is successfully authenticated by the IdP, then he can request data or services from the SPs. A registered user trusts the AA and pre-defines the data access control policy for his PII data. The user initiates the access request for targeted resources from a SP; the IdP generates the local security context based on the request. The context consists of user's information (e.g. username), and SP's information (e.g. SPId).

In order to deliver the requested resources to the user, the SP requests information about the user. In an e-health service use case, the SP needs to know the historical medical information of the user in order to provide medical advice. This raises a security issue: which part of information can be revealed to the e-health system? Most of the existing systems lack mechanisms to protect the user's PII. In default, SPs collect any information or allow a partner site to collect the required information. Hence, user has no control over the data once it leaves his premises. In our proposed model, more sensitive attributes are stored at the AA. AA has the ability to control the attributes disclosure. XACML privacy access policies can be defined by the user at PEC. Thus the user can control the disclosure of his PII from unauthorized access.

On receiving the security context from the IdP, the PEC extracts the information from the security context and then the PEC queries the AA to check the legality of the SP. Since the SP is registered at the AA with required information, the AA can validate the SP. This step ensures that the SP is a registered and legitimate provider. If the SP is verified, then the PEC queries the AA for user's information. In this scenario, the GID is considered as the identifier of the user. The following two phases are carried out to protect the user's privacy:

- SP acquires user's attributes to satisfy its own policy in order to deliver resources to the user.
- PEC receives the request from IdP, evaluates user's pre-defined policies to decide whether to disclose the requested data.

These two phases are crucial for the mutual authorization and solve the potential authorization negotiation between the user and SP. The pre-defined policies have the higher priority than the policy defined for the requested resources. If there is a conflict that the SP is requiring sensitive information in which the pre-defined policies are forbidden, then the PEC rejects the SP's request to protect user's privacy.

5.3.4 Use Case Study

Many hospitals today are providing mobile services to the patients. The e-health service is a typical user case for mobile Cloud users. For example, a patient wants to choose a doctor in dental department. In order to serve the patient, the hospital requires his personal information. He firstly registers with the hospital system as a patient and uploads the PII and related medical records to the Cloud Storage, which

can be managed by the hospital system administrators. And then he defines an access policy that only a doctor in dental department can read his PII and the medical records.

In this scenario, the patient requesting the services from the hospital can be considered as the user in the proposed model. The Cloud storage is the AA who manages attributes for both the users and doctors. There is a central server which enforces access control and is responsible for the verification of user and SP. The central server can be considered as the PEC.

After logging into the system, the patient requests for a dental service. A doctor responds to the request and chooses to treat the patient. The identity provider of the system receives the requests and collects attributes from the patient and the doctor. A local security context is created and sent to the central server. The central server extracts the attributes from the local security contexts. Based on these attributes, it queries the AA in order to verify the doctor and the patient. The verification method can be defined by the practical requirements. For instance, Doctor can be verified by the staff ID, job title, and department. The patient can be challenge not only by the GID, but also through security questions defined by himself. Once the verifications are successful, PEC evaluates the access request from the doctor. Neither a doctor from a department other than the dental department, nor a nurse from the dental department can gain access to user's PII and medical records. This information is available to a doctor from the dental department.

Thus, the patient restricts access on his online sensitive data. Privacy and data security are protected by unauthorized access. The policy-based access control

provides a simple method to the patient if he wants to disclose personal data to a specified entity.

5.4 Proof of Concept

5.4.1 Protocol Verification

This section provides formal protocol verification and validation for the proposed model. Scyther tool [178-180] was developed by CasCremers for security protocol verification. It is a tool for the automatic verification and falsification of security protocols. Scyther provides a graphic user interface which incorporates the Scyther command line tool and python scripting interface.

The syntax of Scyther tool will not be discussed in the thesis, but some basic knowledge is explained in order to explain the protocol validation process.

send_1(X, Y, data1): role X sends data1 to role Y

recv_1(X, Y, data1): role Y receives data1 from role X

claim_x(X, Secret, data1):role X claims that the event of 'sending data1 to role Y' is secret.

The five actors in the proposed model is defined as role in the Scyther validation tool. The message flow sequence in Figure 5.2 is followed and implemented in the Scyther tool as follows:

```
protocolProtocolValidation(User, SP, IdP, PEC, AA)
{
    role User {
        constuname: Data;
```

```
        freshattrU: Data;
        var challenge: Nonce;
        //user uses SSL to send username
        send_1(User, IdP, {{uname}pk(IdP)}sk(User));
        //user sends real-time attributes
        send_2(User, IdP, {{attrU}pk(IdP)}sk(User));
        //receives challenge from PEC
        recv_7(PEC, User, {{challenge}pk(User)}sk(PEC));
        claim_user(User, Secret, uname);
        claim_user(User, Secret, attrU);
        claim_user(User, Secret, challenge);
    }
    role SP {
        freshattrSP: Data;
        //SP sends attributes to IdP
        send_3(SP, IdP,   {{attrSP}pk(IdP)}sk(SP));
        claim_sp(SP, Secret, attrSP);
    }
    roleIdP {
        varuname: Data;
```

```
        varattrU, attrSP: Data;
        freshLSConxt: Data;
        //IdP sends local security context to PEC
        send_4(IdP, PEC, {{LSConxt}pk(PEC)}sk(IdP));
        //receive username from user
        recv_1(User, IdP, {{uname}pk(IdP)}sk(User));
        //receive attributes from user
        recv_2(User, IdP, {{attrU}pk(IdP)}sk(User));
        //receive attributes from SP
        recv_3(SP, IdP,   {{attrSP}pk(IdP)}sk(SP));
        claim_idp(IdP, Secret, uname);
        claim_idp(IdP, Secret, attrU);
        claim_idp(IdP, Secret, attrSP);
        claim_idp(IdP, Secret, LSConxt);
    }
    role PEC {
        varLSConxt, rslt: Data;
        freshqury, challenge: Nonce;
        //PEC quiries AA for legitimate information
        send_5(PEC, AA, {{qury}pk(AA)}sk(PEC));
```

```
        //PEC send challenge to user
        send_7(PEC, User, {{challenge}pk(User)}sk(PEC));
        //PEC receives loncal security from PEC
        recv_4(IdP, PEC, {{LSConxt}pk(PEC)}sk(IdP));
        //PEC receives quiry result from AA
        recv_6(AA, PEC, {{rslt}pk(PEC)}sk(AA));
        claim_pec(PEC, Secret, LSConxt);
        claim_pec(PEC, Secret, qury);
        claim_pec(PEC, Secret, rslt);
        claim_pec(PEC, Secret, challenge);
    }
    role AA {
        varqury: Nonce;
        freshrslt: Data;
        //send quiry result to AA
        send_6(AA, PEC, {{rslt}pk(PEC)}sk(AA));
        //receive quiries from PEC
        recv_5(PEC, AA, {{qury}pk(AA)}sk(PEC));
        claim_aa(AA, Secret, qury);
        claim_aa(AA, Secret, rslt);
```

```
        }
    }
```

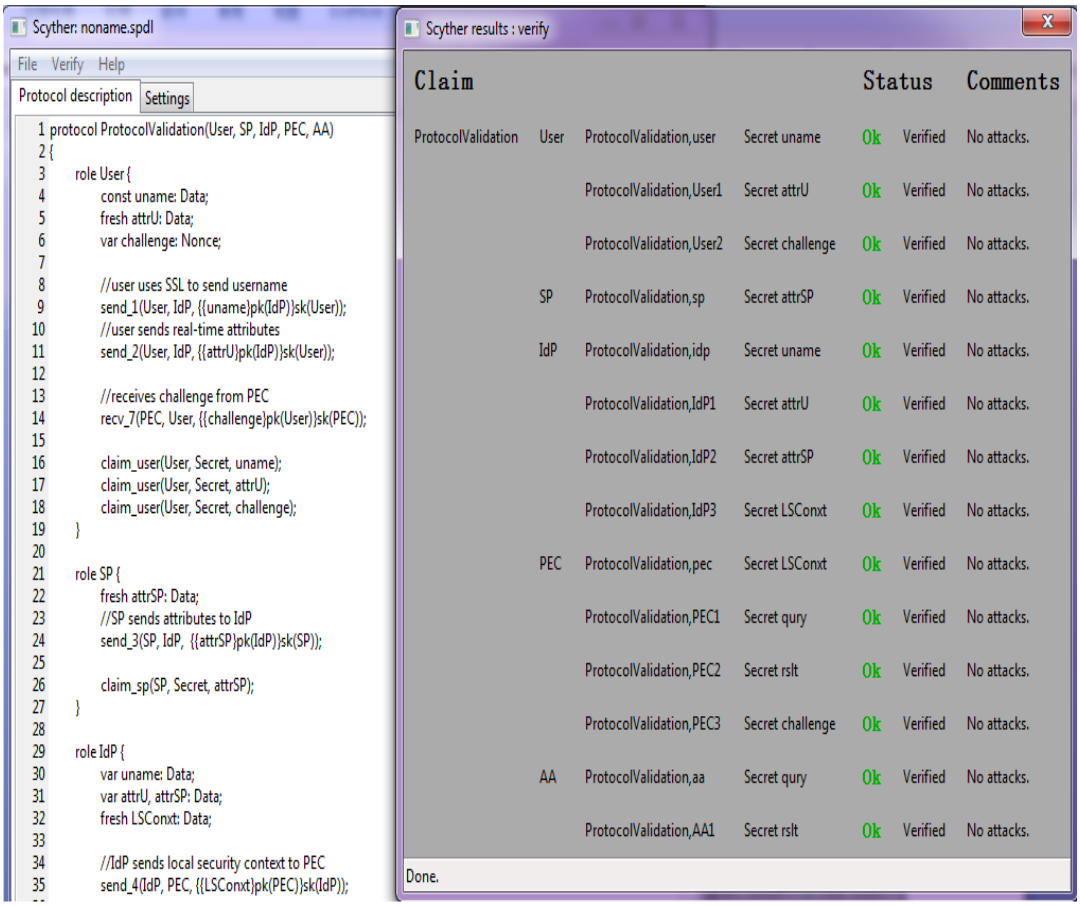

Figure 5.8 Validation of the Proposed Model by Scyther Toll

The result of Scyther has proved the claims of implemented protocol. During the whole message flow, all communications is claims as secret. No potential attacks are found. It can been seen from Figure 5.8 that all secret parameters from each actor in the proposed model is secure.

5.4.2 Implementation and Tests

The proposed model is implemented in a mobile web service platform as a prototype. The development is based on the open source software development platform[15, 181-184]. The functionalities of four entities; IdP, SP, mobile user, and the AA were developed separately as shown in Figure 5.5. The communications between each entity uses SOAP messages.

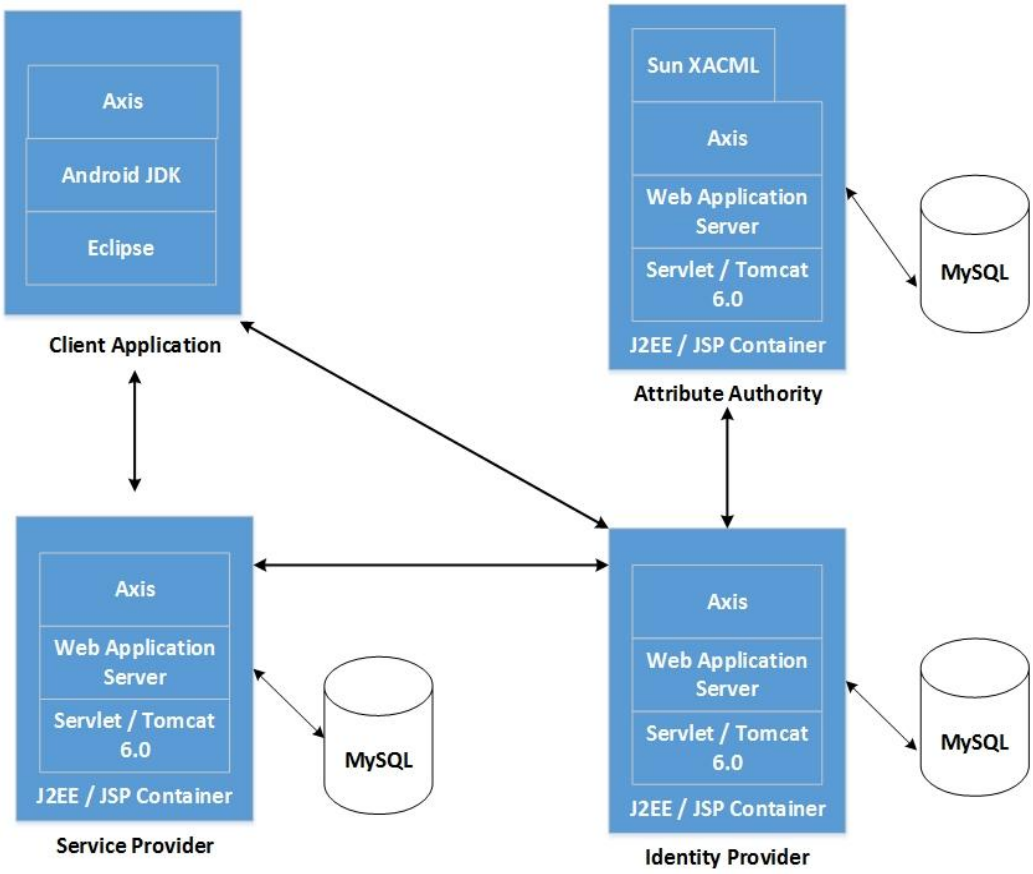

Figure 5.9Proof of Concept

Client app for the mobile device is developed using Eclipse and the implementation is based on Android JDK using version 2.3.3. The AA's and SP's

functionalities are developed and deployed on JAX-WS web service using Apache Tomcat Web Server [183] in Netbeans [185]. The IdP is also built on top of the Tomcat server. The administration functionalities are implemented using JSP containers. The J2EE and JSP containers connect with MySQL database for storing and retrieving of administrative information such as identity information that are stored at the IdP's database. The AA also stores all the attributes in the database. The XACML engine is developed based on Sun XACML 2.0 implementation [182].

The architecture is feasible to integrate more than one SP into the systems and SPs can be implemented using different software development approaches. The implementation of access control system is based on the XACML architecture. The following section will discuss the message standard of XACML language.

5.4.3 Sample Screenshots of the Client Application

The screenshots of Android emulator are provided in Figures5.6-5.9. These figures depict several key steps such as user login (Figure 5.6), the SP's request for the user's PII (Figure 5.7), policy evaluation process (Figure 5.8), and policy decision (Figure 5.9) of the proposed user-centric attribute-based access control model. The author implemented the XACML policy evaluation process as an online back-end service. In Netbeans, the policy evaluation can be tested as a stand-alone service under Glassfish server. The Figure 5.10 depicts the screenshot of the policy evaluation verification page.

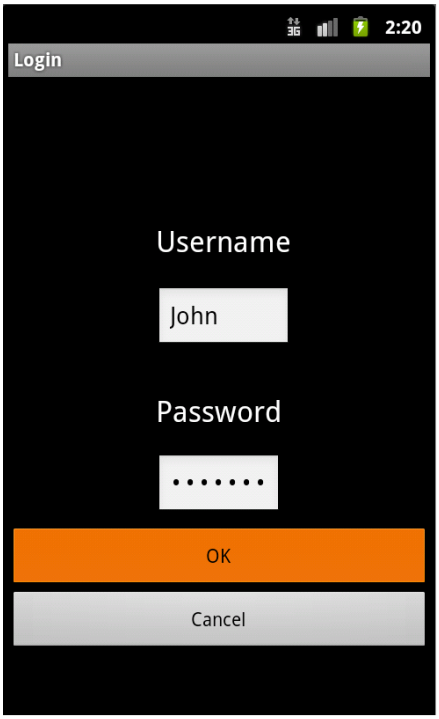

Figure 5.10Login Page of the Client Application

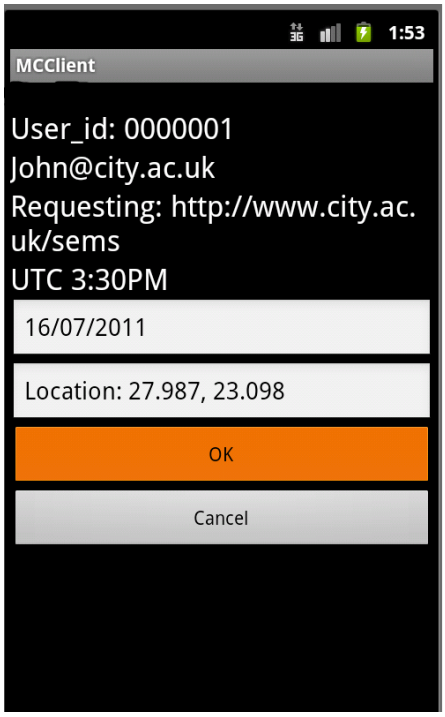

Figure 5.11Requesting Page of the Client Application

Figure 5.12Information Gathering at PEC

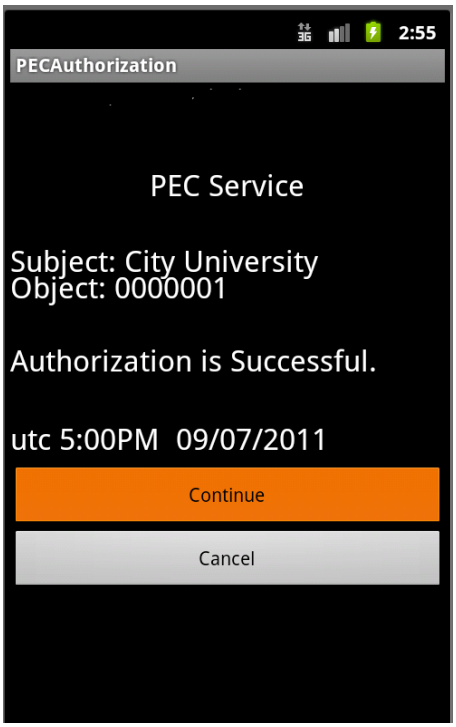

Figure 5.13Authorization Result of PEC Service

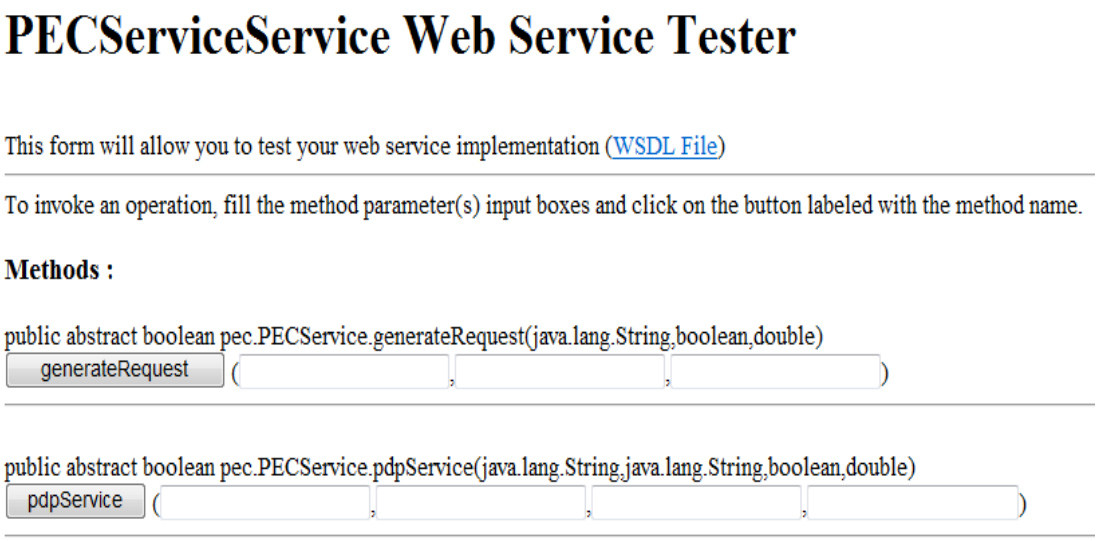

Figure 5.14Service Test Page

Figure 5.14 depicts the test page of the PEC service. In Figure 5.14, the following two different methods are given: generateRequest() for generating XACML request and pdpService() for policy evaluation. The first method is a service for PEC to generate a dynamic XACML request for the SP on user's PII data. The second method is a service of PEC to evaluate the PII access request.Let us make the pdpService() as an example. In order to test the pdpService(), the following four parameters need to be entered: user_id, user's email address, the authentication state of the user, and the current location.

Figure 5.15 shown below is a screenshot of SOAP request and SOAP response when a data requester is successfully granted access on user's PII. The SOAP request part in Figure 5.15 shows the access request, which includes the four required attributes from the data requester: user_id, email address, state of

authentication, and location data. The SOAP request is received by the PEP and formatted by CH. PDP evaluates the formatted request based on the pre-defined policies.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header/>
    <S:Body>
        <ns2:PDPService xmlns:ns2="http://PEC/">
            <user_id>0000001</user_id>
            <attribute_1>John@city.ac.uk</attribute_1>
            <isAuthenticated>true</isAuthenticated>
            <LBSAttribute>0.0345</LBSAttribute>
        </ns2:PDPService>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:PDPServiceResponse xmlns:ns2="http://PEC/">
            <return>true</return>
        </ns2:PDPServiceResponse>
    </S:Body>
</S:Envelope>
```

Figure 5.15PEC Service Test

The SOAP response part in Figure 5.15 shows the decision of the access request. The decision is made by PDP and CH converts it to original format and sends it back to PEP. Thus, PEP can forward the decision to the user and SP.

It shows that the XACML policy evaluation process is successfully loaded and works fine.

5.4.4 Possible Extension

The implementation was currently worked under the scenario for one SP. However, based on the study of the existing identity management system in Section 3, our system can accommodate more than one SPs. User's identity can be considered as a federated identity. Therefore, our system can be extended to accommodate a list of SPs as long as they have done the registration process described in Section 5.1.1.

5.4.5 XACML Message Standard

This section describes polices used for the data access control, and construction of XACML request/response messages during the transmission.

5.4.5.1 XACML Policy

At the root tag of XACML policies is a Policy or a PolicySet. A PolicySet is a container that can hold other Policies or PolicySets, as well as references to policies found in remote locations. A Policy represents a single access control policy, express through a set of Rules. Each XACML policy document contains exactly one Policy or PolicySet root XML tag. Therefore, a Policy or PolicySet may contain multiple policies or Rules, each of which may have different access decisions. XACML defines a collection of Combining Algorithms which reconcile the decisions from each decision that is evaluated by policies or rules.

A policy example which defines that any user who holds an email address from the domain of "city.ac.uk" can get access to read any resources at "http://www.city.ac.uk" is presented in the section Appendix

5.4.5.2 XACML Request Context

In a XACML policy-based system, every access request has to be formatted using XACML's specification. A XACML request contains the necessary information to make authorization decisions. Basically it has the subject's (requestor) attributes, the resources' attributes and the operation that the subject wants to perform on the resource.

Regarding to the above policy example, a user who holds an email account "Alice@city.ac.uk" requests the resources of School of Engineering and Mathematical Sciences at http://www.city.ac.uk. This XACML request is given in Appendix.

5.4.5.3 XACML Response Context

As a result of evaluating the policy, the response context contains only the decision and the requested resource. Regarding the policy example and response context example, the user has an email account "Alice@city.ac.uk", hence she can obtain the data consent on the resources at "http://www.city.ac.uk/". The policy evaluation decision should be *Permit*. The response context is listed in Appendix.

5.5 Discussion

Attributes based access control scheme, which makes access decision based on the attributes of requestors, resources, and environment can provide greater flexibility in today's environment. The model is constructed based on XACML. XACML model use attributes as the key factor to authorize data consent. With standardised specifications and components, it can flexibly solve privacy related issues for

mobile devices. In XACML based model, each access request, the complete XACML hierarchy must be processed in order to find the matching policy sets, policies and rules, to evaluate them ,and to compute access decision result[186].In reality, authorization policies comprise of thousands of users and millions of resources. A XACML based model requires service to establish all the required components to complete a XACML request/response, which reduces the flexibility to build a privacy preserving model for mobile devices. The efficiency of the whole system may be decreased by such process. The computing and memory consumption for evaluating policies are high cost tasks. XACML message is transmitted in XML format and the parsing overhead of XML documents is also an issue [187] that needs attention. As a result, a more light-weight and general solution is required to satisfy today's security and privacy issues of mobile devices.

5.6 Conclusion

The proposed user-centric policy-based access control model uses the attributes from any related stakeholders in the mobile environment. By involving the trusted attribute authority, user has the ability to check the legality of the requested service providers and define his/her own access privacy policies to protect the sensitive personal information. Thus, the user has the control of his online data to decide who can be granted access to the personal information. The access can also be restricted to a certain part of the personal information. By using specific attributes that are provided by the mobile devices, services providers and the related environment, the proposed user-centric access control model establishes a secure mobile Web service environment. In addition to this, it helps to grow the customer confidence and also provides seamless access to their mobile financial services from anywhere, at any time at a touch of a button.

6 Context-Aware Attribute-Based Encryption Schemes

6.1 Introduction

This chapter gives a design of context-aware attribute-based encryption (ABE) scheme for smart mobile devices. Due to the recent technological advancements in mobile devices, powerful processors and various sensors are embedded within smartphones. This trend allows millions of people to use smartphones for their work and day-to-day social activities. In order to support this transformation, the traditional Cloud computing infrastructures are being modified into mobile Cloud computing.

Chapter 5 presented a model for mobile Cloud users to enforce access control over their PII to prevent the privacy attack from unauthorized service providers. However, in Cloud infrastructures, platforms and applications are provided by the third-party Cloud service providers. Hence, the Cloud service providers have control over the data that is stored in the Cloud storage and can monitor the communications between the end user and the Cloud with or without their prior consent. The Cloud computing organisations should invest heavily in risk assessment to ensure system security and compliance.

Furthermore, the user experience and usability are important factors that can influence a mobile application. Users prefer using applications with good performance and user interface. Traditional security mechanisms such as multiple passwords, several memorable security questions, security patterns etc. should be

avoided in mobile Cloud environment as they are complex and infeasible. The smartphone features should be exploited to provide robust access control with less user operations. Therefore, the contextual attributes such as location, time, app usage, unlock failure etc. in mobile Cloud environment should be involved in the security framework.

This chapter presents a context-aware attribute-based encryption scheme for smart mobile devices. It extends the functionality of the proposed model in Chapter 5 for Cloud data storage with data confidentiality and addresses some shortcomings in usability and user experience. The real-time contextual attributes are further investigated via profile-behaviour profiling techniques. The framework is designed to address the following issues:

- The confidentiality and anonymity for data storage in mobile Cloud environment.
- Development of lightweight cryptographic technology, hence the data owner encrypts, uploads, and decrypts the data using mobile device with low computational cost.
- The data owner has the control of his personal data, and can define flexible privacy access policies using context information (e.g. location, time, date).
- The proposed framework reduces the communication overheads compared to the conventional schemes since it is designed for mobile networks and utilizes the existing Cloud infrastructure.

- Reduce the existing trust relationship between each stakeholder to reduce the security risk.

The following three algorithms are proposed in the following sections: (1) context-aware ABE scheme based on single authority, (2)context-aware ABE scheme based on multiple authorities, and (3) low-complexity multi-authority ABE scheme which is designed to reduce the computation and communication overheads of mobile devices.

The available context-related attributes of mobile Cloud environment are used in the proposed algorithms in order to strengthen the data security. The context-related attributes can be determined by behaviour-profiling techniques[93, 188-190]. Mobile user's activities such as app usage, network usage, charging times and unlock failures have been used to profile the user-behaviour. This user-behaviour profile can be used to detect anomaly activities and provide more fine-grained level of security.

Let us assume that there is an app installed in mobile-device which can be used to detect anomalous activities (let us call this app as "behaviour-profiling" app). Installing the behaviour-profiling app in the user's mobile device can be used to capture the real-time attributes. Such attributes can be used to verify whether the current user is the phisical owner of the mobile device. In the following three algorithms, assuming that user's mobile device has the profiling-behaviour app installed and the context-related attributes can be captured by the app. There is a mapping function *M*, embedded within the behaviour-profiling app. *M* outputs "*yes*"

if the particular attributes satisfied the requirements. The outputs will be used by the following three algorithms for the user to recover the encrypted data.

6.2 Context-Aware Single Authority Attribute-Based Encryption Scheme

In this section, a context-aware attribute-based encryption scheme with single authority is described. It is designed for mobile users to share and store their personal data securely in the Cloud storage. A user has his personal data such as location records, medical history stored in the Cloud. He would not allow any unauthorized requesters to access this sensitive information, even the Cloud service provider. This can only be achieved by storing the data in the encrypted form in the Cloud storage. In the meantime, access control techniques should also be deployed to restrict access to the data. Attribute-based encryption techniques are considered as the ideal solution together with the access control models for authentication of the users. A data owner can upload encrypted data in the Cloud, and the encrypted data was enforced with access control polices during the encryption process. Access policies are defined by the data owner, thus, he determines who can access the data. A user will request data from an Attribute Authority, which is responsible for maintaining the attributes, verifying the user authenticity, and also issuing the decryption keys.

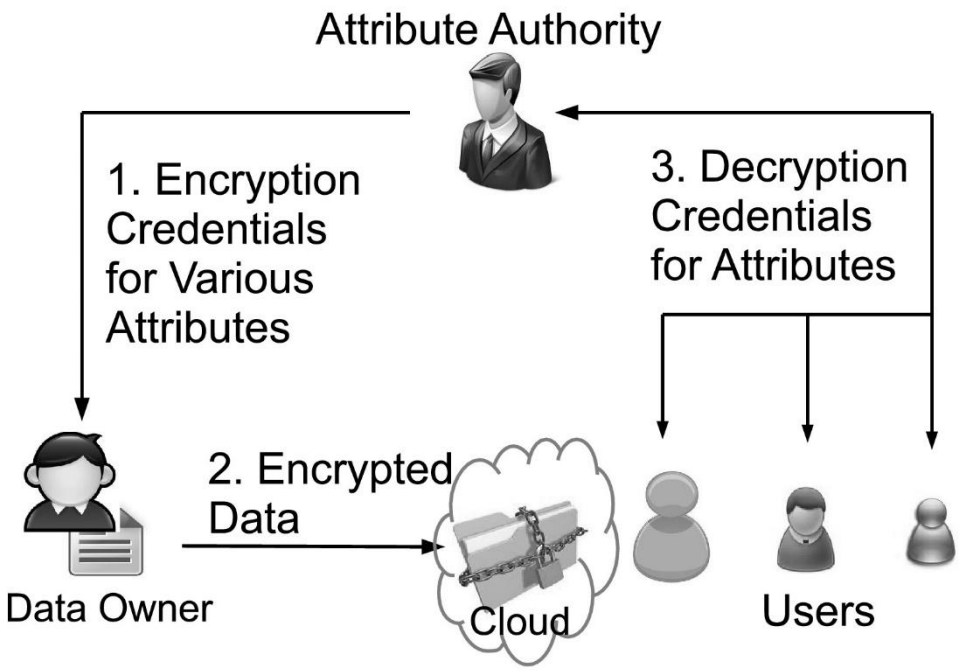

Figure 6.1The Framework of Single Authority ABE scheme.

Figure 6.1 shows the architecture of single authority ABE scheme. Four actors are involved in the scheme, the data owner, the AA, the user and Cloud Storage. A data owner uses the contextual attributes and static attributes from AA to encrypted his data and uploads it to the Cloud storage. A user downloads the data and requests decryption keys based on his attributes. The AA verifies the user and issues the decryption keys to the user.

6.2.1 Preliminaries

This section gives the background information and security assumption for the ABE scheme. Figure 6.1 shows the architecture of the single authority ABE scheme.

6.2.1.1 Bilinear Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$be two cyclic groups of order q for some large prime q and g is a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$between these two groups. The map must satisfy the following properties:

- *Bilinear*: A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$is bilinear if $e(g^a, g^b) = e(g, g)^{ab}$ for all P, $Q \in \mathbb{G}_1$and all $a, b \in \mathbb{Z}$.
- *Non-degenerate*: The map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$to the identity in $\mathbb{G}_2$. Observe that since $\mathbb{G}_1, \mathbb{G}_2$are groups of prime order, this implies that $e(g, g) \neq 1$.
- *Computable*: There is an efficient algorithm to compute $e(P, Q)$ for any P, $Q \in \mathbb{G}_1$.

6.2.1.2 Complexity Assumption: Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Suppose a challenger chooses $a, b, c, z \in \mathbb{Z}q$ at random, the DBDH assumption is that no polynomial-time adversary is to be able to distinguish the tuple ($A = g^a$, $B = g^b$, $C = g^c$,$Z = e(g, g)^{abc}$) from the tuple ($A = g^a$, $B = g^b$, $C = g^c$,$Z = e(g, g)^z$) with more than a negligible advantage.

6.2.1.3 Secret Sharing Scheme

In a secret sharing scenario, a data D is divided into n pieces. In order to reconstruct D, a threshold value k where $1 < k \leq n$ is defined. Only if there are k pieces shares

or more comes together, then user can reconstruct the secret *D*. Complete knowledge of $(k-1)$ pieces reveals no information about *D*. The scheme was firstly invented by Adi Shamir in 1979 [161].

6.2.2 Construction

The section will begin by explain how the traditional ABE scheme can be converted into context-aware ABE scheme. In a single authority ABE scheme, there is only one attribute authority (AA) maintains a set of attributes and corresponding encryption and decryption credentials. In contrast to the conventional ABE scheme, a data owner in the proposed scheme can encrypt the data by not only using credentials obtained from AAs but also using context-related attributes based on the data owner's preferences.

The proposed algorithm is composed of four sub algorithms namely Setup, Key Issuing, Encryption and Decryption. Functionalities of each algorithm are briefly explained in the following:

- *Setup*: The setup algorithm takes a security parameter as input and outputs a bilinear group and a set of security parameters.
- *Key Issuing*: AA generates decryption key for a user *u* that holds a set of attributes.
- *Encryption*: The encryption algorithm takes a set of attributes maintained by AAs as well as a set of context-related attributes

defined by data owner and data as inputs and outputs the ciphertext of the data.

- *Decryption*: The decryption algorithm inputs the decryption credentials received from AAs and context-related parameters obtained from the smart mobile device and the ciphertext. Then it outputs the original data.

The descriptions of the four sub algorithms are detailed below.

Setup $\mathcal{S}$

- For a given security parameter $\lambda$and $\sigma \in \{0,1\}s^{poly(\lambda)}$, group bilinear parameters are generated by the AA: $q, g, \mathbb{G}_1 \leftarrow S(1^{\lambda}; \sigma)$.
- AA randomly picks a secret y, $t_{A,i} \in \mathbb{Z}q$ for each attribute $i(1 \leq i \leq n)$ and publishes corresponding public keys $\{T_i = g^{t_{A,i}}\} \forall i$ and $Y = e(g, g)^y$.

Key Issuing $\mathcal{KI}$

- To issue decryption keys to user u, the AA chooses a random polynomial p_u with degree $(d - 1)$ where $p_u(0) = y$.
- The AA generates decryption credential for i^{th} attribute for the user u as: $D_u i = g^{p_u(i)/t_{A,i}}$ where $\forall i \in A_u$(A_u denotes the attributes set of the user u).

Encryption $\mathcal{E}$

The data owner encrypts the message m for a set of attributes $A_m = A_A \cup A_C$, where $A_A = \{a_{a,1}, \ldots, a_{a,N}\}$ denotes the attributes maintained by AA and $A_C = \{a_{c,1}, \ldots, a_{c,n}\}$ denotes the context-related attributes defined by the data owner, as follows:

- The data owner randomly chooses s_A , $s_B \in \mathbb{Z}q$, and encrypts the message as $E_{nC_m} = mY^{s_B}$.
- The data owner computes $E_0 = h(M(a_{a,1}) \| M(a_{a,2}) \| \ldots M(a_{a,n}))Y^{s_A+s_B}$, $E_i = g^{t_i s_A} \forall i \in A_A$, where $h : \{0, 1\} \longrightarrow \mathbb{Z}q$ is a secure function, M is a mapping function of context related attributes and $\|$ denotes concatenation.
- Now the data owner uploads $CT_m = \{E_{nC_m}, E_0, E_i, \forall i \in A_A$, and $A_C\}$ into the Cloud.

Decryption $\mathcal{D}$

- User downloads CT_m from the Cloud and checks the required attributes to obtain the message m.
- User computes $e(E_i, D_i) = e(g, g)^{p_u(i)s_A}$ for any attribute $i \in A_C \cap A_u$.
- Using interpolation technique, user can compute $Y^{s_A} = e(g, g)^{p(0)s_A} = e(g, g)^{ys_A}$.

Now the behaviour-profiling application installed in user's mobile device computes the hash value of context related attributes such as location, risk-level associated with current location and risk-level associated with user behaviour and outputs $h(M(a'_{a,1}) \| M(a'_{a,2}) \| \ldots M(a'_{a,n}))$.

User can decrypt the data as follows:

$$E_{nC_m} \cdot \frac{h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))\mathrm{Y}^{\mathrm{S_A}}}{E_0}$$

$$= \quad mY^{S_B} \cdot \frac{h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))\mathrm{Y}^{\mathrm{S_A}}}{h(M(aa,1) \parallel M(aa,2) \parallel \cdots M(aa,n))\, Y^{s_A+s_B}}$$

If and only if:

$h(M(a_{a,1}) \| M(a_{a,2}) \| \ldots M(a_{a,n})) = h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))$, then the user can get the message *m*.

The novelty in our scheme compared to the conventional ABE scheme lies in encryption and decryption sub algorithms which are detailed in the following: denote the context related attribute set defined by the data owner as A_C= $\{a_{C,1}, \ldots, a_{C,n}\}$, where $a_{C,I}$ denotes context related attributes. For sake of simplicity, consider the following three context related attributes: $a_{C,1}$ = "location", $a_{C,2}$ = "risk level with his recent app usage" and $a_{C,3}$ = "unlock failures in last two days". Now the data owner defines $\mathrm{A_C}$ as follows:

$\mathrm{A_C}$ = {$a_{C,1}$ = "London", $a_{C,2}$< "3" and $a_{C,3}$< "2"}, assume that risk level varies between 1 to 10 where higher risk denoted by larger value. Then computes

$E_0 = h(London \parallel yes \parallel yes)Y^{S_A+S_B}$.

This is the hash value of the required context-related information and is considered as the access policy of the encrypted message. Using the client

application installed in the mobile device, the user will capture the real-time context related attributes such as, location. The mapping function, *M*, outputs "*yes*" if the current risk level is less than threshold defined by employer. This ensures that the employee has all the credentials from AA, context-related attributes enforced by the employer also need to be satisfied in order to decrypt the message.

The security and performance analysis will be discussed in the Section6.3.3.

6.3 Context-Aware Multi-Authority Attribute-Based Encryption Scheme

In this section, the author describes the context-aware MA-ABE scheme. In a single authority scenario, there is only one AA will monitor all the attributes and issues encryption and decryption credentials for the users. It is a fully trusted party which users have to prove his identity in order to obtain a decryption credential. For instance, to decrypt a message *m*, a user proves his identity with required set of attributes and receives the decryption credentials based on those attributes. In this case, the AA has too much power which provides itself with the ability to decrypt all communication messages and knows all the user's attributes. In the event of corruption and/or compromise, the message's confidentiality cannot be achieved and user's privacy can be revealed to the attackers. This is one of the drawbacks in single authority based ABE scheme.

Meanwhile, it is more convenient to monitor and maintain different sets of attributes by different authorities in real world. As mentioned in Section 3.6.3, NHS manages UK national health information system and the UK drivers and vehicles'

information is maintained by the Department of Vehicle Licensing Agency (DVLA). Hence, in reality, it is impossible for a single authority to monitor all these attributes and hence a viable solution is to go for multiple authorities based ABE scheme (MA-ABE). Several MA-ABE schemes without considering context information were proposed in [164-166].

In the context-aware MA-ABE scheme, each AA manages different sets of attributes and issues credentials for the employees based on a set of attributes. The AA defines a threshold value d_k so that the message can be decrypted only if a user has at least d_k number of given attributes from the AA. There is no trusted central authority exists for issuing keys and verifying identities.

6.3.1 Preliminaries

6.3.1.1 Bilinear Pairings and Complexity Assumption

The bilinear pairings and complexity assumptions are the same in section 6.2.1.

6.3.1.2 Two-phase Commit Protocol

Two-phase commit protocol (2PC protocol) is a distributed algorithm that coordinates all the processes that participate in a distributed transaction on whether to commit or abort the transaction [191, 192]. The protocol is a standard protocol for making commit and abort atomic. In a normal execution of a single distributed transaction, the protocol consists of two phases:

(1) *The Commit-request Phase*: In this phase, a coordinator attempts to prepare all the required participating process for transactions. Define

necessary steps for either committing or aborting the transaction for every participant.

(2) *The Commit Phase*: The coordinator decides whether to commit or abort the transaction based on the response from the participants (commit or abort), and notifies the results to all the participants. Then the participants take actions regards to the result.

6.3.1.3 Anonymous Key Issuing Protocol

In our MA-ABE scheme, the central authority which holds the master secret was removed. An anonymous key issuing protocol under the structure of Chase and Chow's scheme [166] is presented.

In MA-ABE system, public parameters are available for users and AAs. A user can request secret keys from an AA provided he has some attributes which are managed by the AA. Then key generation algorithm will be executed by the AA and the corresponding secret keys will be returned to the user. When a data owner wants to encrypt a message, he makes use of the public parameters together with an attribute set of his choice to carry out the encryption. Any user can obtain the decryption keys from the corresponding keys to the attributes.

Define GID as the global identifier of the user. *u* is the GID of the user; it can be a hash value of the GID. The user uses the anonymous key issuing protocol to obtain decryption keys without the central authority. Before starting the key issuing process, the k^{th} AA has to generate several parameters.

Using the key issuing algorithm presented, the k^{th} AA shares a secret $s_{k,j}$, with j^{th} AA and picks $x_k \in \mathbb{Z}_q$, and computes $y_k = g_1^{x_k}$. Thus, the k^{th} AA and j^{th} AA

can jointly compute the shared secret $g_1^{x_k x_j/(s_{k,j}+u)}$. The k^{th} AA also has private keys: α, β, γ ∈ $\mathbb{Z}_q$. Figure 6.2 depicts the flow of the anonymous key issuing protocol. Each step in the protocol is described below:

(1) A two-phase commit protocol is used to start the key issuing and takes u, ρ_1 from the user and $s_{k,j}$ from the k^{th} AA as input, and outputs $X = (s_{kj} + u)\rho_1$.

(2) The k^{th} AA picks a random $\tau \in \mathbb{Z}_q$, computes $X_1 = g^{\tau/x}$ and $X_2 = h^{\alpha\tau}$.

(3) The k^{th} AA sends X_1, X_2 and proof of knowledge of the secret values (i.e. α, τ, x).

(4) The user picks a random $\rho_2 \in \mathbb{Z}_q$ and computes $Y = \left(X_1^{\rho_1} X_2\right)^{\rho_2}$. Then the user sends *Y* and proof of knowledge of ρ_2 to the k^{th} AA.

(5) The k^{th} AA computes $Z = Y^{\gamma/\tau}$ and forwards *Z* and proof of knowledge of γ and τ to user.

(6) The user computes $D = Z^{1/\rho_2}$. Therefore, user gets the key *D* as a corresponding decryption key.

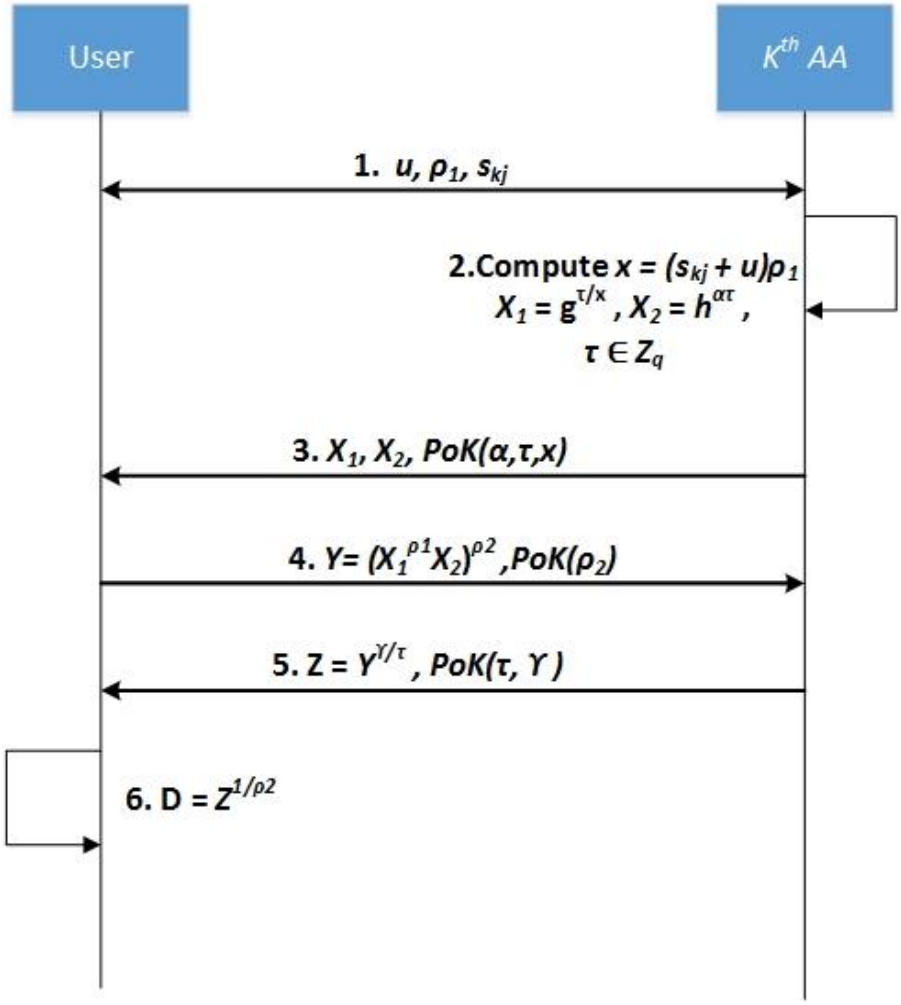

Figure 6.2Anonymous Key Issuing Protocol

6.3.2 Construction

Similar to single authority case, the proposed algorithm is also composed of four sub algorithms namely as Setup, Key Issuing, Encryption and Decryption. In the following functionalities of each algorithm are briefly explained.

- *Setup*: The setup algorithm takes a security parameter as input and outputs a bilinear group and a set of security parameters. Let

us denote the total number of AAs as *K* and number of attributes maintained by k^{th} AA as n_k.

- *Key Issuing*: Each AA generates decryption keys for a user *u* that holds a set of attributes using the anonymous key issuing protocol defined in Section 5.3.1.3.
- *Encryption*: The encryption algorithm takes a set of attributes maintained by AAs as well as a set of context-related attributes defined by data owner and data as inputs. Then it outputs the ciphertext.
- *Decryption*: The decryption algorithm inputs the decryption credentials received from each AA and context-related parameters obtained from smart mobile device and the ciphertext. The output will be the original data.

Let us describe the context-aware MA-ABE scheme below:

Setup $\mathcal{S}$

For a given security parameter λ and $\sigma \in \{0,1\}^{poly(\lambda)}$, group bilinear parameters are generated by the AAs: $q,\ g_1 g_2,\ \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \leftarrow S(1^{\lambda};\ \sigma)$.

Now, the AAs interact with each other and execute the following:

- k^{th} AA randomly chooses $v_k \in_R \mathbb{Z}_q$ and computes $Y_k = e(g_1,\ g_2)^{v_k}$, and sends Y_k to other AAs, where each AA computes $Y = \prod Y_k = e(g_1,\ g_2)^{\sum_k v_k}$.

- Each pair of AAs shares a secret, k^{th}AA and j^{th} AA randomly choose $s_{kj} \in \mathbb{Z}_q$, such that $s_{kj} = s_{jk}$.
- k^{th} AA randomly chooses $x_k \in \mathbb{Z}_q$ and computes $y_k = g_1^{x_k}$. Using the share secret s_{kj} and u, k^{th} AA and j^{th} AA computes $y_k^{x_j/(s_{kj}+u)}$ and $y_j^{x_k/(s_{jk}+u)}$ respectively.

Then each AA carries out the following steps individually:

- For i^{th} attribute stored in the k^{th} AA, where $i \in \{1, \dots, n_k\}$, and $k \in \{1, \dots, K\}$ the k^{th} AA randomly chooses a secret $t^{k,i} \in \mathbb{Z}_q$ and computes corresponding public key as $T_{k,i} = g_2^{t_{k,i}}$.

Key Issuing $\mathcal{KI}$

- User u executes the following steps with each k^{th} AA:
- For $j \in \{1, \dots, K\}/k$, user gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{kj} \in \mathbb{Z}_q$ is a random value.

After obtaining all D_{kj}, user computes

- $D_u = \prod_{(k,j)\in\{1,\dots,K\}\times\{1,\dots,K\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j)\in\{1,\dots,K\}\times\{1,\dots,K\}/\{k\}} R_{kj}$.

- If user u satisfies d_k number of attributes, then k^{th} AA randomly picks a d_k-degree polynomial $p_{k,u}$ with $p_{k,u}(0) = v_k - \sum_{j \in \{1,\dots,K\}/\{k\}} R_{kj}$.
- k^{th} AA computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \dots, n_k]$ for each eligible attribute *i* for the user.

Encryption $\mathcal{E}$

The data owner encrypts the data *m* for attribute set $A_m = A_A^1 \cup A_A^2 \cup \dots A_A^K \cup A_C$ as follows:

- The data owner randomly picks $s_A, s_B \in \mathbb{Z}_q$ and encrypts the data as follows: $E_{nC_m} = mY^{s_B}$.

The data owner then computes

- $E_0 = h(M(a_{a,1}) \| M(a_{a,2}) \| \dots M(a_{a,n}))Y^{s_A+s_B}$, $E_1 = g_2^{s_A}, \{C_{k,i} = T_{k,i}^{s_A}\}$, $i \in A_A^k, k \in [1, \dots, K]$.
- Now the data owner uploads the encrypted data $CT_m = \{E_{nC_m}, E_0, E_1, C_{k,i} i \in A_A and A_C\}$ into the Cloud.

Decryption $\mathcal{D}$

The user downloads the CT_m from the Cloud and checks the required attributes to decrypt the data.

For each k^{th} AA:

- Using $S_{k,i}$ and the corresponding $C_{k,i}$, user computes $e(S_{k,i}, C_{k,i}) = e(g_1, g_2)^{s_A p_{k,u}(i)}$.
- User interpolates all $e(g_1, g_2)^{s_A p_{k,u}(i)}$ and gets $P_{k,u} = e(g_1, g_2)^{s_A p_{k,u}(0)} = e(g_1, g_2)^{s_A (v_k - \sum_{j \in \{1,\dots,N\}/\{k\}} R_{kj})}$.
- User multiplies all $P_{k,u}$ together, gets

$$Q = e(g_1, g_2)^{s_A (\sum v_k - R_u)} = \frac{Y^{s_A}}{e\left(g_1^{R_u}, g_2^{s_A}\right)}.$$

Now the client application installed in user's mobile device computes the hash value of context-related attributes such as current location, risk-level associated with current location and risk-level associated with user behaviour and outputs $h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))$.

The user can decrypt the data as follows.

$$E_{nC_m} \cdot \frac{h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))\mathrm{e}(\mathrm{D_u}, \mathrm{E_1})\mathrm{Q}}{E_0}$$

$$= mY^{S_B} \cdot \frac{h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))\mathrm{Y^{S_A}}}{h(M(aa,1) \parallel M(aa,2) \parallel \cdots M(aa,n))\, Y^{s_A+s_B}}$$

If and only if:

$h(M(a_{a,1}) \| M(a_{a,2}) \| \dots M(a_{a,n})) = h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))$, then the user can get the message *m*.

6.3.3 Security Analysis

In this section, the security of the proposed framework is analyzed. The data confidentiality which is the most important feature of MA-ABE scheme is firstly analysed. Then the security analyse of key issuing protocol is presented to prove the anonymity key issuing protocol.

6.3.3.1 Confidentiality

The proposed scheme is to be proved secure in the selective ID (SID) model. In SID, the adversary must provide the identity he wishes to attack before receiving the parameters of the system. It is analysed following by Sahai's method in [193].

Assume there are *N* number of attribute authorities. Let us denote $n_k$to be the number of attributes monitored by each AA. Consider the following scenario:

Setup

- The adversary sends a list of attribute sets $A_C = A_C^1 \ldots . A_C^K$, one set for each AA. He also provides a list of corrupted AAs.
- The challenger generates parameters for the system and sends them to the adversary.

Key Issuing

- The adversary receives all the parameters which includes the system parameters, honest AA' public keys, public and secret keys of the corrupt AAs.
- The adversary makes secret key queries as he wants the AAs follow two rules: 1) for each GID, there must be at least one honest AA

from which the adversary has fewer than d_k number of attributes, and 2) the adversary never queries the same authority twice with the same GID.

Challenge

- The adversary sends two messages M_0 and M_1.
- The challenger chooses a bit b, computes the encryption of M_b for attribute set A_C and forwards this encryption to the adversary

Guess

- The adversary outputs a guess b' that $M_{b'}$ has been encrypted.
- The adversary is said to succeed if he can correctly identify the encrypted message if $b = b'$.

In our scheme, the challenge encryption $b = \frac{E_n C_m}{Y^{s_B}} = \frac{E_n C_m}{e(g_1, g_2)^{s_B \sum_k v_k}} =$

$$e(g_1,\ g_2)^{s_B \sum_k v_k} \cdot \frac{h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))\mathrm{e}(\mathrm{D_u}, \mathrm{E_1})\mathrm{Q}}{E_0}$$

From the algorithms in the previous section, the following three values are known.

- $\mathrm{e}(D_u, E_1) = \mathrm{e}\left(\sum_{(k,j)\in\{1,\ldots,K\}\times\{1,\ldots,K\}/\{k\}} D_{kj}\ , g_2^{S_A}\right)$;
- $\mathrm{Q} = e(g_1, g_2)^{s_A(\sum v_k - R_u)} = \frac{Y^{s_A}}{e\left(g_1^{R_u}, g_2^{s_A}\right)}$;

- $h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))$;

These three values are the key parameters to decrypt the encrypted message. If the adversary can successfully compute these values, the algorithm can be considered as an insecure algorithm. The possibility for the adversary to get these values is investigated.

Given that:

$$\mathrm{e}(D_u, E_1) = \mathrm{e}\left(\sum_{(k,j)\in\{1,\ldots,K\}\times\{1,\ldots,K\}/\{k\}} D_{kj}\ , g_2^{S_A}\right)$$

where $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{kj} \in \mathbb{Z}_q$ and $E_1 = g_2^{S_A}$ is a public key.

For k^{th} AA, the adversary queries each AA for the secret. Assume j^{th} $(1 < j < k)$ AA receives the request. j^{th} AA issues $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$.

In $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$, the s_{kj} is the secret shared between j^{th} and k^{th} AA. Since there is only one honest AA, it will exchange the secret with all other corrupt authorities by computing $y_k^{x_j/(s_{kj}+u)}$ and $y_j^{x_k/(s_{jk}+u)}$ during the Setup stage.

The $R_{kj} \in \mathbb{Z}_q$ is issued by k^{th} AA during the Key Issuing stage. The adversary can get the value of D_{kj}. This is a potential risk of the MA-ABE scheme and it will be studied in Section 6.3.3.2.

For Q = $e(g_1, g_2)^{s_A(\sum v_k - R_u)} = \frac{Y^{s_A}}{e\left(g_1^{R_u}, g_2^{s_A}\right)}$, let us recall that the k^{th} AA issues a secret key $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$ for the eligible attributes of user u. $p_{k,u}(i)$ is a polynomial that can be computed as a linear combination of d known points. A (d – *1)* degree polynomial can be solved by d number of points on the polynomial. For the k^{th} AA, if the number of user's eligible attributes is greater than the threshold d_k. Then the user can get $p_{k,u}(0)$.

In order to obtain Q, for any d_k attributes $i \in A_C^k \cap A_u^k$ at k^{th} AA, three steps are executed:

Computes$e\left(S_{k,i}, C_{k,i}\right) = e(g_1, g_2)^{s_A p_{k,u}(i)}$.

The k^{th} AA interpolates all the values $e(g_1, g_2)^{s_A p_{k,u}(i)}$ together to get $P_{k,u} = e(g_1, g_2)^{s_A p_{k,u}(0)} = e(g_1, g_2)^{s_A\left(v_k - \sum_{j \in \{1,\ldots,K\}/\{k\}} R_{kj}\right)}$. Q can be obtained by multiplying all $P_{k,u}$.

For the adversary, all the corrupt AAs issue decryption keys for him. However, for the honest AA, the adversary with insufficient attributes cannot compute the value of $P_{k,u}$ because he is unable to get the value $p_{k,u}(0)$. Thus Q is not available for the adversary.

For the last the value of $h(M(a'a,1) \parallel M(a'a,2) \parallel \cdots M(a'a,n))$. With the behaviour-profiling app, user's context-related attributes can be captured at runtime and the necessary operations executed during decryption. Therefore, the mobile operating system should have a secure mechanism to protect the app and the data from being modified.

The recently introduced robust security software such as KNOX [194] and BES [195] virtually divided the mobile device into two isolated containers: personal user container and work container. The apps in one container are separated from the apps in the other container. The apps in work container are also virtually located in the user mobile device management server. Thus, the behaviour-profiling app can be verified by the device management server to prevent from being modified. Thus, modifying the behaviour-profiling app by malicious user in order to feed false results for the context-related attributes can be easily detected.

From the analysis above, it shows that if the adversary request secret keys from an authority which he has insufficient attributes (the honest authority), the adversary cannot output a correct guess of *b'* such that *b'* = *b.* Our scheme is secure based on the selective ID model.

Hence, our proposed context-aware single authority ABE scheme and context-aware MA-ABE scheme is secure in the selective ID model.

6.3.3.2 Key Issuing Analysis

In the anonymous key issuing protocol, the k^{th} AA will set up several parameters before staring the key issuing. Then it computes $X = (s_{kj} + u)\rho_1$, $X_1 = g^{\tau/x}$, and $X_2 = h^{\alpha\tau}$, where $g = y_j^{x_k}, h = g_1$. g and h are unknown to the user..

Now adversary can get that $X_1 = g^{\tau/x} = y_j^{x_k\tau/\left((s_{kj}+u)\rho_1\right)}$ and $X_2 = h^{\alpha\tau} = g_1^{\alpha\tau}$ where $\alpha = R_{kj}\ when\ k > j\ and\ \alpha = -R_{kj}\ when\ k < j.$

The challenger then computes $Y = \left(X_1^{\rho_1} X_2\right)^{\rho_2} = y_j^{\rho_2 x_k \tau/(s_{kj}+u)} g_1^{\alpha\tau\rho_2}$ with a random$\rho_2 \in \mathbb{Z}_q$. The k^{th} authority computes the $Z = Y^{\gamma/\tau} = y_j^{\gamma\rho_2 x_k/(s_{kj}+u)} g_1^{\gamma\alpha\rho_2}$.

Finally the user gets the key using

$D_{kj} = Z^{1/\rho_2} = y_j^{\gamma x_k/(s_{kj}+u)} g_1^{\gamma\alpha} = g_1^{\gamma x_k x_j/(s_{kj}+u)} g_1^{\gamma R_{kj}}$,where γ = 1 when k > j and γ = -1 when k < j.

Note that during the k^{th} AA issuing a key, the other AAs will also participate to compute the pseudorandom values: the shared secret s_{kj} and the public key $y_j = g_1^{x_j}$.

However, values of discrete logarithm between $g = y_j^{x_k} = g^{x_j x_k}$ and $h = g_1$ should be unknown. This also requires that the collusion group authorities cannot learn the other honest authority's private key x_j.

If there is only one honest AA, the adversary can extract the master secret such as s_{kj} ,during the key issuing process because of the incorporation of some pseudorandom values from other corrupt authorities. Then the protocol is not secure.

For k^{th} corrupted authority, the adversary obtains $y_j = g_1^{x_j}$, from the honest j^{th} AA ($1 < j < k$). The value is a pseudorandom value based on the shared secret s_{kj} and u. It also incorporates some pseudorandom values to get $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$. If there are only one honest authority, recall that each pair of authorities shares the s_{kj}, with the knowledge of $y_j, s_{kj}, u, and\ R_{kj}$ the adversary can easily compute D_{kj} of the honest authority as a result. Thus, the secret value

$e(D_u, E_1)$ is revealed to the adversary. All the other corrupt AAs are able to compute the pseudorandom values to compromise the system.

In this scenario, therefore, at least one more AAs are required to adjust its pseudorandom to compensate. This authority must be an honest AA that the adversary cannot request sufficient values.

Therefore, it can be concluded that our proposed context-aware MA-ABE scheme maintains the collusion resistance against up to $(N-2)$ Attribute Authorities.

6.3.4 Performance Analysis

In this section, computation and communication costs associated for both the single and multi-authority algorithms proposed in Section 6.2 and 6.3 are analysed. The efficiency of the proposed algorithms is demonstrated by comparing them against the related conventional ABE schemes. The single authority ABE scheme is discussed first followed by the discussion of MA-ABE scheme.

6.3.5 Computational Complexity Analysis

In single authority ABE scheme, the user is only involved in the computation process during the Decryption stage and the data owner involves computation process in the Encryption stage. The computational cost involved in the Setup and Key Issuing stages can be ignored since those can be done in the idle time. The computational cost of hash function is negligible compared to pairing and

exponentiation. Let us denote the computational time for one multiplication, one exponentiation, and one bilinear pairing as $C_m, C_{ex}, and\ C_p$, respectively. Denote the total number of attributes used for encryption as *n* and the total number of context-related attributes used by the data owner as *d.* Table 6.1 shows the total time required for encryption and for decryptions for the proposed scheme and conventional scheme.

	Conventional ABE Scheme	Proposed Scheme
Encryption	$(n + 1)C_{ex} + C_m$	$(n + 2)C_{ex} + 2C_m$
Decryption	$nC_p + nC_m$	$nC_p + (n + 2)C_m$

Table 6.1 Comparison of Computational Cost for the Single Authority ABE Scheme and the proposed Context-Aware Single Authority ABE Scheme

From Table 6.1, it shows that in the single authority scenario, our scheme takes just one more exponentiation operation and one more multiplication than the conventional ABE scheme. The decryption time of the proposed scheme is increased only by one multiplication time and independent of the number of context-related attributes. The proposed scheme adds an additional layer of security for negligible computational expense.

Table 6.2 compares the computational complexity of the proposed MA-ABE scheme with Chase and Chow's scheme [166]. In the multi-authority scenario, the users are involved in the Decryption stage while the data owner is involved in the Encryption stage. For simplicity, denote the total number of AAs in the system as *K*

and the data owner uses *n* number of attributes from each AA for encryption. Let us use the same benchmark time values given for jPBC library for comparison[196]. Table 6.3 shows the time values (in *ms*) for $C_m, C_{ex},$ and C_p based on two different test beds:

(1) Intel (R) Core (TM) 2 Quad CPU Q6600 with 2.50GHz and 3GB memory running on the Ubuntu 10.04

(2) HTC Desire HD A9191running on Android 2.2.

The time values given in Table 6.3 are for a symmetric elliptic curve called Type *A* curve which has a base field size of 512-bit and the embedding degree is 2. The type *A* curve has a 160-bit group order. Let us assume that the data owner uses an environment similar to the Test bed 1 for encryption while the user uses a mobile device similar to Test bed 2 for decryption.

	Conventional ABE Scheme	**Proposed Scheme**
Encryption	(nK + 2)C_{ex} + C_m	(nK + 3)C_{ex} + 2C_m
Decryption	(nK+1)C_p + (nK + 1)C_m	(nK + 1)C_p + (nK + 3)C_m

Table 6.2 Comparison of Computational Complexity of Proposed MA-ABE Scheme with Chase and Chow's Scheme

	Test Bed 1 (ms)	**Test Bed 2 (ms)**
C_p	14.6	491.2
C_{ex}	2.8	34.1
C_m	1.8	20

Table 6.3 Time Complexity Measures for Two Different Test Beds

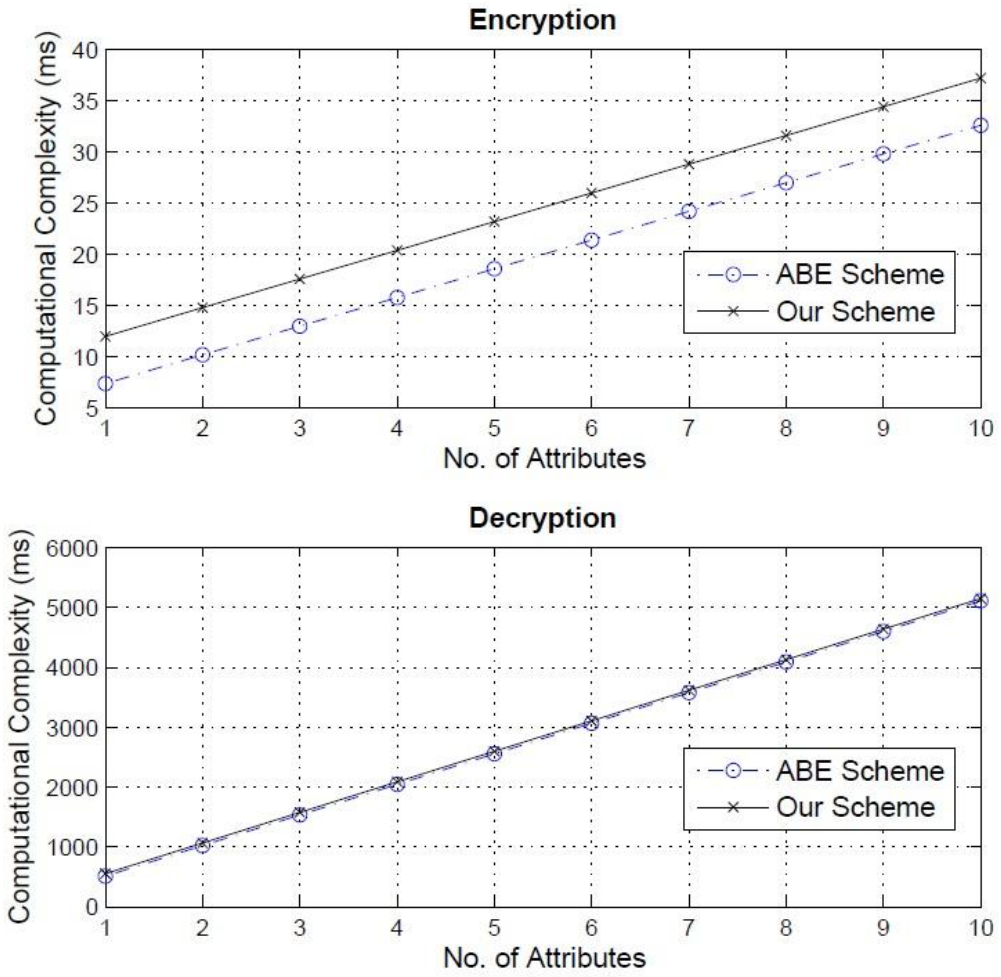

Figure 6.3Comparison of Computational Costs for Encryption Between Conventional MA-ABE Scheme and the proposed Context-Aware MA-ABE Scheme

Figure 6.3 shows the computational complexity in terms of total time required for the data owner and the user to encrypt and decrypt the data, respectively for conventional MA-ABE scheme and the proposed scheme when $K = 1$. For encryption, our scheme consumes nearly *5ms*more than the conventional ABE. However, the proposed scheme incorporates the context-related attributes during the encryption which provides runtime security to the data owner. It is worth noting that from Figure 6.3 that the time difference between our scheme and the conventional ABE for encryption is independent of the number of attributes (i.e. time complexity orders for both schemes are same for encryption). However, our scheme is capable of including context–related attributes on top of the regular attributes. For decryption, it is obvious from Figure 6.3 that our scheme performs equally well as the conventional ABE scheme.

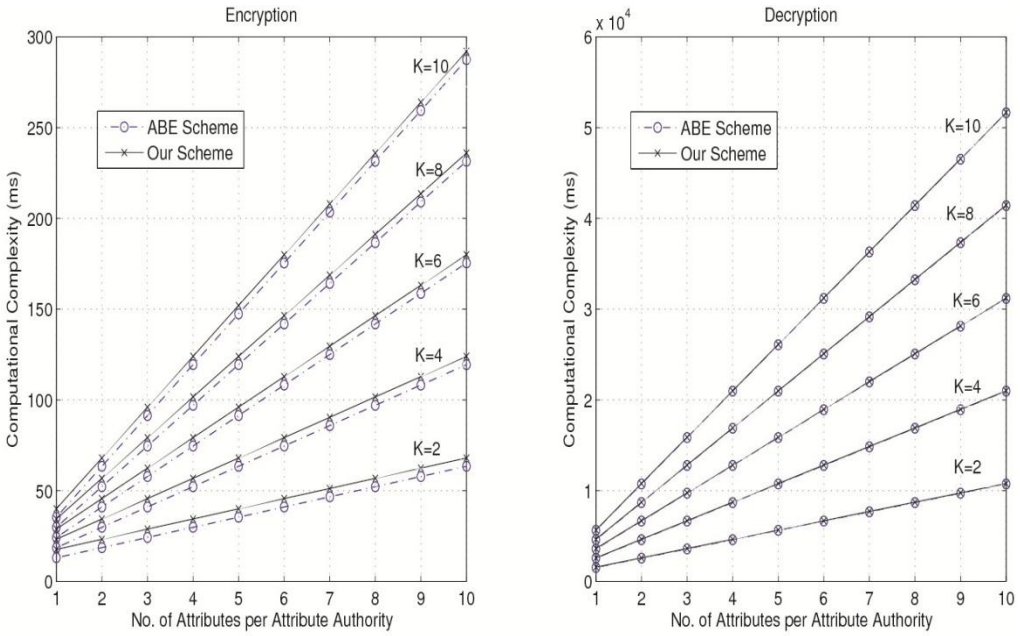

Figure 6.4Comparison of Computational Costs for Decryption Between Conventional MA-ABE Scheme and the Proposed MA-ABE Scheme

Figure 6.4 compares the decryption costs for both the proposed scheme and the conventional MA-ABE scheme in terms of time complexity for different number of AAs (i.e. K = 2, 4, 6, 8, 10).Encryption and decryption time increases with the total number of AAs. For encryption, similar to K = 1 case, the time complexity orders of both the schemes are same (i.e. our scheme consume nearly *5ms* more than the conventional MA-ABE scheme irrespective of number of attributes and number of AAs). Moreover, for decryption, our scheme performs equally well as the conventional MA-ABE scheme regardless of number of AAs involved in the encryption. As seen from both Figure 6.3 and Figure 6.4, time complexity for decryption is nearly 100 times more than the encryption due to the limited process power at the mobile device. The time complexity can be reduced if less number of attributes are used for encryption. The proposed scheme enables the data owner to reduce the time complexity at the user end by reducing the number of attributes from the AA. However, our scheme adds an extra layer of security by adding the context-related attributes with negligible increment in complexity.

The proposed scheme enables the data owner to reduce the time complexity at the user end by adding more context-related attributes with negligible increment in complexity. For instance, the data owner can include five attributes from AAs and another five context-related attributes which almost reduces the complexity by half compared with the conventional MA-ABE scheme. However, the proposed scheme has additional security in terms of context-related attributes which is not possible in conventional ABE scheme. In a nutshell, the proposed scheme does not degrade the performance of conventional ABE scheme while including context-related

attributes to enhance the security of the employer's data while reducing the complexity at user's end.

6.3.6 Communication Complexity Analysis

The communication costs for the proposed schemes and the conventional schemes are relying on the Key Issuing stage and when uploading and downloading the data. Since, Key Issuing stage purely dependent on communication between authorities and the data owner. Communication costs for both the schemes in Key Issuing Stage are equal. During uploading and downloading stage, only the extra components added to the proposed scheme compared to the conventional ABE scheme are E_0 = h(M($a_{a,1}$)‖M($a_{a,2}$)‖…M($a_{a,n}$))$Y^{s_A+s_B}$, which requires 160-bits and A_C (3-bits are enough to represent one context-related attribute out of eight context-related attributes).Overall, increment in the communications cost in the proposed algorithm is negligible.

From the performance analysis in the previous section, it shows that the workload of the computation and communications are heavy for the mobile users. For the communication, each user has to communicate with all the authorities for requesting keys and execute required computations in order to obtain the decryption keys and data. These interactions and computational works are a burden for the mobile user in terms of communication and computation complexity. The reliability of mobile data network is also a challenge for completing all required communications. The next section will present a solution to address these issues.

6.4 Low-Complexity Multi-Authority Attribute-Based Encryption Scheme

This section presents the low-complexity MA-ABE scheme which substantially reduces the communication and computation workload to the user compared to both the conventional MA-ABE scheme and the proposed context-aware MA-ABE scheme presented in Section 6.3.

Since mobile Cloud computing serves mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platform [197]. The author introduced a Cloud server based semi-trusted-authority (STA) between the user and the AAs. The user equipped with smart devices only provides pseudonym of his identity to the STA. Then the STA interacts with all the AAs on-behalf of the user and obtains decryption-keys. Later, STA combines all the keys to obtain a key and pass it to the user. The user has the ability to do necessary computations on the received keys and obtains the final decryption keys to recover the data. Since all the distributed keys provided by the AAs are masked, STA cannot decrypt the data. Moreover, STA cannot pool all the keys and obtain attributes of the user. Hence, our algorithm preserves the security and privacy of Chase and Chow's MA-ABE scheme while outsourcing the computational and communication overheads to the STA.

Assume that the STA will execute the protocol correctly that he will behave in a semi-honest manner, e.g. he is honest but curious so privacy is a real issue.

6.4.1 Constructions

There are five different parties involved in the proposed framework: the data owner (encryptor), the user (decryptor), the AA, STA and the Cloud storage server. Figure 6.5 depicts the main framework and work flow of the proposed system.

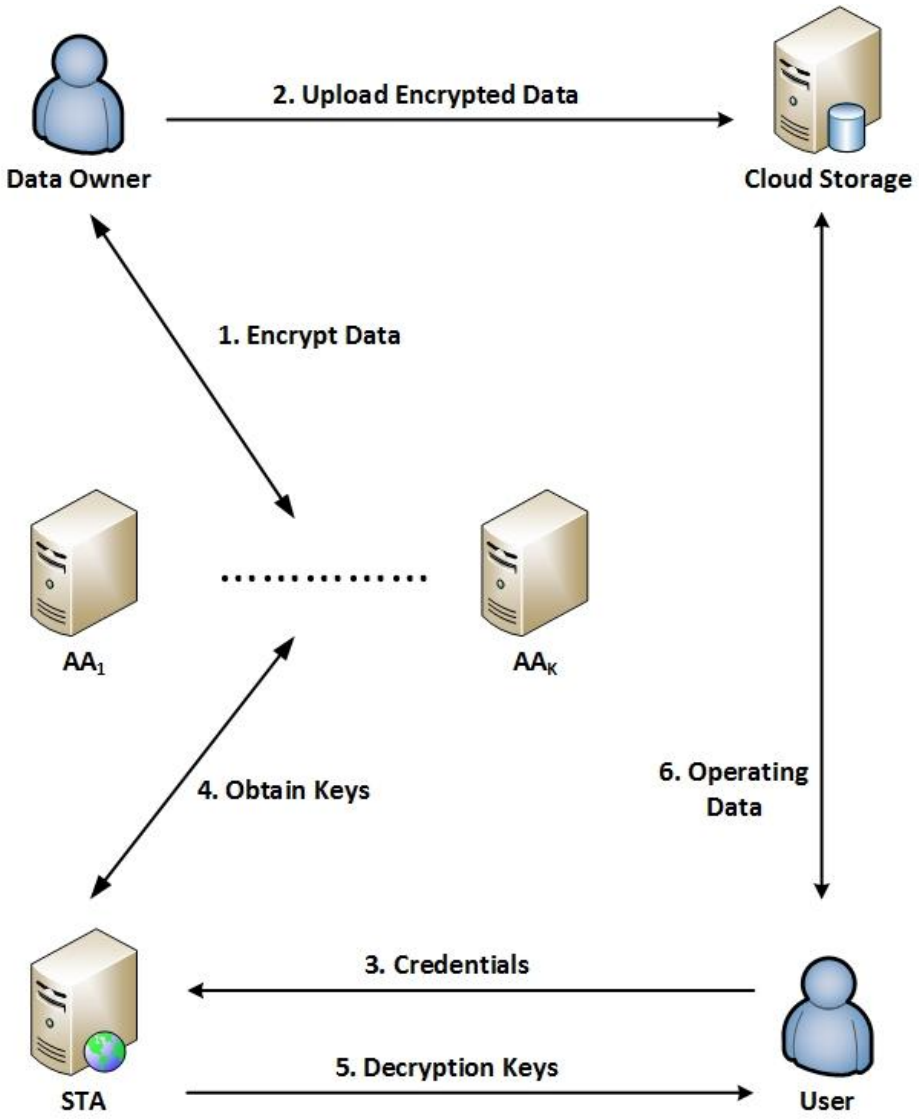

Figure 6.5Illustration of the proposed Framework of Low-Complexity Context-Aware Multi-Authority ABE Scheme for Mobile Cloud Environment

The algorithm is composed of four sub algorithms which are named as: Setup, Key Issuing, Encryption and Decryption. In the following functionalities of each algorithm are briefly explained.

- *Setup*: The setup algorithm takes a security parameter as input and outputs a bilinear group and a set of security parameters.
- *Key Issuing*: AAs generate decryption keys for a user *u* that holds a set of attributes.
- *Encryption*: The encryption algorithm takes a set of attributes maintained by AAs as well as a set of context-related attributes defined by data owner as inputs and it outputs the ciphertext.
- *Decryption*: The decryption algorithm takes as input the decryption credentials received from AAs and context-related parameters obtained from smart mobile device and the ciphertext. The output will be the original data.

Define each algorithm as follows:

Setup $\mathcal{S}$

For a given security parameter λ and $\sigma \in \{0,1\}^{poly(\lambda)}$, group bilinear parameters are generated by the AA: $q,\ g_1, g_2,\ \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \leftarrow S(1^\lambda;\ \sigma)$.

Now, the AAs interact with each other and execute the following:

- k^{th} AA randomly chooses $v_k \in_R \mathbb{Z}_q$ and computes $Y_k = e(g_1, g_2)^{v_k}$, and sends Y_k to other AAs, where each AA computes $Y = \prod Y_k = e(g_1, g_2)^{\sum_k v_k}$.
- Each pair of AAs shares a secret, k^{th} AA and j^{th} AA randomly choose $s_k \in \mathbb{Z}_q$, such that $s_{kj} = s_{jk}$.
- k^{th} AA randomly chooses $x_k \in \mathbb{Z}_q$ and computes $y_k = g_1^{x_k}$. Using the shared secret s_{kj} and u, k^{th} AA and j computes $y_k^{x_j/(s_{kj}+u)}$ and $y_j^{x_k/(s_{jk}+u)}$ respectively.

Then each AA carries out the following steps individually:

- For i^{th} attribute stored in the k^{th} AA, where $i \in \{1, \dots, n_k\}$, and $k \in \{1, \dots, K\}$ the k^{th} AA randomly chooses a secret $t^{k,i} \in \mathbb{Z}_q$ and computes corresponding public key as $T_{k,i} = g_2^{t_{k,i}}$.

Key Issuing

The STA executes the following steps with k^{th} AA on behalf of user u, hence the following communication and computational overheads have been offloaded to STA. In order to mask the decryption keys to the STA, a pre-shared secret between user and k^{th} AA is used when issuing keys, the key will be combined into the pseudorandom and only the user can derive the decryption key.

- For $j \in \{1, \dots, K\}/k$, STA gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{kj} \in \mathbb{Z}_q$ is a random value.

After obtaining all D_{kj}, STA computes

- $D_u = \prod_{(k,j)\in\{1,\ldots,K\}\times\{1,\ldots,K\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j)\in\{1,\ldots,K\}\times\{1,\ldots,K\}/\{k\}} R_{kj}$.
- If user u satisfies d_k number of attributes, then k^{th} AA randomly picks a $(d_k - 1)$ degree polynomial $p_{k,u}(\cdot)$
- If the user doesn't satisfy d_k number of attributes, the k^{th} AA randomly picks a polynomial $p_{k,u}(\cdot)$ with degree n_k+1, where $n_k$denotes the number of attributes managed by k^{th} AA.
- Now, k^{th} AA use the pre-shared secret, $r_k$between user u and k^{th}AA, define $p_{k,u}(0) = v_k + r_k - \sum_{j\in\{1,\ldots,K\}/\{k\}} R_{kj}$.
- The kth AA computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}, i \in [1, \ldots, n_k]$ for each eligible attribute i for the user.

Encryption $\mathcal{E}$

- The data owner encrypts the message m for attribute set A_c= $\{A_C^1, \ldots, A_C^K\}$ The data owner picks $s \in \mathbb{Z}_q$ and outputs $\langle E_0 = mYs, E1=g2s, Ck,i= Tk,isi\in ACs, \forall k\in 1, \ldots, K$.
- Now the data owner uploads the encrypted data and related parameters to the cloud storage.

Decryption $\mathcal{D}$

The decryption stage has two steps. Firstly STA obtains the masked keys from all the authorities and combine them together. Then the combined key will be forwarded to the user to get the unmasked the combined key.

- Decryption by STA
  - For each authority *k*:

    (1) For any attribute $i \in \mathbb{A}_C^k \cap \mathbb{A}_u^k$, STA computes $e\left(S_{k,i}, C_{k,i}\right) = e(g_1, g_2)^{sp_{k,u}(i)}$.

    (2) STA interpolates all $e(g_1, g_2)^{sp_{k,u}(i)}$ together and gets $P_k = e(g_1, g_2)^{sp_{k,u}(0)} = e(g_1, g_2)^{s\left(v_k + r_k - \sum_{j \in \{1,\ldots,K\}/\{k\}} R_{kj}\right)}$.
  - STA multiplies all P_k together, and gets $Q = e(g_1, g_2)^{s\sum(v_k + r_k) - sR_u} = \frac{Y^{s+s\sum r_k}}{e\left(g_1^{R_u}, g_2^s\right)}$.
  - Then STA computes $T = e(D_u, E_1) \cdot Q = e\left(g_1^{R_u}, g_2^s\right) \cdot Q = Y^{s+s\sum r_k}$ and forwards T to the user.
- Decryption by User
  - User holds the pre-shared secret r_k, computes $\prod e\left(g_1^{r_k}, g_2^s\right) = e(g_1, g_2)^{s\sum r_k} = Y^{s\sum r_k}$.
  - In order to recover the message, the user computes $Y^s = T/Y^{s\sum r_k}$, recover *m* as $m = \frac{E_0}{Y^s}$.

6.4.2 Security Analysis

This section analyses the security of the proposed low-complexity MA-ABE scheme. As the proposed scheme is an extension of Chase and Chow's MA-ABE scheme [166], the security analysis showed that the proposed scheme does not degrade the security and privacy of the encrypted message and mobile user compared to original scheme. It satisfies data confidentiality of encrypted data against unauthorized users and the curious Cloud service providers under the selective ID model. It maintains the collusion resistance against up to $(N-2)$ AAs. The user's privacy is also protected by the anonymous key issuing protocol.

The stages that differ between the two schemes are focused on, the Key Issuing stage and the Decryption stage.

During the Decryption stage, the STA performs the steps in place of the authority AA. In more detail, the STA only computes $T = e(D_u, E_1) \cdot Q = e\left(g_1^{R_u}, g_2^s\right) \cdot Q = Y^{s+s\sum r_k}$ in contrast to Y^s that is computed by the authority in the Chase and Chow's scheme. As the required decryption key to decrypt the message *m* is Y^s, the STA cannot decrypt to obtain the message *m*, therefore the confidentiality of the message is ensured. More precisely, since the shared secret r_k is only known between the k^{th} AA and the mobile user, and thus the summation $\sum_k r_k$ can only be obtained by a mobile user; therefore the STA cannot obtain Y^s for its know expression of $T = Y^{s+s\sum r_k}$.

During the Key Issuing stage, the STA performs most of the steps in place of the user in Chase and Chow's scheme. The k^{th} AA computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}, i \in [1, \dots, n_k]$ and sends them to STA. If the user satisfies the minimum d_k number of attributes, then the degree of polynomial chosen by AA is equal to d_k. Hence, d_k number of $S_{k,i}$ can be used to get the secret $p(0) = v_k + r_k - \sum_{j \in \{1,\dots,K\}/\{k\}} R_{kj}$ during the interpolation. If the user does not satisfy the minimum d_k number of attributes then the user cannot obtain enough number of key $S_{k,i}$ to recover the secret $p(0)$. This is the crucial point that for the polynomial with degree $(d_k - 1)$. A number of d_k points are required in order to get $p(0)$. If the user with insufficient attributes, then the k^{th} AA chooses a polynomial with degree $(n_k + 1)$. Therefore, the STA is not able to pool all $S_{k,i}$ from all AAs in order to find attributes of mobile user, and cannot distinguish which set of attributes belongs to the mobile user. This also preserves the privacy of the user.

6.4.3 Performance Analysis

In Chase and Chow's scheme, the user takes part in the computation during the Key Issuing and Decryption stages. Most of the required computation and communications take place in these two stages. Our proposed scheme successfully offloads these time-cost operations to the semi-trusted Cloud based server. Since the Cloud servers have powerful computing capacity and reliable network, utilizing these powerful resources is an ideal way.

6.4.3.1 Reduction in Communication Overhead

The communication overhead is always an important factor for mobile Cloud environment. It is important for the service performance and user experience. In Chase and Chow's scheme, user needs to start $(N-1)$independent invocations for each AA during the key issuing stage. In order to compute the decryption key D_u, a user has to request the key D_{kj} from all the authorities. This process requires a large number of communications. The more authorities requested, the more communications take place. It is a big challenge for mobile data networks to complete the process. Our proposed scheme offloads these communications to the Cloud based server, so that the communication overheads can be avoided at the user's end. Figure 6.6 shows the comparisons of number of communications between the conventional MA-ABE scheme and our proposed scheme. From Figure 6.6, the number of communications in Chase and Chow's scheme goes up much more than our proposed scheme. In our scheme, the user only needs to do communications with the STA, which improves the communication performance.

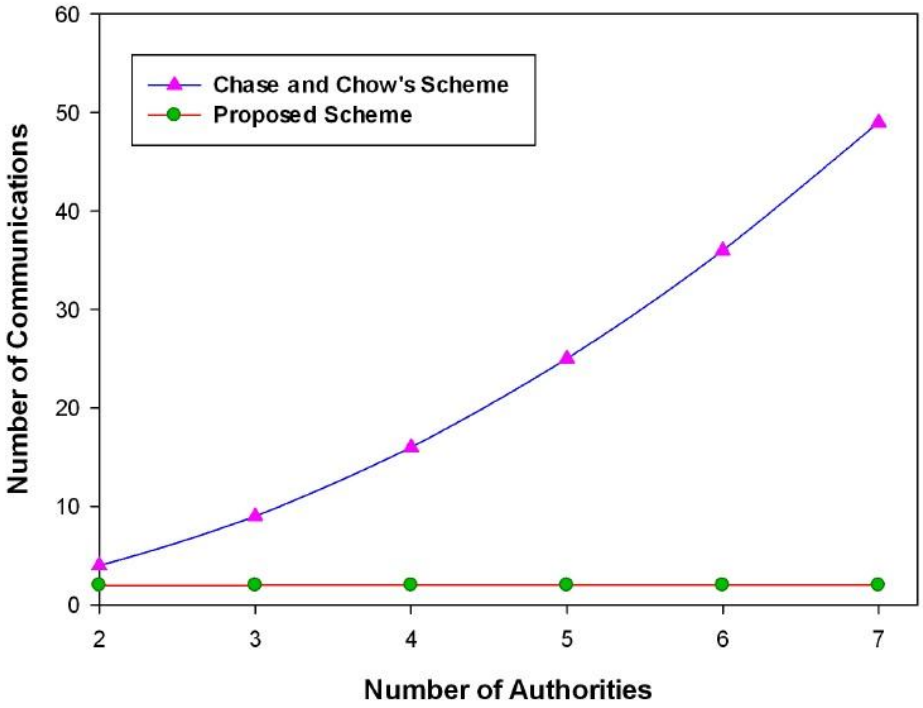

Figure 6.6Comparisons of Communications between Chase and Chow's MA-ABE Scheme and proposed Low Complexity MA-ABE Scheme

6.4.3.2 Reduction in Computational Overhead

In both the conventional MA-ABE scheme and proposed low complexity MA-ABE scheme, the user (data requester) has to execute several computations which are required in the Key Issuing and Decryption stages in order to decrypt the data. The computational time (in ms) for one multiplication, one exponentiation and one pairing as C_m, C_{ex}, C_p, respectively. Let us also denote the total number of attributes of each authority *n* and total number of attributes used by the data owner as *d,* and *N* denotes the total number of authorities. The following table shows a comparison of the number of required computations between the conventional MA-ABE scheme and the proposed scheme. From the results, it shows that for both

Key Issuing and Decryption stages, the high-cost computational tasks are substantially offloaded to the Cloud-based server.

	Conventional MA-ABE Scheme	**Proposed Low Complexity MA-ABE Scheme**
Key Issuing	N(N-1)C_m	_____
Decryption	(Nn+1)C_p + (Nn + 1)C_m	NC_p + (N+1)C_m
Total	(Nn+1)C_p +(N^2 + Nn +1)C_m	NC_p + (N+1)C_m

Table6.4Comparison of Number of Required Computations between the Conventional MA-ABE Scheme and Proposed scheme

6.5 Conclusion

The concept of ABE scheme has been demonstrated to perform a step closer in securely sharing the data. The traditional ABE is based on a single authority which manages all the attributes, communications and also issues the decryption keys. From the security aspect, if the authority was attacked, the whole system will be compromised. From the performance aspect, the authority is designed with too much responsibility. It can be the bottleneck of the whole system when a large number of users request keys at the same time. Furthermore, in the mobile Cloud environment, the available contextual attributes should be used to strengthen the security and data access control due to mobility.

In this chapter first a context-aware single authority ABE scheme is proposed which extended the traditional ABE scheme by incorporating contextual attributes in the mobile Cloud environments. Thus, the decryption keys are not only issued based on the attributes that are maintained by AAs, but also rely on policies defined by the data requestor. Following this, a context-aware MA-ABE scheme which is more practical is presented. In reality, attributes are managed by different AAs. A user requests the decryption keys from different authorities. The proposed scheme also removes the dependency on the central authority which issues final decryption keys and monitors all the communications to reduce the risk level of the system.In this way, users can combine all the keys received from all the AAs and the real-time contextual attributes captured by the mobile device in order to obtain the final decryption key. Then, the author proposed the third scheme, the low-complexity MA-ABE scheme, which effectively offloads the high-cost computational and communication workloads to Cloud based Semi Trusted Authority (STA). By

analysing the second scheme, the author found that the performance of mobile device to complete all the processes may degrade together with the user experience. High computation workloads should be offloaded to the Cloud. The STA which works on behalf of the user is introduced, thus the performance of mobile devices are much improved. The communication overheads and computation work are reduced compared with other schemes. Meanwhile, the designed protocol maintains the system security and data access at the same level as the previous schemes.

7 Conclusions and Future Work

7.1 Summary and Conclusions

This thesis investigated security and privacy issues in the emerging mobile Cloud paradigm. Traditional access control techniques developed for Cloud environment do not support both the privacy and security issues in the mobile Cloud environment. In mobile Cloud environment, due to the seamless interaction between the user and smart devices, the user generates a large volume of personal data. These data are eventually collected by the mobile Cloud service providers and users do not have much control over their utilisation. In order to enhance the user privacy together with security in the mobile Cloud environment, various access control techniques were proposed in this thesis. The conventional attribute-based access control and attribute-based encryption (ABE) schemes were modified in the proposed schemes in order to support the access control in mobile Cloud environments.

Mobile Cloud environment is a combination of Cloud computing technology together with mobile Internet. It delivers tailored conventional Cloud services to smart mobile devices. This enables the smart devices to offload computational, communicational, and storage tasks to the Cloud servers. In mobile Cloud environment, Cloud service providers collect very sensitive real-time user information such as location, user behaviour and contact details. The existing management systems such as .NET Passport, OpenID, Liberty Alliance, Higgins,

and OAuth were solely developed to protect user's identities in conventional Cloud environments. The author compared these identity management systems and investigated the issues where such identity management systems are used in mobile Cloud environments. Moreover, conventional access control mechanisms such as DAC, ManAC, RBAC, and ABAC, privacy protection languages such as P3P, EPAL, and XACML and ABE encryption techniques cannot be directly extended to the mobile Cloud environment.

For mobile Cloud environment, three techniques were proposed to protect users' privacy and security of online data: (1) deploying an access control technique to control the usage of user data in Chapter 5, (2) privacy-preserving technique to protect the user data in Chapters 5 and 6, and (3) storing the online data in the encrypted form in Chapter 6. Implementation of a user-centric attribute-based accesscontrol model is also given for mobile Cloud environments. The proposed techniques mutually authorize the user and the service provider (SP). XACML-based access control protocol was proposed to enable mobile usersto define access control policies for their online data. SPs need to satisfy the privacy access policy in order to obtain users' data. An attribute authority (AA) is proposed to maintain attributes of the users and SPs. This assures the status of a SP so that user's sensitive data will not be revealed to a malicious third party. The available real-time context-related attributes such as time, date, and location are used during the authorization process. When a requester requests user's privacy data, the system will pick required real-time attributes to verify the requesterduring the policy evaluation process. The model is built on top of the XACML standard, which is one of the popular privacy-preserving languages. he standard XACML architecture was

extended by using attributes from the mobile users, SPs, and surrounding environment so that the authorization can be done at runtime. Thus, the access control of user's data in mobile Cloud environment is more secure than the traditional access control schemes.

Using XACML requires the system to follow the standards of XACML. All the message flows should be formatted into the XACML specification, which is a drawback in the collaboration environment such as the mobile Cloud environment. Different domains would define their own structures and protocols to set up an access control system. For a mobile user of domain A, it would be infeasible to request data from domain B if domain B could not understand the requested message from domain A. Furthermore, policy evaluation would be a long-time process if there are a large number of policies. In order to address these issues, the author investigated the ABE schemes.

ABE scheme was modified to ensure the security of data together with privacy. There were three different ABE schemes proposed in this thesis for data and services access in the mobile Cloud environment. In the proposed schemes, the data are stored in the encrypted formatso that data confidentiality is guaranteed. ABE enables the data owner to define access policy during the encryption process. Due to the features of mobile users and mobile platforms in mobile Cloud environment, contextual attributes are available and can be used to strengthen the security and data privacy. The first technique is a single authority context-aware ABE scheme which incorporates contextual attributes. The data owner defines privacy access policy during the encryption process and uploads the encrypted data. An AA maintains a set of attributes, issues decryption keys and can monitor all the

communications. After receiving the decryption keys from the AA, the contextual attributes of the data requester are also collected by data requester's mobile device. These attributes are combined with the decryption keys to recover the original data.

Single authority ABE scheme is impractical since the single authority has too much control, such as manages users' attributes, issues decryption keys, and monitor communication. All the data will be lost if it is compromised by an attacker. Also single authority system cannotbe scaled due to the bottleneck problem if there are a large number of users requesting keys. Meanwhile, in reality, different sets of attributes are stored in different AAs. In order to incorporate this reality, a context-aware multi-authority attribute-based encryption (MA-ABE) scheme was proposed. More than one AA maintains different sets of attributes and issues decryption keys. In order to protect user's identity from being tracked by each AA, an anonymous key issuing protocol was proposed. Using this protocol, the data requester's real identity cannot be revealed to each AA.Hence, an AA cannot track the requester's data transactions. Due to the nature of the anonymous key issuing protocol, as long as there are two honest authorities and other AAs are malicious; the rest of the malicious AAs cannot combine the requester's data that they hold to recover the original text. Thus, the collusion attacks are prevented. The proposed scheme removed the central authority, which takes charge of issuing decryption keys and monitoring all the activities. As a result, the central authority will not be the bottleneck if the system is designed in a large scale. The loss of corruption of central authority can be avoided using the proposed scheme.

In order to decrypt the ciphertext, the data requester must satisfy the access policy. The requestor's mobile device capturesrequired contextual attributes and

combines them with the decryption keys received from all the AAs. The data requester has to contact all the AAs to get the decryption keys. The communication overheads will be high if there are a large number of AAs. In mobile Cloud environment， considering network conditions, the reliability of data network to complete such communicationsis unpredictable.This can bea drawback. Furthermore, final decryption key is computed based on the keys received from all the AAs and attributes captured by the mobile devices, hence the computational work is a burden. Those two limitations affect the performances of the mobile Cloud services and lower the quality of services.

A low-complexity ABE scheme was proposed to address the issues that exist in the context-aware MA-ABE scheme as a third technique. A Cloud-based Semi-Trusted Authority (STA) is introduced, thus, the high-cost communications and computationsare migrated from the mobile end to the Cloud end. This method improves the performance at the mobile end. In this proposed algorithm, a user is not required to make contact with the AAs; STA requests the keys on behalf of the user. The user has a pre-shared secret with each AA, which is unknown to the STA. When issuing decryption keys, each AA embeds the pre-shared secret into the decryption keys.Therefore, STA cannot compute the final decryption key to recover the message. The security analysis showed that the proposed low-complexity technique prevents collusion attacks. The proposed techniques not only secure the users' data, but also empower the mobile users to control the privacy of their data.

Finally, this thesis proposes the data access control techniques and ABE schemes to protect users' privacy and sensitive data from unauthorized access.

In this part of the research work, an attributes-based access control model for mobile Cloud environment is proposed to restrict access of users' online data. The author deployed access control techniques to protect the user's data access. XACML is used as the privacy access policy language, which is a flexible and user-centric solution for mobile environment..

(1) The MA-ABE scheme for mobile Cloud environments is also developed and implemented to protect user privacy and data security from unauthorized access and fraud. As each of the AAs maintains different sets of attributes for issuing decryption keys for the use. The MA-ABE scheme removes the central authority and improves the system performance by removing single point of failure and providing end to end security

(2) The incorporation of context-related attributes compared with the conventional ABE enables the data owner to define the privacy access policies for their personal data at runtime. The mobile devices of users will capture the dynamic contextual attributes to satisfy the privacy access policies. It strengthens the privacy and data protection in a more flexible, user-centric, and fine-grained manner.

(3) In addition to this, an anonymisation key issuing protocol is designed to protect the users' identity during data transactions. An

AA cannot track a transaction and identify the user who initiates the transaction. Furthermore, the AAs cannot combine the information they hold to reveal the original text as long as there are two honest authorities. The proposed scheme maintains the collusion resistance against up to $(N - 2)$ AAs.

7.2 Recommendations for Future Work

The following summaries some of the future research work that could be carried out in this area based on our research findings.

- The use of context-related attributes can be further investigated. With the development of the mobile operating systems, mobile applications for detecting and analysing user's behaviours can derive more valuable and accurate contextual attributes to define the risk level of a user. The incorporation of such applications with Cloud based services can be investigated to develop highly secure solutions in mobile Cloud environments.
- Enable the data owner to define more complex access structures to restrict the access to his online resources.
- Adding a key revocation mechanism. If a user no longer holds an attribute, the relevant private key should be updated by the authority in parallel. The user could not use the previous key to operate the resource any longer.

- Developing completely decentralized ABE scheme where new AA can join and leave the mobile Cloud environment without bootstrapping the whole system. This enables seamless interaction between the user and SPs and will be ideal for multi-Cloud environments. However, developing such algorithms under standard security assumptions such as decisional bilinear Diffie-Hellman is very challenging at the same time promising.

Bibligraphy:

[1] S. Weiguang and S. Xiaolong, "Review of Mobile cloud computing," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 1-4.

[2] C. Ivan and R. Popa, "Cloud based Cross Platform Mobile Applications Building and integrating cloud services with mobile client applications," *Advances in Computer Science: an International Journal*, vol. 3, pp. 69-77, 2014.

[3] S. Hui, L. Zhuohua, W. Jiafu, and Z. Keliang, "Security and privacy in mobile cloud computing," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, 2013, pp. 655-659.

[4] V. C. LEUNG, Y. WEN, M. CHEN, and C. RONG, "MOBILE CLOUD COMPUTING," *IEEE Wireless Communications*, p. 13, 2013.

[5] A. N. Khan, M. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, pp. 1278-1299, 2013.

[6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, pp. 1587-1611, 2013.

[7] W. Jansen and T. Grance, "Guidelines on security and privacy in public cloud computing," *NIST special publication*, vol. 800, p. 144, 2011.

[8] K. Sangani, "Rolling out the mobile future," *Engineering & Technology*, vol. 7, pp. 80-81, 2012.

[9] S.-Z. Yang, "The marketing chain in the mobile Internet era," in *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, 2011, pp. 1058-1061.

[10] E. report. (Jan, 2013). *Three Out of Four UK Mobile Users to Own Smartphones by 2016*. Available: http://www.emarketer.com/Article/Three-of-Four-UK-Mobile-Users-Own-Smartphones-by-2016/1009614

[11] W3C. (2014). *Extensible Markup Language (XML)*. Available: http://www.w3.org/XML/

[12] D. Crockford. (2014). *Introducing JSON*. Available: http://json.org/

[13] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon, "Simple object access protocol (SOAP) 1.2," *World Wide Web Consortium*, 2003.

[14] (Apr, 2013). *Apple iOS*. Available: http://www.apple.com/uk/ios/

[15] (Apr. 2013). *Google Android*. Available: http://www.android.com/

[16] Statista. (June 2014). *Cumulative number of apps downloaded from the Apple App Store from June 2008 to June 2014*. Available: http://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store/

[17] Statista. (Sep. 2014). *Number of available apps in the Apple App Store from July 2008 to September 2014*. Available: http://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/

[18] Statista. (July 2013). *Cumulative number of apps downloaded from the Google Play Android app store as of July 2013*. Available: http://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play/

[19] V. H. (24, July, 2013). *Android's Google Play beats App Store with over 1 million apps, now officially largest*. Available: http://www.phonearena.com/news/Androids-Google-Play-beats-App-Store-with-over-1-million-apps-now-officially-largest_id45680

[20] AppBrain. (2014). *Number of Android applications*. Available: http://www.appbrain.com/stats/number-of-android-apps

[21] T. Soyata, H. Ba, W. Heinzelman, M. Kwon, and J. Shi, "Accelerating mobile cloud computing: A survey," *Communication Infrastructures for Cloud Computing*, 2013.

[22] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, 2011.

[23] M. Almorsy, J. Grundy, and A. S. Ibrahim, "Collaboration-based cloud computing security management framework," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 364-371.

[24] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50-58, 2010.

[25] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication,* vol. 800, p. 7, 2011.

[26] L. Zhong, B. Wang, and H. Wei, "Cloud computing applied in the mobile internet," in *Computer Science & Education (ICCSE), 2012 7th International Conference on*, 2012, pp. 218-221.

[27] R. Jain, "Quality of experience," *IEEE MultiMedia,* vol. 11, pp. 96-95, 2004.

[28] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *Selected Areas in Communications, IEEE Journal on,* vol. 14, pp. 1228-1234, 1996.

[29] F. c. G. LLC. (2013). *The Definition of Customer Quality of Experience*. Available: https://www.findyourcloud.com/Articles/Quality-of-Experience/The-Definition-of-Customer-Quality-of-Experience/

[30] Z. Peng and Y. Zheng, "A QoS-aware system for mobile cloud computing," in *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, 2011, pp. 518-522.

[31] F. c. G. LLC. (2013). *How to Measure QX Evidence*. Available: http://www.findyourcloud.com/Articles/Research-2.0/How-to-Measure-QX-Evidence/

[32] (2013). *Instagram*. Available: http://instagram.com/

[33] (2013). *Flickr*. Available: http://www.flickr.com/

[34] (2013). *Facebook*. Available: https://www.facebook.com/

[35] Gartner. (April 2013). *Gartner Says By 2016, 40 Percent of Mobile Application Development Projects Will Leverage Cloud Mobile Back-End Services*. Available: http://www.gartner.com/newsroom/id/2463615

[36] L. Zhang, X. Ding, Z. Wan, M. Gu, and X.-Y. Li, "WiFace: a secure geosocial networking system using WiFi-based multi-hop MANET," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, 2010, p. 3.

[37] M. Pitkänen, T. Kärkkäinen, J. Ott, M. Conti, A. Passarella, S. Giordano, D. Puccinelli, F. Legendre, S. Trifunovic, and K. Hummel, "SCAMPI: Service platform for social aware mobile and pervasive computing," *ACM SIGCOMM Computer Communication Review,* vol. 42, pp. 503-508, 2012.

[38] D. Recordon and D. Reed, "OpenID authentication 2.0-final," ed: December, 2007.

[39] Eclipse. (2011). *Higgins 2.0*. Available: http://eclipse.org/higgins/

[40] T. LLC. (2013). *Tripadvisor*. Available: http://www.tripadvisor.co.uk/

[41] O. K. a. S. Moritz. (2013). *Carriers Sell Users' Tracking Data in $5.5 Billion Market*. Available: http://www.bloomberg.com/news/2013-06-06/carriers-sell-users-tracking-data-in-5-5-billion-market.html

[42] G. Sloane. (2014). *Foursquare Starts Selling All That Data*. Available: http://www.adweek.com/news/technology/foursquare-starts-selling-all-data-158628

[43] A. TROIANOVSKI. (2013). *Phone Firms Sell Data on Customers*. Available: http://online.wsj.com/news/articles/SB10001424127887323463704578497153556847658?mg=reno64-wsj&url=http%3A%2F%2Fonline.wsj.com%2Farticle%2FSB10001424127887323463704578497153556847658.html

[44] (January 2014) Mobile Malware Infects Millions; LTE Spurs Growth. *Infosecurity*. Available: http://www.infosecurity-magazine.com/news/mobile-malware-infects-millions-lte-spurs-growth/

[45] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "Xmandroid: A new android evolution to mitigate privilege escalation attacks," *Technische Universität Darmstadt, Technical Report TR-2011-04*, 2011.

[46] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, "Towards Taming Privilege-Escalation Attacks on Android," in *NDSS*, 2012.

[47] O. Publishing, *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*: Organisation for Economic Co-operation and Development, 2002.

[48] U. S. D. o. H. H. Services. *HIPAA*. Available: http://www.hhs.gov/ocr/privacy/

[49] E. Commission. (2013). *Protection of personal data*. Available: http://ec.europa.eu/justice/data-protection/

[50] E. Commission, "The Data Protection Directive (Directive 95/46/EC)," 2013.

[51] S. S. Greene, *Security Policies and Procedures*: New Jersey: Pearson Education, 2006.

[52] G. Stoneburner, C. Hayden, and A. Feringa, "Engineering principles for information technology security (a baseline for achieving security)," DTIC Document2001.

[53] J. Jonsson and B. Kaliski, "Public-key cryptography standards (PKCS)# 1: RSA cryptography specifications version 2.1," 2003.

[54] T. Dierks, "The transport layer security (TLS) protocol version 1.2," 2008.

[55] N.-F. Standard, "Announcing the Advanced Encryption Standard (AES)," *Federal Information Processing Standards Publication*, vol. 197, 2001.

[56] P. Karn, W. A. Simpson, and P. Metzger, "The ESP triple DES transform," 1995.

[57] K. Kaukonen and R. Thayer, "A stream cipher encryption algorithm "arcfour"," *The Internet Society*, 1999.

[58] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*, 1985, pp. 10-18.

[59] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[60] C. Furlani, "FIPS 186-3: Digital Signature Standard (DSS)," *Online, National Institute of Standards and Technology (NIST) Std.(June 2009)*, 2009.

[61] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Advances in Cryptology—CRYPTO'96*, 1996, pp. 1-15.

[62] S. H. Standard, "FIPS Pub 180-1," *National Institute of Standards and Technology*, vol. 17, p. 15, 1995.

[63] X. Wang, Y. L. Yin, and H. Yu, "Finding collisions in the full SHA-1," in *Advances in Cryptology–CRYPTO 2005*, 2005, pp. 17-36.

[64] A. C. Weaver, "Secure Sockets Layer," *Computer*, vol. 39, pp. 88-90, 2006.

[65] K. Hickman and T. Elgamal, "The SSL protocol," *Netscape Communications Corp*, vol. 501, 1995.

[66] P. Hallam-Baker and E. Maler, "Assertions and protocol for the oasis security assertion markup language (saml)," *OASIS XML-Based Security Services Technical Committee*, 2002.

[67] S. Cantor, I. J. Kemp, N. R. Philpott, and E. Maler, "Assertions and protocols for the oasis security assertion markup language," *OASIS Standard (March 2005),* 2005.

[68] W. Redmond, "Microsoft Passport: Streamlining Commerce and Communication on the Web," *Microsoft News Center,* 1999.

[69] M. Walker. (27 Oct 2008). *Windows Live ID now supports OpenID Identity Framework.* Available: http://blogs.msdn.com/b/mikewalker/archive/2008/10/27/windows-live-id-will-support-openid-identity-framework.aspx

[70] D. Burt. (27 Aug 2009). *Windows Live ID OpenID Status Update*. Available: http://blogs.technet.com/b/privacyimperative/archive/2009/08/28/windows-live-id-openid-status-update.aspx

[71] S. S. Y. Shim, B. Geetanjali, and P. Vishnu, "Federated identity management," *Computer,* vol. 38, pp. 120-122, 2005.

[72] H. S. Al-Sinani, W. A. Alrodhan, and C. J. Mitchell, "CardSpace-Liberty integration for CardSpace users," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, 2010, pp. 12-25.

[73] S. Cantor, J. Hodges, J. Kemp, and P. Thompson, "Liberty ID-FF Architecture Overview," *Wason, Thomas (Herausgeber): Liberty Alliance Project Version,* vol. 1, 2005.

[74] J. Tourzan and Y. Koga, "Liberty id-wsf web services framework overview," *Liberty Alliance,* 2004.

[75] S. Kellomäki and R. Lockhart, "Liberty ID-SIS Employee Profile Service Specification," ed: Version, 1945.

[76] D. Recordon and D. Reed, "OpenID 2.0: a platform for user-centric identity management," in *Proceedings of the second ACM workshop on Digital identity management*, 2006, pp. 11-16.

[77] W. Huping, F. Chunxiao, Y. Shuai, Z. Junwei, and Z. Xiaoying, "A New Secure OpenID Authentication Mechanism Using One-Time Password (OTP)," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1-4.

[78] O. Hyun-Kyung and J. Seung-Hun, "The Security Limitations of SSO in OpenID," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, 2008, pp. 1608-1611.

[79] D. Recordon and B. Fitzpatrick, "OpenID Authentication 1.1," *Finalized OpenID Specification, May,* 2006.

[80] G. Alpár, J.-H. Hoepman, and J. Siljee, "The identity crisis. security, privacy and usability issues in identity management," *arXiv preprint arXiv:1101.0427*, 2011.

[81] (2013). *OAuth*. Available: http://oauth.net/about/

[82] E. Hammer-Lahav, D. Recordon, and D. Hardt, "The OAuth 2.0 authorization protocol," *Network Working Group Internet-Draft*, 2011.

[83] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *IEEE Security & Privacy*, vol. 8, pp. 24-31, 2010.

[84] Z. Peng, S. Hanlin, and Y. Zheng, "Building up Trusted Identity Management in Mobile Heterogeneous Environment," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 2011, pp. 873-877.

[85] B. En-Nasry and M. D. E.-C. El Kettani, "Towards an Open Framework for Mobile Digital Identity Management through Strong Authentication Methods," in *Secure and Trust Computing, Data Management, and Applications*, ed: Springer, 2011, pp. 56-63.

[86] V. Paruchuri and S. Chellappan, "Context Aware Identity Management Using Smart Phones," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on*, 2013, pp. 184-190.

[87] M. Kim, H. Jeong, and E. Choi, "Context-aware Platform for User Authentication in Cloud Database Computing," in *International Conference on Future Information Technology and Management Science & Engineering Lecture Notes in Information Technology*, 2012, pp. 170-176.

[88] H. Witte, C. Rathgeb, and C. Busch, "Context-Aware Mobile Biometric Authentication based on Support Vector Machines," in *Emerging Security Technologies (EST), 2013 Fourth International Conference on*, 2013, pp. 29-32.

[89] D. Huang, T. Xing, and H. Wu, "Mobile cloud computing service models: a user-centric approach," *IEEE Network*, vol. 27, pp. 6-11, 2013.

[90] D. Boger, L. Barreto, J. Fraga, P. Urien, H. Aissaoui, A. Santos, and G. Pujolle, "User-centric Identity Management based on secure elements," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, 2014, pp. 1-6.

[91] S. Sin-seok, K. Joon-Myung, H. Yoonseon, and J. W. K. Hong, "Context management for user-centric context-aware services over pervasive

networks," in *Network Operations and Management Symposium (APNOMS), 2012 14th Asia-Pacific*, 2012, pp. 1-4.

[92] M. Bourimi, J. M. Haake, M. Heupel, B. Uebersch ̈ar, D. Kesdogan, and T. Barth, "Enhancing privacy in mobile collaborative applications by enabling end-user tailoring of the distributed architecture," *International Journal for Infonomics (IJI)*, vol. 3, pp. 563-572, 2011.

[93] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Behaviour profiling on mobile devices," in *Emerging Security Technologies (EST), 2010 International Conference on*, 2010, pp. 77-82.

[94] L. Ki Jung and S. Il-Yeol, "Modeling and Analyzing User Behavior of Privacy Management on Online Social Network: Research in Progress," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, 2011, pp. 1344-1351.

[95] P. Paraskevopoulos, T. Dinh, Z. Dashdorj, T. Palpanas, and L. Serafini, "Identification and characterization of human behavior patterns from mobile phone data," in *Intl. Conf. the Analysis of Mobile Phone Datasets (NetMob 2013), Special Session on the Data for Development (D4D) Challenge*, 2013.

[96] N. Lathia, V. Pejovic, K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Smartphones for Large-Scale Behavior Change Interventions," *IEEE Pervasive Computing*, vol. 12, pp. 66-73, 2013.

[97] H. T. Tavani and J. H. Moor, "Privacy protection, control of information, and privacy-enhancing technologies," *SIGCAS Comput. Soc.*, vol. 31, pp. 6-11, 2001.

[98] A. RFC, "Internet Security Glossary," ed: Version, 2007.

[99] H. M. Levy, *Capability-based computer systems* vol. 12: Digital Press Bedford, 1984.

[100] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *Communications Magazine, IEEE*, vol. 32, pp. 40-48, 1994.

[101] C. S. Jordan, *Guide to Understanding Discretionary Access Control in Trusted Systems*: DIANE Publishing, 1987.

[102] B. W. Lampson, "Protection," *ACM SIGOPS Operating Systems Review*, vol. 8, pp. 18-24, 1974.

[103] R. N. Watson, "A decade of OS access-control extensibility," *Communications of the ACM*, vol. 56, pp. 52-63, 2013.

[104] P. Samarati and S. C. de Vimercati, "Access control: Policies, models, and mechanisms," in *Foundations of Security Analysis and Design*, ed: Springer, 2001, pp. 137-196.

[105] E. Bertino, C. Bettini, and P. Samarati, "A discretionary access control model with temporal authorizations," in *New Security Paradigms Workshop, 1994. Proceedings., 1994 ACM SIGSAC*, 1994, pp. 102-107.

[106] K. Lehmann and F. Matthes, "Meta model based integration of role-based and discretionary access control using path expressions," in *E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, 2005, pp. 443-446.

[107] L. Ninghui and M. V. Tripunitara, "On safety in discretionary access control," in *Security and Privacy, 2005 IEEE Symposium on*, 2005, pp. 96-109.

[108] T. Thomas, "A mandatory access control mechanism for the Unix file system," in *Aerospace Computer Security Applications Conference, 1988., Fourth*, 1988, pp. 173-177.

[109] W. Rjaibi and P. Bird, "A multi-purpose implementation of mandatory access control in relational database management systems," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 1010-1020.

[110] L. LI, Y.-Z. HE, and D.-G. FENG, "A fine-grained mandatory access control model for XML documents," *Journal of software*, vol. 15, pp. 1528-1537, 2004.

[111] L. Sung-Min, S. Sang-bum, J. Bokdeuk, and M. Sangdok, "A Multi-Layer Mandatory Access Control Mechanism for Mobile Devices Based on Virtualization," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, 2008, pp. 251-256.

[112] R. J. Robles, C. Min-kyu, Y. Sang-Soo, and K. Tai-hoon, "Application of Role-Based Access Control for Web Environment," in *Ubiquitous Multimedia Computing, 2008. UMC '08. International Symposium on*, 2008, pp. 171-174.

[113] H. Lindqvist, "Mandatory access control," *Master's Thesis in Computing Science, Umea University, Department of Computing Science, SE-901*, vol. 87, 2006.

[114] R. Watson, W. Morrison, C. Vance, and B. Feldman, "The TrustedBSD MAC Framework: Extensible Kernel Access Control for FreeBSD 5.0," in *USENIX Annual Technical Conference, FREENIX Track*, 2003, pp. 285-296.

[115] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," *arXiv preprint arXiv:0903.2171,* 2009.

[116] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer,* vol. 29, pp. 38-47, 1996.

[117] A. C. O'Connor and R. J. Loomis, "2010 Economic Analysis of Role-Based Access Control," *RTI International report for NIST,* 2010.

[118] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A temporal role-based access control model," *ACM Transactions on Information and System Security (TISSEC),* vol. 4, pp. 191-233, 2001.

[119] E. Coyne and T. Weil, "An RBAC implementation and interoperability standard: The INCITS cyber security 1.1 model," *Security & Privacy, IEEE,* vol. 6, pp. 84-87, 2008.

[120] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca, "GEO-RBAC: a spatially aware RBAC," presented at the Proceedings of the tenth ACM symposium on Access control models and technologies, Stockholm, Sweden, 2005.

[121] S. Oh and S. Park, "Task–role-based access control model," *Information Systems,* vol. 28, pp. 533-562, 2003.

[122] M. Strembeck and G. Neumann, "An integrated approach to engineer and enforce context constraints in RBAC environments," *ACM Trans. Inf. Syst. Secur.,* vol. 7, pp. 392-427, 2004.

[123] M. S. Kirkpatrick and E. Bertino, "Enforcing spatial constraints for mobile rbac systems," in *Proceedings of the 15th ACM symposium on Access control models and technologies,* 2010, pp. 99-108.

[124] B. Shafiq, J. B. Joshi, E. Bertino, and A. Ghafoor, "Secure interoperation in a multidomain environment employing RBAC policies," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 17, pp. 1557-1577, 2005.

[125] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 17, pp. 4-23, 2005.

[126] D. Ferraiolo, R. Kuhn, and R. Sandhu, "RBAC Standard Rationale: Comments on "A Critique of the ANSI Standard on Role-Based Access Control"," *Security & Privacy, IEEE,* vol. 5, pp. 51-53, 2007.

[127] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," presented at the Proceedings of the 13th ACM symposium on Access control models and technologies, Estes Park, CO, USA, 2008.

[128] E. Yuan and J. Tong, "Attributed based access control (ABAC) for web services," in *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*, 2005.

[129] E. Coyne and T. R. Weil, "ABAC and RBAC: Scalable, Flexible, and Auditable Access Management," *IT Professional*, vol. 15, pp. 14-16, 2013.

[130] S. Hai-bo and H. Fan, "An Attribute-Based Access Control Model for Web Services," in *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on*, 2006, pp. 74-79.

[131] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman, "A Flexible Attribute Based Access Control Method for Grid Computing," *Journal of Grid Computing*, vol. 7, pp. 169-180, 2009/06/01 2009.

[132] R. Bobba, O. Fatemieh, F. Khan, C. A. Gunter, and H. Khurana, "Using Attribute-Based Access Control to Enable Attribute-Based Messaging," in *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, 2006, pp. 403-413.

[133] L. Wang and B. Wang, "Attribute-Based Access Control Model for Web Services in Multi-Domain Environment," in *Management and Service Science (MASS), 2010 International Conference on*, 2010, pp. 1-4.

[134] S. BEJI, Y. JAMMOUSSI, and N. EL KADHI, "Towards context-awareness security for mobile applications."

[135] J. Li, I. Ari, J. Jain, A. H. Karp, and M. Dekhil, "Mobile in-store personalized services," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009, pp. 727-734.

[136] H. Ould-Slimane, M. Bande, and H. Boucheneb, "WiseShare: A collaborative environment for knowledge sharing governed by ABAC policies," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, 2012, pp. 21-29.

[137] H. Shen and Y. Cheng, "A Context-Aware Semantic-Based Access Control Model for Mobile Web Services," in *Advanced Research on Computer Science and Information Engineering*. vol. 153, G. Shen and X. Huang, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 132-139.

[138] P. Kumaraguru, L. Cranor, J. Lobo, and S. Calo, "A survey of privacy policy languages," in *Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM*, 2007.

[139] K. Bohrer and B. Holland, "Customer profile exchange (cpexchange) specification," ed, 2000.

[140] M. Schunter and C. Powers, "The enterprise privacy authorization language (epal 1.1)," *W3C Working Group*, 2003.

[141] S. Godik, T. Moses, A. Anderson, B. Parducci, C. Adams, D. Flinn, G. Brose, H. Lockhart, K. Beznosov, and M. Kudo, "extensible access control markup language (xacml) version 1.0," ed, 2003.

[142] J. Reagle and R. Wenning, "P3P and Privacy on the Web FAQ," *The World Wide Web Consortium, http://www. w3. org/P3P/P3FAQ. html,* vol. 12, 1997.

[143] P. C. K. Hung, E. Ferrari, and B. Carminati, "Towards standardized Web services privacy technologies," in *Web Services, 2004. Proceedings. IEEE International Conference on*, 2004, pp. 174-181.

[144] L. Cranor, M. Langheinrich, and M. Marchiori, "A P3P preference exchange language 1.0 (APPEL1. 0)," *W3C working draft*, vol. 15, 2002.

[145] P. P. Privacy, "An Assessment of P3P and Internet Privacy," ed: Electronic Privacy Information Center, 2000.

[146] P. A. S. H. G. Karjoth and C. P. M. S. E. Privacy, "Authorization Language (EPAL 1.1) Oct. 1, 2003 IBM Research mts at zurich. ibm. com," *Source: http://www. zurich. ibm. com/security/enterprise-privacy/epal/Specification*.

[147] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum, "Privacy and contextual integrity: framework and applications," in *Security and Privacy, 2006 IEEE Symposium on*, 2006, pp. 15 pp.-198.

[148] XAMCL and O. S. S. T. Committee, "extendible Access Control Markup Language (xacml) committee specification 2.0," ed: Feb, 2005.

[149] C. A. Ardagna, S. D. C. d. Vimercati, S. Paraboschi, E. Pedrini, and P. Samarati, "An XACML-based privacy-centered access control system," presented at the Proceedings of the first ACM workshop on Information security governance, Chicago, Illinois, USA, 2009.

[150] A. Anderson, "A comparison of two privacy policy languages: EPAL and XACML," Sun Microsystems, Inc.2005.

[151] OASIS. (2013). *OASIS eXtensible Access Control Markup Language (XACML) TC*. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[152] A. Anderson and B. Devaraj, "XACML-Based Web Services Policy Constraint Language (WS-PolicyConstraints)," *Working Draft,* vol. 6, p. 24, 2005.

[153] V. S. Y. Cheng, P. C. K. Hung, and D. K. W. Chiu, "Enabling Web Services Policy Negotiation with Privacy preserved using XACML," in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 2007, pp. 33-33.

[154] A. Anderson and H. Lockhart, "SAML 2.0 profile of XACML," *OASIS, September,* vol. 51, 2004.

[155] C. A. Ardagna, S. De Capitani di Vimercati, G. Neven, S. Paraboschi, F. S. Preiss, P. Samarati, and M. Verdicchio, "Enabling Privacy-preserving Credential-based Access Control with XACML and SAML," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 2010, pp. 1090-1095.

[156] W. Hommel, "Using XACML for Privacy Control in SAML-Based Identity Federations," in *Communications and Multimedia Security*. vol. 3677, J. Dittmann, S. Katzenbeisser, and A. Uhl, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 160-169.

[157] N. Ulltveit-Moe and V. Oleshchuk, "Mobile Security with Location-Aware Role-Based Access Control," in *Security and Privacy in Mobile Information and Communication Systems*. vol. 94, R. Prasad, K. Farkas, A. Schmidt, A. Lioy, G. Russello, and F. Luccio, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 172-183.

[158] Q. Xuebing and C. Adams, "XACML-Based Policy-Driven Access Control for Mobile Environments," in *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, 2006, pp. 643-646.

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

[160] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, ed: Springer, 2005, pp. 457-473.

[161] A. Shamir, "How to share a secret," *Communications of the ACM,* vol. 22, pp. 612-613, 1979.

[162] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," in *Advances in Cryptology—CRYPTO'88*, 1990, pp. 27-35.

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

[164] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–EUROCRYPT 2011*, ed: Springer, 2011, pp. 568-588.

[165] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*, ed: Springer, 2007, pp. 515-534.

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

[167] K. Yang and X. Jia, "Attributed-Based Access Control for Multi-authority Systems in Cloud Storage," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, 2012, pp. 536-545.

[168] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," 2013.

[169] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," presented at the Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, Chicago, Illinois, USA, 2011.

[170] H.-C. Liao and Y.-H. Chao, "A new data encryption algorithm based on the location of mobile users," *Information Technology Journal*, vol. 7, pp. 63-69, 2008.

[171] N. Chen, M. Gerla, D. Huang, and X. Hong, "Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, 2010, pp. 1-8.

[172] X. Zhiqian and K. M. Martin, "Dynamic User Revocation and Key Refreshing for Attribute-Based Encryption in Cloud Storage," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, 2012, pp. 844-849.

[173] T. Priebe, W. Dobmeier, and N. Kamprath, "Supporting attribute-based access control with ontologies," in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, 2006, p. 8 pp.

[174] R. Dhamija and L. Dusseault, "The seven flaws of identity management: Usability and security challenges," *Security & Privacy, IEEE,* vol. 6, pp. 24-29, 2008.

[175] E. McCallister, *Guide to protecting the confidentiality of personally identifiable information*: DIANE Publishing, 2010.

[176] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: validating SSL certificates in non-browser software," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 38-49.

[177] H. Jhe-Yi, S. Chien-Cheng, L. Wei-Hsiang, and C. C. Ho, "Android-based mobile payment service protected by 3-factor authentication and virtual private ad hoc networking," in *Computing, Communications and Applications Conference (ComComAp), 2012*, 2012, pp. 111-116.

[178] C. Cremers, "The Scyther tool: Automatic verification of security protocols," ed, 2009.

[179] C. J. F. Cremers, "Scyther: Semantics and verification of security protocols," vol. 68, ed, 2006.

[180] C. J. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification*, 2008, pp. 414-418.

[181] Oracle. (2013). *MySQL :: The world's most popular open source database*. Available: http://www.mysql.com/

[182] S. XACML, "Sun's XACML Implementation Programmer's Guide," ed, 2007.

[183] A. S. Foundation. (2013). *Apache Tomcat* Available: http://tomcat.apache.org/

[184] T. E. Foundation. (2013). *Eclipse - The Eclipse Foundation open source community website*. Available: http://www.eclipse.org/

[185] N. Community. (2013). *Netbeans IDE*. Available: https://netbeans.org/

[186] G. Karjoth and A. Schade, "SERIALIZATION OF XACML POLICIES," ed: Google Patents, 2009.

[187] N. Ulltveit-Moe and V. Oleshchuk, "Mobile Security with Location-Aware Role-Based Access Control," in *Security and Privacy in Mobile Information and Communication Systems*, ed: Springer, 2012, pp. 172-183.

[188] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Misuse detection for mobile devices using behaviour profiling," *International Journal of Cyber Warfare and Terrorism (IJCWT),* vol. 1, pp. 41-53, 2011.

[189] M. Miettinen, P. Halonen, and K. Hatonen, "Host-based intrusion detection for advanced mobile devices," in *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, 2006, pp. 72-76.

[190] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, pp. 255-268, 2006.

[191] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency control and recovery in database systems* vol. 370: Addison-wesley New York, 1987.

[192] G. Weikum and G. Vossen, *Transactional information systems: theory, algorithms, and the practice of concurrency control and recovery*: Elsevier, 2001.

[193] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Advances in Cryptology – EUROCRYPT 2005*. vol. 3494, R. Cramer, Ed., ed: Springer Berlin Heidelberg, 2005, pp. 457-473.

[194] Samsung. (2013). *Samsung KNOX-Solutions-Security | Samsung*. Available: http://www.samsung.com/global/business/mobile/solution/security/samsung-knox

[195] Blackberry. (2013). *Black Enterprise Server - BES - US*. Available: http://us.blackberry.com/business/software/bes/overview.html

[196] A. D. Caro. (2012). *The Java Pairing Based Cryptography Library (jPBC)*. Available: http://gas.dia.unisa.it/projects/jpbc/index.html

[197] Z. Sanaei, S. Abolfazli, A. Gani, and M. Shiraz, "SAMI: Service-based arbitrated multi-tier infrastructure for Mobile Cloud Computing," in *Communications in China Workshops (ICCC), 2012 1st IEEE International Conference on*, 2012, pp. 14-19.

Appendix

XACML Language

XACML Policy

The following polices defines that any user who holds an email address from the domain of “city.ac.uk” have the access to read resources at http://www.city.ac.uk.

```
<?xml version="1.0" encoding="UTF-8"?>
  <Policy
  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  PolicyId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIC082:policy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
  algorithm:permit-overrides">
  <Rule
  RuleId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIC082:rule"
              Effect="Permit">
  <Description>
              Anyone from city.ac.uk can perform any action on any
resource.
  </Description>
    <Target>
  <Subject>
              <SubjectMatchMatchId="urn:oasis:names:tc:xacml:1.0:function:rf
              c822Namematch">
              <AttributeValueDataType="http://www.w3.org/2001/XMLSchema
              #string">
              city.ac.uk</AttributeValue>
  <SubjectAttributeDesignator
      SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-
```

```
        category:access-subject"
AttributeId="urn:oasis:names:tc:xacml:1.0:
subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:datatype:
rfc822Name"/>
  </SubjectMatch>
  </Subject>
  <Resources>
  <Resource>
  <ResourceMatch
  MatchId="urn:oasis:names:tc:xacml:1.0:
  function:anyURI-equal">
                        <AttributeValue
                        DataType="http://www.w3.org/2001/
  XMLSchema#anyURI">
  http://www.city.ac.uk</AttributeValue>
  <ResourceAttributeDesignator
  AttributeId="urn:oasis:names:tc:xacml:1.0:
  resource:resource-id"
  DataType="http://www.w3.org/2001/
  XMLSchema#anyURI"/>
  </ResourceMatch>
  </Resource>
  </Resources>
  <Actions>
  <Action>
  <ActionMatch
  MatchId="urn:oasis:names:tc:xacml:1.0:
  function:string-equal">
                    <AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#
                    string">
```

```
                    Read</AttributeValue>
<ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:
action:action-id"
DataType="http://www.w3.org/2001/
XMLSchema#string"/>
</ActionMatch>
</Action>
</Actions>
</Target>
</Rule>
</Policy>
```

XACML Request

A user who holds an email account "Alice@city.ac.uk" requests the resources of School of Engineering and Mathematical Science at http://www.city.ac.uk.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request
xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:
context:schema:os
access_control-xacml-2.0-context-schema-os.xsd">
<Subject>
<Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:
subject:subject-id"
DataType="urn:oasis:names:tc:xacml:1.0:
data-type:rfc822Name">
<AttributeValue>Alice@city.ac.uk</AttributeValue>
</Attribute>
<AttributeAttributeId="urn:oasis:names:tc:xacml:2.0:
conformance-test:age"
DataType="http://www.w3.org/2001/XMLSchema#integer">
<AttributeValue>45</AttributeValue>
</Attribute>
</Subject>
<Resource>
<AttributeAttributeId="urn:oasis:names:tc:xacml:1.0:
resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>http://www.city.ac.uk</AttributeValue>
</Attribute>
</Resource>
```

```
<Action>
<AttributeAttributeId="urn:oasis:names:tc:xacml:1.0:
action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>read</AttributeValue>
</Attribute>
</Action>
</Request>
```

XACML Response

Regarding to the policy example and response context example, the user has an email account "Alice@CITY.AC.UK" can obtain the data consent on the resources at "http://www.city.ac.uk/". The policy evaluation decision should be *Permit*. The response context is listed as follows.

```
<Response>
<Result ResourceId="http://www.city.ac.uk/">
<Decision>Permit</Decision>
<Status>
<StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
</Status>
</Result>
</Response>
```