Securing Cloud By ID-Based Hybrid Encryption Scheme

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Engineering

By

Gautam Verma (Roll No. 2K15/SWE/10)

Under the Guidance of

Prof. (Dr.) Daya Gupta

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

2015-2017

CERTIFICATE

This is to certify that the thesis entitled “**Securing Cloud By ID-Based Hybrid Encryption Scheme**” submitted by **Gautam Verma** in partial fulfillment of the requirements for the award of degree Master of Technology in Software Engineering, is an authentic work carried out by him under my guidance. The content embodied in this thesis has not been submitted by him earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Date:

Prof. (Dr.) Daya Gupta
Department of Computer
Science and Engineering
Delhi Technological University

DECLARATION

I hereby declare that the thesis entitled "**Securing Cloud By ID-Based Hybrid Encryption Scheme**" which is being submitted to Delhi Technological University, in partial fulfillment of requirements for the award of degree of Master of Technology (Software Engineering) is an authentic work carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Gautam Verma

2K15/SWE/10

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation and gratitude to all those who have helped me directly or indirectly towards the successful completion of this work.

Firstly, I would like to express my sincere gratitude to my guide **Prof. (Dr.) Daya Gupta**, **Department of Computer Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, encouragement, constant support and valuable inputs were always there for me throughout the course of my work. Without her continuous support and interest, this thesis would not have been the same as presented here.

Also I would like to extend my thanks to **Mrs. Shruti Jaiswal** (Research Scholar, Delhi Technological University) for her critical review and support, and the entire staff in the Department of Software Engineering, DTU for their help during my course of work.

Last but not the least I would like to express my sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Gautam Verma

2K15/SWE/10

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AWS	Amazon Web Services
IBE	Identity Based Encryption
PKE	Public Key Encryption
ABE	Attribute Based Encryption
ID	Identity
CP-ABE	Ciphertext Policy Attribute Based Encryption
KP-ABE	Key Policy Attribute Based Encryption
PKG	Private Key Generator
HIBE	Hierarchical Identity Based Encryption
CSP	Cloud Service Provider
KU-CSP	Key Update Cloud Service Provider
NIST	National Institute of Standards and Technology
Iaas	Infrastructure as a service
Saas	Software as a service
Paas	Platform as a service
CA	Certification Authority
AES	Advanced Encryption Standard
MS	Master Secret
RL	Revocation List
TL	Time List
BGK	Boldyreva Goyal Kumar

ABSTRACT

Cloud computing has seen tremendous growth in the last decade. It allows end users to share their data with each other easily. Cloud computing comes with numerous possibilities and challenges simultaneously. With increasing cloud capabilities, security has become a major challenge to the cloud. Can users trust cloud completely? Is their data secure on cloud? These questions are surfacing with no trustworthy solutions yet.

Multi user data sharing should be secure and integrity should be achieved on cloud. The methods like IBE (Identity Based Encryption), ABE (Attribute Based Encryption) etc. are widely used in cloud computing environment to achieve the data security. However, the problems associated with IBE are extra overhead on private key generator (PKG) for computations required during the user revocation process.

The two main research problems related to using IBE for cloud computing security are efficient revocation scheme and security enhancement. The goal of this research is to introduce a scheme to address both of these problems. In this thesis a novel hybrid Identity Based scheme is presented with an end goal to manage both security enhancement and efficient revocation. This hybrid technique is a mix of two widely used security methods- IBE and ABE. The Attribute Based Encryption method is blended with Identity Based Encryption to realize robust security against different threats. Another issue with efficient revocation is tended to by introducing outsourced computation into hybrid IBE method with server aided settings.

Keywords: Cloud computing, Identity Based Encryption, Attribute Based Encryption, Private Key Generator, Revocation

CHAPTER 1

INTRODUCTION

This chapter briefly describe cloud computing and the various security issues. Section 1.1 gives an overview of cloud computing, its benefits and scope. Section 1.2 briefly explains security issues in cloud and security mechanism used to take care of these issues. Section 1.3 discusses the work done by different researchers to secure cloud using IBE and ABE technique. Section 1.4 discusses motivation for this research. Section 1.5 formulates the problem statement. Section 1.6 discusses the scope of the research work. Section 1.7 presents an outline of this thesis and labeling the remaining chapters.

1.1. Overview

Cloud computing offers the ability to utilize storage and computing resources on a pay as per usage basis and lower the expenditure of an organization/business on setting up infrastructure and computing power. The omission and creation of hypervisors' controlled virtual machines executing on actual machine is a cost-effective and adaptable computing model.

Cloud computing is moving computation from the actual hardware and locally controlled software empowered platforms to virtualized cloud hosted services. Cloud providers like Microsoft Azure, Rackspace, Amazon Web Services (AWS), GoGrid, and so on, offer customers the choice to install their applications upon a collection of virtually unlimited resources with essentially no financial expenditure. It is the elasticity, cost effectiveness, and large availability of resources that force, motivate, and encourage companies to shift from enterprise applications to cloud computing.

It is a business and an economic model. It is the next step in the growth of the Internet. Cloud computing has spawned start-ups in a variety of business domains. It has forced the present corporations to adjust and familiarize rapidly to last in such an innovative atmosphere. It comprises a set of methods that can assist organizations in swift and efficient increase and decrease of resources in nearly real time.

Additionally, the incorporation of extensively accessible enormous amounts of preprocessed information such as medical history of patients can be of great assistance to professionals and researchers. Nevertheless, the maximum capability of the cloud cannot be realized without considering its potential, benefits, compromises and vulnerabilities.

However, cloud computing is in its infancy stage. The term cloud in cloud computing refers to the ways using which everything from computation power to computation infrastructure, business processes, applications, and private association—will be provided to you as a service anywhere and at any time you need it. A cloud is a group of interconnected network servers or PCs that may be private or public. The data and the applications served by the cloud are accessible to a group of users throughout the network. Yet, the cloud infrastructure and technology are not visible to the end users. Cloud services comprise the software, storage, and infrastructure delivered on top of the Internet as per the demands of the end users. The cloud accumulates huge networks of virtualized services. These include hardware services like storage and network, computing services, and infrastructure services such as databases, web servers, monitoring systems, and message queuing systems. Cloud computing is fluid in that it can expand and contract depending on the customer/business needs. From this viewpoint, the users can add or remove resources according to their needs. This quality makes cloud computing an elastic system, which can operate either manually or using automated tools.

In recent surveys conducted by different organizations, the following predictions were offered:

- Gartner Research, 2014, observed that cloud computing would be a $150 billion business.
- AMI Partners predicts SMEs (Small and Medium-sized Enterprises) are likely to expend more than 100 billion dollars on cloud computing.
- IDC recently predicted that spending on public cloud-hosted applications would exapand from 16.5 billion dollars to over 55 billion dollars in 2014.

- Software companies are migrating to the cloud now to reap the ultimate benefits of cloud computing.
- Recently McKinsey and Co. stated "clouds are hardware centered services presenting network, computing and storage capability where hardware administration is extremely abstracted from the customer, customers have to bear infrastructure expenses as variable operating expense and infrastructure capability is extremely flexible."
- The University of California, Berkeley, presented a report and suggested that the key features of cloud computing are as follows:
  - The delusion of unlimited computation resources.
  - The removal of direct obligation by cloud customers.
  - The facility to pay as per usage as and when needed.

1.2. Basic Concepts

1.2.1. Security Issues

Cloud computing has many advantages but as an outcome of its geographical distribution , size, and structure it also lead to apprehensions about the privacy and security of data present on it. The following issues are involved with these concerns:

- Cloud service provider's (CSPs) incompetency to properly manage and guard sensitive data
- Delivery of decisive and sensitive information to government agencies or law enforcement without the knowledge and/or consent of the user
- Unauthorized access and leakage of data among virtual machines functioning on the same machine
- To allow a customer to effortlessly move applications among various CSPs and evade "lock-in" a high level of interoperability is offered.
- Ability to meet controlling and conformity requirements
- The strength of the security measures established by the CSP.
- Hackers breaching into customer applications facilitated on the cloud and obtaining and sharing confidential data.

- System malfunctions and breakdowns that make the cloud facility inaccessible for longer duration of time.

Cloud clients ought to be apprehensive about the continuous accessibility of their information over prolonged duration and whether a CSP can sneakily misuse critical information for its own benefit. An approach that can be used to mitigate concerns about protecting data on cloud is encryption. Data encryption can shield it from hackers or from expose by the CSP, but it is problematic to search or make computations on that data. The below sections give a brief description of the two encryption techniques that can be used to help secure cloud storage.

1.2.2. Identity Based Encryption (IBE)

IBE is a PKE technique which permits a user to determine a public key from a random string. We usually think of this string as representing an identity of some kind, but it is usually useful to use more than just an identity to calculate such a public key. For example, to prevent a user holding the unchanged IBE key indefinitely, it is convenient to incorporate some information in this string regarding the validity period of the key. Or, to ensure that a user will receive different keys from different IBE systems, it may be useful to include information in this string that is unique to a particular IBE implementation, perhaps a URL that identifies a server that is used in the implementation of each of the different IBE systems. Because the string used to calculate a key almost every time contains beyond just an identity, it will be more accurate to use the phrase Identifier Based Encryption in its place, but this phrase is not widely employed to describe the technology. The facility to compute keys as required gives Identity Based Encryption systems diverse properties than those of conventional public key systems, and these properties deliver substantial realistic benefits in certain conditions. So even though there are possibly rare circumstances in which it is not possible to crack any challenge with conventional public key techniques which can be cracked with Identity Based Encryption, systems which make use of Identity Based Encryption are far less costly to maintain and easier to implement in comparison to other possibilities.

1.2.3. Attribute Based Encryption (ABE)

The development of ABE can be backtracked to IBE. In IBE, an ID or identity is represented as a string which is one-to-one mapped to each user. A user can obtain a private key computed from his/her identity from trusted agency in an offline mode and his/her identity is treated as public key. The encryption is one-to-one i.e. the encrypted secret text under a specific identity can be decrypted by a user only if he owns the private key corresponding to the public key.

ABE was first recommended as a fuzzy Identity Based Encryption, with a set of expressive attributes used to represent the identity. The private key for an ID w can be used to decrypt the secret message encrypted by the ID w' if w and w' are nearby than a predefined threshold in terms of distance metric - set overlap. The threshold based set overlap distance metric to descriptive access policies with OR and AND gates is summed up.

The two major types of ABE are Ciphertext Policy Attribute Based Encryption (CP-ABE) and Key Policy Attribute Based Encryption (KP-ABE). The fundamental distinction between attribute and identity is attributes are many to many mapped to users while identities are many to one mapped to users.

A more general model is called Predicate Encryption. These systems have the upsides of offering a level of “anonymity” by concealing the associated access structures themselves. The down side is the Predicate Encryption is not so expressive compared with ABE.

1.3. Related Work

In 2001, Dan Boneh and Matthew Franklin [21] offered the first IBE scheme from the Weil-pairing and suggested a simple method of revoking users in which each non revoked user obtains a new private key created by the PKG periodically. A period can be fixed as a single day, a week, a month, or a year. A sender uses a designated receiver’s identity and present time to encrypt messages while the designated receiver uses the current private key to decrypt the cipher text. Hence, it is necessary for the users to update their private-keys periodically. To remove/revoke a user, the Private Key Generator simply ends delivering the fresh private-key for the user. It is obvious that a secure channel need to be set up between the Private Key Generator and each user to transfer the new private-key. Be that as it may, in this method PKG is highly overloaded. In other words, every user irrespective of whether its key has been revoked or not, has to contact with PKG intermittently to verify its identity and update its private-key. It expects the PKG to be online and a secured channel should be provided for each exchange, which may prove to be a hindrance for Identity Based Encryption system as the no. of users increases.

In order to alleviate the load of the PKG in Boneh and Franklin's scheme, Boneh *et al*. [22] proposed another revocation method, called immediate revocation. Immediate revocation approach uses an appointed semi-trusted and online agency (i.e. mediator) to lessen the management load of the Private Key Generator and support users to decrypt secret message [23], [24], [25], [26]. In such a case, the online mediator must hold shares of all the users' private keys. Since the decryption operation must involve both parties, neither the user nor the online mediator can cheat one another. When a user was revoked/removed, the online agency is ordered to discontinue assisting the user. However, the online mediator must help users to decrypt each ciphertext so that it becomes a bottleneck for such schemes as the number of users grows enormously.

On the other hand, in revocation method of Dan Boneh and Matthew Franklin [21], all the users must periodically update new private keys sent by the PKG. As the no. of users increses, the load of updating the key becomes a hindrance for the PKG. In 2008, Boldyreva et al. [18] suggested a revocable IBE scheme to improve key update efficiency. Their revocable IBE scheme is constructed on the concept of Fuzzy IBE [20] and adopts the complete subtree method to record identities of users at leaf nodes. It decreased the no. of key-updates operations from linear to logarithmic in terms of no. of users. Indeed, by representing users as binary tree data structure, the scheme efficiently alleviates the key-update load of the PKG. Furthermore, B. Libert and D. Vergnaud [17] enhanced the security of Boldyreva et al.'s revocable IBE scheme by presenting an adaptive-identity secure scheme. Nevertheless, Boldyreva et al.'s scheme still results in several problems: (1) The private key size of each user is 3logc points in an elliptic curve, where c is the no. of users/leaf-nodes in the binary tree. (2) It also results in massive computation load for decryption and encryption processes. (3) It is enormous load for PKG to maintain the binary tree with a large amount of users.

JH Seo and K Emura [14] improved the security model of Boldyreva et al.'s revocable IBE scheme [18] by considering a new risk, called decryption key exposure attacks. Based on the idea of Libert and Vergnaud's scheme [17], they also proposed a revocable IBE system which is resistant to decryption key exposure attack. In order to reduce sizes of both private-keys and update keys, Park et al. [9] suggested a new revocable IBE scheme by operating on multilinear maps, but the public parameters size is dependent on the no. of users. For achieving constant size public parameters, Wang et al. [12] employed both the dual system encryption methodology [27] and the complete subtree method [18] to propose a new revocable IBE scheme.

Moreover, JH Seo and K Emura [13] extended the concept of revocable IBE scheme to propose the first revocable HIBE scheme. In Seo and Emura's system, each user generates a secret key for each period by multiplication of some of the partial keys, which is dependent on the partial keys applied by predecessors in the hierarchy tree. In such a case, the secret key size of each user increases quadratically in the hierarchy tree where a user at low level should be aware of the record of key-updates accomplished by predecessors in the current time period, and it makes the scheme very complicated. In 2015, Seo and Emura [8] proposed a new method to develop a novel revocable HIBE scheme with history-free updates. Nevertheless, the mentioned revocable IBE and HIBE schemes above [14], [9], [12], [13], [8] employed the complete subtree method to decrease the number of operations of key updates from linear to logarithmic in terms of no. of users. However, these schemes also suffered from the same disadvantages of Boldyreva et al.'s revocable IBE scheme [18] and still used a secured channel to communicate private-keys periodically.

In 2012, Tseng and Tsai [16] proposed a new revocable IBE scheme to remove the usage of secured channel between each user and the agency and use a public channel instead to transfer users' private keys. They partitioned private-key of users into two elements, namely, an identity key and a time update key. The identity component is a secret-key connected with user's identity, which is delivered to the user via a secured channel and remains fixed forever. The time-update key is a key associated with user's identity and time-period, which is changed with time. The Private Key Generator periodically creates current time update keys for non-revoked users and transmits them via a public channel to these users. A user is allowed to decipher the ciphertext if he/she possesses both the identity key and the legitimate time-update key. In other words, to revoke a particular user, the PKG merely stops supplying new time update key to the user. However, key update efficiency is linear in terms of no. of users so the computation burden of PKG is still enormous.

In 2015, Li et al. [11] a cloud-aided service provider, added an outsourced computation method into IBE to suggest a revocable IBE system with a KU-CSP. They shifted the key update procedure to a KU-Cloud Service Provider to reduce workload of Private Key Generator. Li et al. also adopted the similar technique used in Tseng and Tsai's proposed scheme [16], which partitions private-key of a user into a time-update key and an identity-key. The Private Key Generator transmits a user the corresponding identity key via a secured channel. In the meantime, the Private Key Generator need to create a random time key for all user and transmit it to the Key Update CSP. Then the KU-CSP creates the current time update key of a user by using the associated time key and transmits it via a public channel to the user. To revoke a user, the Private Key Generator asks the KU-CSP to stop distributing the new time-update key of the user. However, their scheme has two shortcomings. One is that the transmission and computation costs are greater than earlier revocable Identity Based Encryption schemes [21], [16]. The other limitation is that it is not scalable in the sense that the Key Update CSP need to keep a time key for each user so that it will experience the management load.

The table 1.1 below summarized the work done in the field of IBE and ABE

Table 1.1: Summary of Related Work

S.NO	Title	Author	Year	Proposed Technique	Remarks
1	Identity-based encryption with outsourced revocation in cloud computing	Li et al.	2015	Outsource the key update operations to KU-CSP to offload PKG.	Not scalable, High transmission and computation costs.
2	Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption	Seo and Emura	2015	Revocable Hierarchical IBE (HIBE) scheme with history free updates	Require secure channel to communicate private keys
3	New Constructions of Revocable Identity-	Park et al.	2015	A new revocable IBE scheme by operating on multilinear maps	Public parameter size is dependent on no. of users

	Based Encryption from Multilinear Maps				
4	An Efficient and Provable Secure Revocable Identity-Based Encryption Scheme	Wang et al.	2014	New Revocable IBE with constant public parameters size	A little less secure scheme.
5	Efficient Delegation of Key Generation and Revocation Functionalities in Identity-Based Encryption	Seo and Emura	2013	Revocable HIBE scheme with logarithm key update efficiency	Require secure channel to communicate private keys, key updates are history dependent
6	Revocable Identity-Based Encryption Revisited: Security Model and Construction	Seo and Emura	2013	A revocable IBE scheme resilient to decryption key exposure attack.	Difficult to implement due to its complexity.
7	Attribute-Based Encryption with Fast Decryption	Hohenberger and Waters	2013	A KP-ABE scheme where decryption can be performed with constant no. of pairing	Long key size and slow decryption
8	Efficient revocable ID-based encryption with a public channel	Tseng and Tsai	2012	Removed the need of safe channel between PKG and user	Key update efficiency is linear in no. of users
9	Adaptive-ID Secure Revocable Identity-Based Encryption	Libert and Vergnaud	2009	Enhanced Boldyreva's scheme with Adaptive-identity	Costly encryption - decryption process, high load at PKG
10	Identity-based encryption with efficient revocation	Boldyreva et al.	2008	A revocable scheme with efficient key update	Long key size, costlier encryption/decryption process, high load at PKG
11	Attribute-based encryption for fine-grained access control of encrypted data	Goyal et al.	2006	A scheme to enable fine-grained access of data	Ciphertext size decryption time is proportional to no. of attribute
12	Identity-Based Encryption from the Weil Pairing	Boneh et al.	2001	First practical IBE scheme	High load at PKG

1.4. Motivation

Cloud computing comes with numerous possibilities and challenges simultaneously. With increasing cloud capabilities, security has become a major challenge to the cloud. Can users trust cloud completely? Is their data secure on cloud? These questions are surfacing with no trustworthy solutions yet. Now a day, cloud has become primarily attractive to cyber crooks. The security threats faced by cloud are both internal and external like software bugs, malicious software, administrator mistakes, media crashes and malicious insiders.

Security is deemed to be a significant barrier for cloud computing in its road to success. The security challenges faced by cloud computing approach are dynamic and vast to some extent. Security and reliability are the two major concerns about cloud storage. Clients are unlikely to hand over their information to another company without an assurance that they'll be able to access their data at any time they want and no one else will be able to access it.

Businesses that are shifting from the conventional standalone environment to the cloud are having serious apprehension about the cloud security. The CSP has to offer more information security measures to the cloud consumer to build up the confidence. With day to day advancement in cloud technology encryption techniques associated with it should also be improving progressively. Understanding the underlying principle of cloud data encryption is the basis to understand the security system of the cloud. To enhance the security level a number of cryptographic techniques have been used. This thesis presents a hybrid ID based encryption scheme that combines IBE and ABE techniques to enhance the data and information security.

1.5. Problem Statement

Since cloud computing is utility available on web, so a number of issues are raised like user privacy, data theft, unauthorized accesses and data leakage. The user’s data security is primary obligation of cloud service provider. So, for effective security of data we require a system that offers both data encryption as well as secure defense against data theft. Researchers have proposed a variety of mechanisms to secure data in cloud environment. A number of researchers have concentrated on the detail that normally user has to get bulk of information from the cloud in a protected way. But they have not given much importance to the complexity of the cryptographic technique used. The complexity of the algorithm directly impact the speed at which data is accessed. We need some techniques that will help in swift and efficient data access in a secured manner.

Encryption is often offered as the key for addressing confidentiality threats within the cloud. When a cloud service stores data in an encrypted format it is essential to know which party between the mediator or the CSP is responsible for administration of the encryption-keys. It is significant to observe that if the cloud service provider has access to, or administrate, the encryption keys then they will be able to access and decrypt the information stored at provider’s location. The party that administrate the encryption-keys should have a sound key-management plan. Key-management is crucial to guarantee that encryption-keys are prevented from being compromised, which would cause unauthorized access of the information or denial of access to organization.

The natural risk whenever sensitive data is transmitted over a network is interception of data in transit, especially when the network is not managed or owned by the organization such as the Internet. Organization/User must confirm that all the sensitive data in transit including authentication credentials is the encrypted by the cloud provider with only globally accepted encryption tools and algorithms.

Data theft attack or unauthorized access to sensitive information by the cloud service provider’s employees is a big concern for organizations intending to make use of cloud services. The methods needed to manage such risks are no way different from those used to mitigate malicious insiders within the organization or a conventional outsource service provider.

The methods like IBE, ABE etc. have been widely used in cloud computing environment to achieve data security. However, the problems associated with IBE are extra overhead on PKG for computations required during the user revocation process. The recent studies indicate that there are two main research problems related to using IBE for cloud computing security, these are efficient revocation scheme and security enhancement. The goal of this research is to introduce a scheme to address both of these problems. In this thesis a novel hybrid Identity Based scheme is presented with an end goal to manage both security enhancement and efficient revocation. This hybrid technique is a mix of two widely used security methods- IBE and ABE. The Attribute Based Encryption method is blended with Identity Based Encryption to realize robust security against different threats. The user's ID along with his/her attributes like profession or nationality are utilized for procedure of decryption, encryption and revocation. Another issue with efficient revocation is tended to by introducing outsourced computation into hybrid IBE method with server aided settings.

Hence the problem of this thesis can be stated as:

"Design a scheme to Secure Cloud storage which can effectively protect the system from the data theft attacks expected on it and enable users to share data among peers in a secure and efficient manner."

1.6. Scope of the Work

IBE (Identity-Based Encryption) is one of the best approach for public key encryption which is presented basically for simplifying the process of key administration in certificate centered PKI (public key infrastructure). This can be done by using as public keys the human intelligible parameters such as e-mail address, distinct name, IP address etc. After that, public key & the digital certificate need not to be investigated by the sender to investigate, he/she can directly encrypt the data using recipient's ID. The PKG allocates the recipient its private-key corresponding to its ID to decipher the secret message. Although Identity Based Encryption allows a random string to be used as the public key which is considered an attractive advantage over Public Key Infrastructure, it requires a revocation scheme. In general, if private keys of many users got traded off, a plan to renounce such users from the system should be in place. In Public Key Infrastructure system, a user is revoked by adding validity duration to digital certificates or using combinations of involved techniques. Never the less, the given burdensome administration of digital certificates is specifically the problem that IBE tries to reduce.

A novel performing model is needed for producing cloud services with efficient Identity Based Encryption revocation in order to resolve the issues of storage and efficiency revocation. An immature method would be to merely deliver the master key of PKG to the CSPs. From that point onwards Cloud Service Provider can easily modify users' private keys and transfer them to the unrevoked users. This simple approach is centered on a doubtful hypothesis that the Cloud Service Provider is completely trustworthy and permitted to retrieve the master key for Identity Based Encryption system. In contrast, public clouds are typically exterior to the trusted area of the clients & are curious for clients' secrecy. So designing a secure and efficient revocable IBE system to lessen the calculation load at Private Key Generator with the untrusted Cloud Service Provider is a challenging problem.

In this research, a hybrid scheme is proposed to mitigate the current research challenges associated with cloud security. To further increase the security of IBE, the properties of ABE method is combined with efficient revocable IBE. In this approach, the operations such as key generation, key transmission and key-update is handled by the KU-CSP leaving only a fixed number of easy computation steps for Private Key Generator and sender/receiver to carry out.

Hence scope of our work can be summarized as:

- To mitigate effects of data theft attacks from malicious insider.
- To protect cloud user's data confidentiality.
- To provide a mean to revoke a user with compromised private key from the system.
- To outsource all the key generation functions to KU-CSP to reduce overhead load at PKG.
- To provide experimental findings to demonstrate the effectiveness of proposed technique.

1.7. Organization of Thesis

The thesis is further organized as follows:

Chapter 2: This chapter describes cloud computing and its different types of service and deployment models.

Chapter 3: This chapter presents the concepts of IBE and ABE and their preliminaries.

Chapter 4: This chapter presents the detailed description of our proposed work.

Chapter 5: This chapter presents the implementation details of our proposed work.

Chapter 6: This chapter presents the evaluation of the proposed hybrid technique and its comparison against the existing work of IBE with revocation.

Chapter 7: This chapter concludes the thesis and present the possible improvements in this research work in future.

Chapter 8: This chapter deals with publications from this research work.

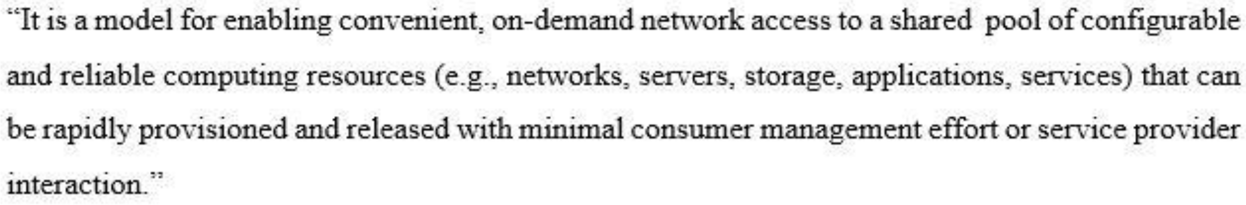
CHAPTER 2

CLOUD COMPUTING

This chapter explains the cloud computing model, its characteristics, service models and deployment models in detail.

2.1. Cloud Computing Model

Definition of cloud computing as given by NIST is as follows:



"It is a model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction."

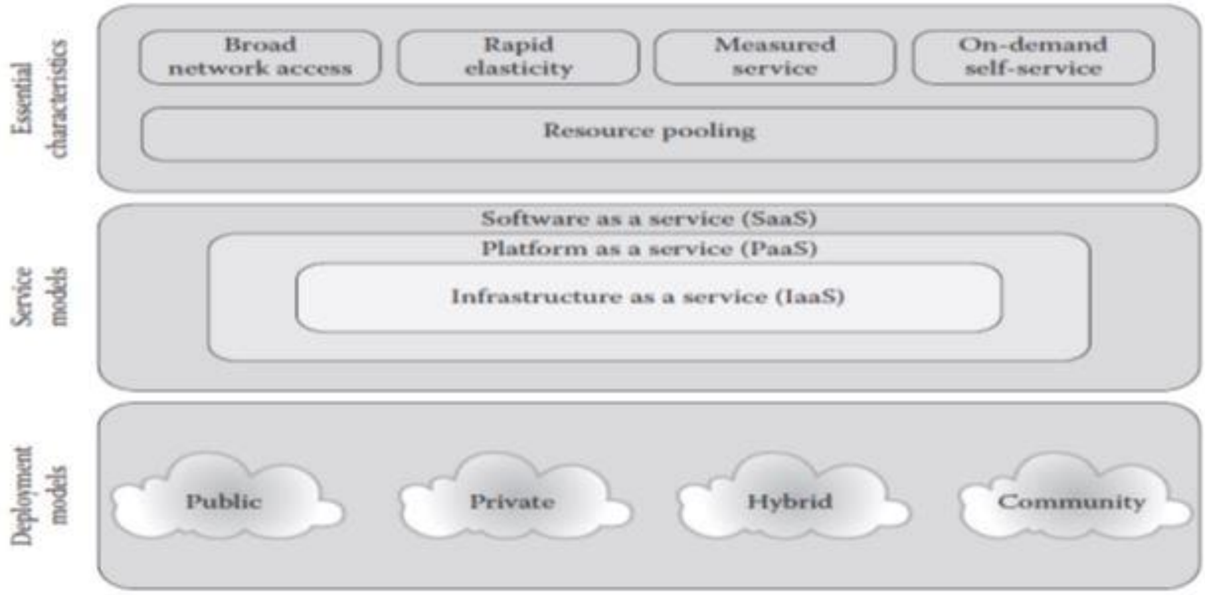


Figure 2.1: Cloud computing elements.

The elements of cloud computing are shown in Figure 2.1. This model consist of five vital characteristics, four deployment models and three services models.

2.2. Characteristics

NIST have made attempts to provide a combined way to define cloud computing and its main functionality. Notwithstanding its complexity and heterogeneous nature, NIST has recognized five vital features that signify the platform:

- **On-demand self-service***:* Vendors of cloud computing offers providing of its resources on request whenever they are wanted by the customers. Providing its resources is a crucial feature of it, as this allows customers to gauge the required infrastructure up to a substantial level without disrupting the operations by host.

- **Broad network access:** Resources of the cloud computing paradigm can be accessed and provisioned over basic network connection and for multiple device types.

- **Resource pooling***: For* using effectively and efficiently resources are pooled. Multiple users could be served by the identical physical hardware through multi-tenancy and virtualization techniques,

- **Rapid elasticity***:* Resource are elastic, to the degree that they may be increased or decreased as per its requirement in real time. Resource allocation can be done as per the customer requirement of more or less server or storages. At its core, cloud elasticity involves continual reconfiguration in its network and related controls from the internet. NIST distinguishes two types of scaling options: horizontal and vertical, which involve launching additional services and/or resources, and changing the computing capacity of assigned resources, respectively.

- **Vertical scaling:** Vertical scaling involves changing the computing capacity assigned to resources while keeping the number of physical machines constant.

2.3. Cloud Computing Service Models

Resources are heterogeneous, which varies from data storage to software services, hardware infrastructure to operating systems. Based on different kinds of graininess in facility, service models are classified into three categories:

- Infrastructure-as-a-service (IaaS),
- Software-as-a-service (SaaS)
- Platform-as-a-service (PaaS)

Cloud consumers will access cloud resources via cloud client applications that can be installed in a wide range of premises (organizations buildings) and devices (laptops, tablets, desktops and smartphones). These three models are explained in the following subsections.

2.3.1. IaaS

Raw IT resources of the model such as IP addresses, hardware, storage and firewalls are provided to the cloud customers on the web. Hypervisors run a set of virtual machines such as VirtualBox, Xen, KVM, Oracle, VMware, ESX or Hyper-V, on actual IT assets and give virtualized forms of these assets to cloud users. Cloud users have the liberty to install any environment and software they need on any platform, and experience awesome flexibility in overseeing these assets and administering their reliability and security. **Examples** of cloud providers for IaaS include Windows Azure, AWS, IBM SmartCloud Enterprise, Rackspace Open Cloud, and Google Compute Engine.

2.3.2. PaaS

For cloud consumers who want a greater level of computing and administration outsourcing, cloud providers also offer ready-to-use platforms as a service. A complete virtual environment with an operating system image installed can be used. Development platforms, different web servers, and databases are also provided. Having acquired a precise platform, consumers are free to install and administer applications on the virtual setting. The level of governance and control over the system also decreases, as the CSP installs, administers, and fixes the platform. Security at hardware and OS level is completely subjected to the CSP policies and mechanisms.

2.3.3. SaaS

The minutely grained delivery model is when cloud consumers access third-party software via the Internet. Free access can be granted (e.g., Google Docs) or via subscription models (e.g., SmugMug for managing the pictures or DropBox for file synchronization. The consumer have little control over the way the software runs on the cloud and the data's security it accesses. All the administrative burden is borne by the cloud software provider.

2.3.4. Other Services

Apart from above services, the additional cloud service categories are as follows:

- **Communications as a service (CaaS):** To optimize business processes, real-time interaction and collaboration services is integrated. It provides a unified interface and consists of customer experiences across multiple devices. Video teleconferencing, voice over IP, web conferencing, and instant messaging are some of the examples of services.
- **CompaaS:** The arrangement and utilization of handling resources needed to deploy and run software. CompaaS might be thought of as a simplified IaaS, the emphasis being on giving compute capacity.
- **DSaaS:** The arrangement and utilization of information stockpiling and related abilities. DSaaS depicts a storage model where the user leases storage from the dealer. User's information are transmitted to the CSP by means of the Internet and the user at that point uses software given by the CSP to get back information. The product is utilized to perform regular tasks related to capacity, for instance, information backup and information exchanges.
- **NaaS:** Transport availability administrations and additionally intercloud organize network management. It comprises the enhancement of asset assignments by taking into consideration processing assets and the system as a single enity. NaaS can incorporate adaptable and developed virtual private system (VPN), transfer speed on request, custom steering, multicast conventions, security firewall, interruption identification and anticipation, wide range arrange (WAN), content observing and sifting, and antivirus.

2.4. Deployment Models

The deployment of cloud services might vary according to the ownership of the service, the size of the cloud resources, and the restrictions to client access. There are basically four models:

2.4.1. Public Cloud

These (Figure 2.2) are owned by third parties, which commercialize cloud resources to the general public. Everything works as if the organization delegated the service of providing IT assets, environments, and software to a third party. In this several different organizations or entities might share an actual asset, like memory, via virtualization and multi-tenancy. Security is stimulating because cloud consumers rely on the CSP to provide assured isolation of computation and data between varied set of clients. Examples of public cloud providers include Google, Amazon, Microsoft and AWS.

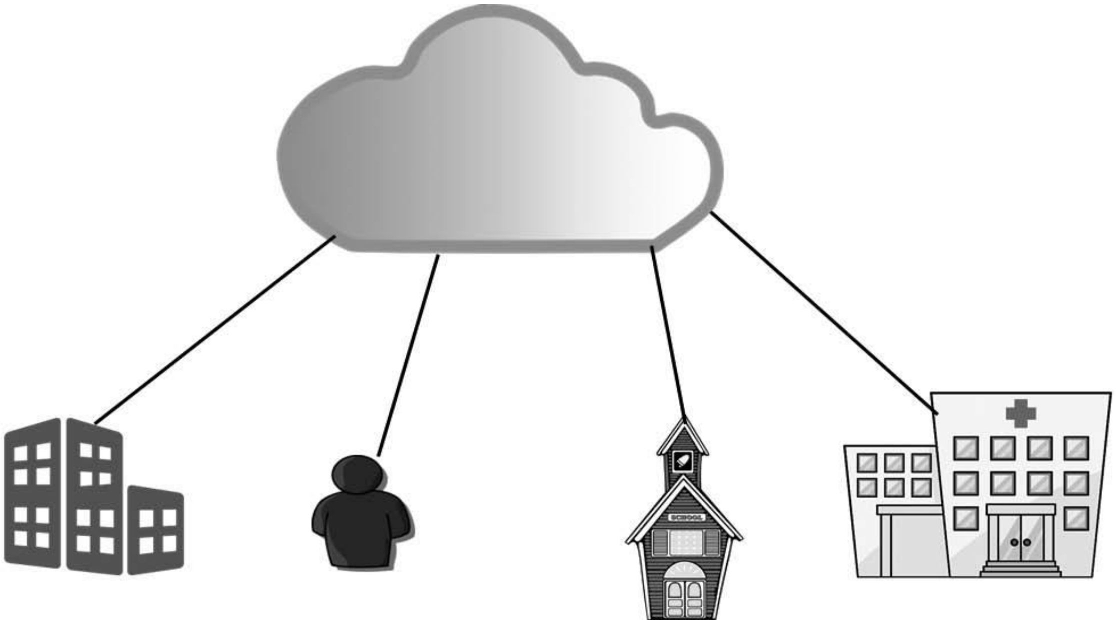


Figure 2.2: A public cloud is accessible to the general public.

2.4.2. Private Cloud

It (Figure 2.3) is owned by an organizational body, located near the premises, and offers a collection of different IT resources to various departments or anybody of the organization. It centralizes IT resources within a usually large organization so that its various parts experience all the advantages of cloud computing: elasticity, on-demand self-service, and scaling. The organization serves as cloud provider and a cloud consumer at the same time. On being a cloud provider, the organization assumes every costs of capability planning for the IT assets, reliability and security assurances, and the burden of resource administration. It increases control level and security level of the organizational resources as it can control and enforce their own safety policies and mechanisms.

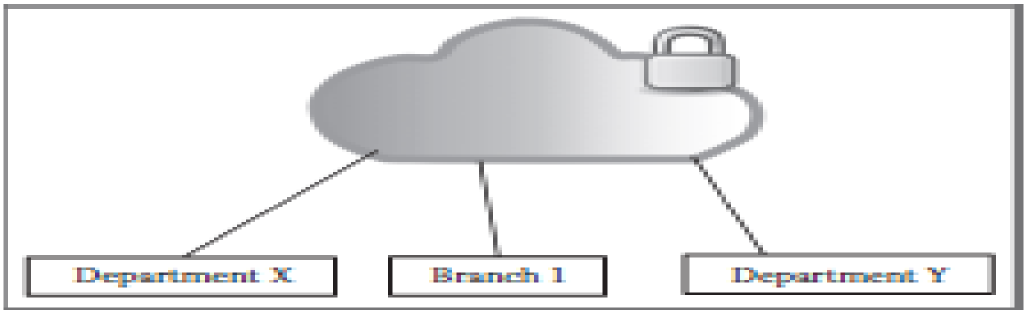


Figure 2.3: A private cloud is generally owned by an organization.

2.4.3. Community Cloud

It shares the components of both private and public cloud. As a private cloud, it has confined access and the cloud assets are shared among various free associations as public cloud. The authoritative body which share the group cloud have same prerequisites and, ordinarily, a need of trading information with each other. One of the cases of the business which is employing the community cloud idea is the healthcare industry. It might be executed to agree to governments approach and different directions. The members can trade data in a controlled way.

The cloud foundation might be overseen by a third party or participating groups and might be present on/off the premises. In this arrangement display, the expenses are distributed over less clients than a public cloud (yet in excess of a private cloud), so just a portion of the cost savings possible by cloud computing is achieved.

2.4.4. Hybrid Cloud

This crossover model is a course of action of at least two clouds (community, private, or public) that stay special but are bound together by institutionalized or selective innovation that empowers information and application conservativeness. This half breed open/private cloud arrangement might be especially helpful for littler organizations. Various applications for which security concerns are less of an issue can be offloaded at broad cost reserves without presenting the organization to moving more fragile data and applications to individuals out in the public cloud. Figure 2.4 shows hybrid cloud as a crossover of public and private cloud.

As for example, the organizations can have a private cloud to hoard sensitive intellectual property data but could make use of public cloud services to lease servers for executing performance intensive tasks or just because the private cloud runs at peak capacity. The association needs to utilize some protected convention for communications between these cloud environments. For instance, there ought to be someone to regulate network traffic between these cloud environments and access regulator for communications between virtual machines of these environments.

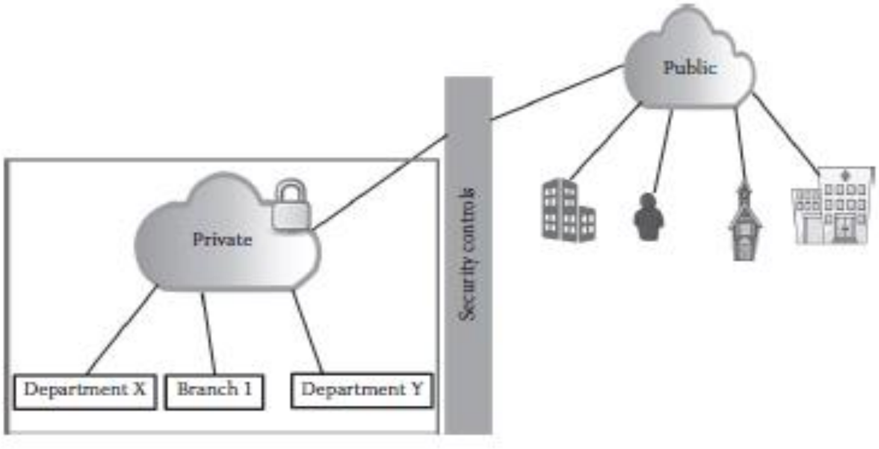


Figure 2.4: A hybrid cloud

CHAPTER 3

IDENTITY BASED AND ATTRIBUTE BASED ENCRYPTION

This chapter describes the basic mathematical constructs that are needed to build an IBE system. Section 3.2 explains the IBE scheme in detail. Section 3.3 describes the assumptions made in designing a cryptography scheme. Section 3.4 explains the Boneh-Franklin IBE scheme. Section 3.5 explains the ABE scheme.

3.1. Basic Mathematical Concepts and Properties

3.1.1. Field

A field (F, +, *) is a set F and two binary operations + and * on F that have the following properties for all u, v, w in F.

i. (F, +) is an Abelian group.

ii. u * (v + w) = u * v + u * w (distributivity).

iii. Let F* represent the set of elements of F not equal to the identity element for the operation +. Then (F*, *) is an Abelian group.

Note that only two operations are defined in a field, which we think of as addition and multiplication. Subtraction and division are not defined, so when (F, +, *) is a field and u and v are elements of F, when we write u − v we really mean u + (−v) where −v is the inverse of v under the operation + and when we write u/v we really mean uv^{-1} where v^{-1} is the inverse of v under the operation *.

Example: (R, +, .) is a field.

3.1.2. Order of the Field

If (F, +, *) is a field, then the number of elements in the set F is called the order of the field. This can be infinite or finite. We write F_q for a finite field with q elements.

3.1.3. Characteristic of the Field

If (F, +, *) is a field and m is the smallest positive integer such that

$$\underbrace{x + x + \ldots + x}_{\text{m times}} = mx = 0 \qquad \textbf{Equ. (3.1)}$$

$\forall x \in F$ is called the characteristic of the field. If there is no integer like this, then the field has characteristic zero.

Example

i. If p is a prime, then the field Z_p has characteristic p.
ii. The field of real numbers has characteristic zero.
iii. If p is a prime, the field of polynomials with coefficients from Z_p will have characteristic p. This field is infinite, yet has characteristic p.

3.1.4. Elliptic Curve

It is the set of points satisfying an equation of the form

$$y^2 = x^3 + a_1x + a_2 \qquad \textbf{Equ. (3.2)}$$

Where the **coefficients a_1 and a_2 are elements of a field**. An elliptic curve E **over the field** F is denoted by E/F. These curves are called to exist in Weierstrass normal form. We can think of the points as being either points in a set or as rational functions of x and y, and can freely change between the two points of view as needed. The requirement that the characteristic of F be greater than 3 is not strictly required for an elliptic curve, but this restriction limits us to the elliptic curves of interest. If the characteristic of the field is equal to 2 or 3, alternative forms other than the Weierstrass normal form need to be used.

We also consider the point at infinity, and write this special point as O, and we have that $P + O = P$ for any point P, so O acts much like the number 0 does in the real numbers.

3.1.5. Algorithm for Elliptic Curve Point Addition

The following algorithm describes the process for addition of two elliptic curve points.

Algorithm:

INPUT: $P_1 = (u_1, v_1)$, $P_2 = (u_2, v_2)$, points on an elliptic curve $y^2 = x^3 + a_1x + a_2$

OUTPUT: P3 = P1 + P3

```
1. If $u_1 = u_2$ return $O$
2. If $P_1 = P_2$ then
3. If $v_1 = 0$ return $O$
4. Else m ← $\frac{3u_1^2+a_1}{2v_1}$
5. Else m ← $\frac{v_2-v_1}{(u_2-u_1)}$
6. $u_3 \leftarrow m^2 - u_1 - u_2$
7. $v_3 \leftarrow m(u_1 - u_3 ) - v_1$
8. Return $P_3 = (u_3, v_3)$
```

3.1.6. E(F) Group

If E is an elliptic curve over a field F then we write E(F) to denote the set of points on E along with the operation of adding points.

If E is an elliptic curve and F is a field and then E(F) is a group. *O* is the identity element.

3.1.7. Elliptic Curve Point Multiplication by an Integer

Multiplication of a point *P* by an integer m is the result of adding a point to itself m times, so that

$$mP = \underbrace{P + P + P + \cdots + P}_{m\ times} \qquad \textbf{Equ. (3.3)}$$

Let P ∈E(F) for some E /F. The **order of a point** P is m if m is the smallest positive integer such that mP = *O*.

3.1.8. m-Torsion Points of the curve

If m is a positive integer and E/F, then E(F)[m] denote the set of points of order m in E(F) and are called the m-torsion points of the curve. If the field F is obvious from the context, it is written as E[m].

3.1.9. Order of Group E(F)

It is the number of points on an elliptic curve E/F, including the point *O*. It is denoted by #E(F).

3.1.10. Embedding Degree of E/F_q

Assume m to be an integer such that m | #E(F_q). If l is the smallest positive integer such that m | (q^l − 1) then l is called the embedding degree of E with respect to m. If m = #E(F_q) then we can simply say that l is the embedding degree of E.

We can think of F_{q^l} as being an extension of F_q in which E(F_q) is a subgroup of $F^*_{q^l}$.

3.1.11. Divisors

In this context, a divisor is a way of characterizing a function f based only on its zeroes, where f (x) = 0, and poles, where f (x) =±∞, like when dividing by zero. We say that a function f (x) has a pole at infinity if f (1/x) has a pole at x = 0, so that a polynomial of degree n has a pole of degree n at infinity. Similarly, we say that a function f (x) has a zero at infinity if f (1/x) has a zero at x = 0. For example,

$$f(x) = \frac{(x-5)^2}{(x+9)^3} = (x-5)^2\ (x+9)^{-3} \qquad \textbf{Equ (3.4)}$$

has a zero of order 2 at *x* = 5, a zero of order 1 at infinity, and a pole of order 3 at *x* = −9. Because a divisor characterizes a function based on its zeroes and poles, two functions that differ by a constant will have the same divisor.

We monitor zeroes and poles of a rational function *f* in what we call a divisor, which we write as *div* (f*)*. We write such a divisor as the sum of the points where *f has* a zero or pole weighted by the multiplicities of the zeroes and poles, with the convention that zeroes get positive weights according to their multiplicities and poles get negative weights according to their multiplicities. In the example above, we write *div* (f*)* = $2(5) + (\infty) - 3(-9)$, to indicate that *f* has a pole of order 3 at *x*=-9. In general, if we can write

$$f(x) = \prod_i (x - x_i)^{a_i} \qquad \textbf{Equ. (3.5)}$$

then we write

$$f(x) = \sum_i a_i(x_i) \qquad \textbf{Equ. (3.6)}$$

The notation for divisors can be a bit tricky, and we will need to be able tell from the context that we dealing with divisors instead of numbers, so that we are not tempted to treat divisors as numbers, trying to simplify expressions like 2(5) − 3(−9) to get a number instead of a divisor.

Note that multiplying rational functions corresponds to addition of their divisors and division of rational functions corresponds to subtraction of their divisors. So if we have *f* (*x*) as defined above and

$$g(x) = \frac{(x+9)^3}{(x+5)^4} \qquad \textbf{Equ. (3.7)}$$

then

$$f(x)g(x) = \frac{(x-5)^2}{(x+5)^4} \qquad \textbf{Equ. (3.8)}$$

which corresponds to adding the divisors:

$$\begin{aligned} div(fg) &= div\ (\ f\ ) + div\ (\ g\ ) \\ &= 2(5) + (\infty) - 3(-9) + 3(-9) + (\infty) - 4(-5) \\ &= 3(5) + 2(\infty)\ - 4(-5) \end{aligned}$$

3.1.12. Formal Sum of a Set

For a set S it is series $\{s_0, s_1, s_{2,} \ldots\}$ of elements of S. It is usually written by means of a placeholder, provided the placeholder is not evaluated.

Example:

i. A power series is a formal sum which we usually write as $a_0 + a_1x + a_2x^2 + ...$, where each $a_i \in S$ for some set S. We write a power series provided the placeholder x is not evaluated, and we could also write the same power series as $\{a_0, a_1, a_{2,}, \ldots\}$.
ii. If $P = \{P_1, P_2, P_{n, \ldots\}}$ is a set of points on E, then $D = a_1(P_1) + a_2(P_2) + . . . + a_n(P_n)$ is a formal sum of the elements of P. In this case, we understand that in D the points in the set P are just placeholders like the variable x in a power series.

Let E be an elliptic curve. A divisor on E is a formal sum of the form

$$D = \sum_{P \in E} n_P(P) \qquad \textbf{Equ. (3.9)}$$

where each n_P is an integer and all but finitely many n_P are 0.

Example:

For points P_1 and P_2 , $D = 3(P_1) + (P_2) - 3(O)$ is a divisor.

3.1.13. Principal Divisor

A divisor D is a principal divisor if there exists a rational function f such that D = div (f). Another definition is that a divisor D on an elliptic curve is principal if it can be written as

$$D = \sum_i a_i(P_i) \qquad \textbf{Equ. (3.10)}$$

where $\sum \mathrm{a_i} = 0$ and $\sum \mathrm{a_i P_i} = O$. If P is a point of order n, then the divisor n(P) - n(O) is a principal divisor.

Example:

Let P_1, P_2 and P_3 be elliptic curve points with $P_3 = P_1 + P_2$. Then $D = (P_1) + (P_2) + (-P_3) - 3(O)$ is a principal divisor.

3.1.14. Support of Divisor

If E is an elliptic curve and

$$\mathrm{D} = \sum_{\mathrm{P}\in\mathrm{E}} \mathrm{n}_{\mathrm{P}}(\mathrm{P}) \qquad \textbf{Equ. (3.11)}$$

is a divisor then the support of D is the set of all points P such that $n_P \neq 0$.

Example:

For the divisor $D = 2(P_1) + 3(P_2) + 5(-P_3) - 3(O)$, the set $\{P_1, P_2, -P_3, O\}$ is the support.

3.1.15. Disjoint Support

Let D_1 and D_2 be divisors. Then we say that D_1 and D_2 have disjoint support if $D_1 \cap D_2 = \emptyset$.

Example:

i. The divisors $D_1 = (P_1) - (O)$ and $D_2 = (P_1 + R) - (R)$ have disjoint support as long as $\{P_1, O\} \cap \{P_1 + R, R\} = \emptyset$.

ii. The divisors $D1 = (P) - (O)$ and $D2 = (Q) - (O)$ do not have disjoint support.

We can think of the divisors as keeping track of where the graph of a function f (x) intersects the graph of an elliptic curve E, or where E = f (x), so they monitor zeroes and poles of E = f (x). In particular, we get a zero when E = f (x), or when the function f (x) crosses the elliptic curve E and we get a pole when f (x) has a pole.

3.1.16. Evaluate a Rational Function f at D

If D is a divisor of the form

$$D = \sum_i a_i(P_i) \qquad \textbf{Equ. (3.12)}$$

then we evaluate a rational function f at D by

$$f(D) = \prod_i f(P_i)^{a_i} \qquad \textbf{Equ. (3.13)}$$

Example:

i. If D = 5(P1) - 2(P2) then

$$f(D) = f(P1)^5 f(P2)^{-2}$$

$$= \frac{f(P_1)^5}{f(P2)^3}$$

ii. If P = (2, 3) and Q =(0, 1) are points on E /F$_{11}$ and D is the divisor D = (P) - (Q) and f is the rational function $f(x, y) = y + 1$, then

$$f(D) = \frac{3+1}{1+1} = 4.2^{-1} = 4.6 \equiv 2\ mod\ 11$$

3.1.17. Weil Reciprocity

Let f and g be rational functions defined on some field F. If div (f) and div (g) have disjoint support then we have that f (div (g)) = g (div (f)).

Example:

Suppose that we have two rational functions f and g defined on F$_{11}$ where

$$f(a) = \frac{a-2}{a-7}$$ **Equ. (3.14)**

and

$$g(a) = \frac{a-6}{a-5}$$ **Equ. (3.15)**

then we have

$$div\ (f) = (2) - (7)$$

and

$$div\ (g) = (6) - (5)$$

then

$$f\big(div(g)\big) = \frac{f(6)}{f(5)} = \frac{7}{4} = 7.3 = 10\ mod\ 11$$

and

$$g\big(div(f)\big) = \frac{g(2)}{g(7)} = \frac{5}{6} = 5.2 = 10\ mod\ 11$$

3.1.18. Equivalent Divisor

Divisors D1 and D2 are equivalent if D = D1 − D2 is a principal divisor.

Example: If f is a rational function, the divisors (P) − (O) and (P) − (O) + div (f) are equivalent.

3.1.19. Tate Pairing

Let E/F_q be an elliptic curve, P ∈$E(F_q)[n]$ and Q ∈$E(F_{q^k})$. Let f_P be a rational function, div (f_P) be equivalent to n(P) − n(O) and A_Q be a divisor equivalent to (Q) − (O) so that support of div (f_P) and A_Q is disjoint. Then the Tate pairing is defined to be e (P, Q) = f_P (A_Q). This definition does not produce a unique value, and will include a constant that is an nth power of some element of F_{q^k} .

The Tate pairing operates on pairs of points P ∈$E(F_q)[n]$ and Q ∈$E(F_{q^k})$, and produces a result in $F^*_{q^k}$. e (P, Q) denote Tate pairing of the points P and Q. For a point P of order n, to get e (P, Q) first we need to find a rational function f_P so that div (f_P) is equivalent to n(P) − n(O) and then evaluate f_P at a divisor equivalent to (Q) − (O).

3.1.20. Properties of the Tate Pairing

i. **Non-Degenerate:** For each P $\in \mathrm{E}(\mathrm{F_q})[\mathrm{n}]/\{O\}$ there is some Q $\in \mathrm{E}(\mathrm{F}_{\mathrm{q}^k})$ with e (P, Q) ≠1.

ii. **Bilinear:** For each P, P_1, P_2 $\in \mathrm{E}\big(\mathrm{F_q}\big)[\mathrm{n}]$ and Q, Q_1, Q_2 $\in \mathrm{E}(\mathrm{F}_{\mathrm{q}^k})$we have e ($P_1$ + P_2, Q) = e (P_1, Q) e (P_2, Q) and e (P, Q_1 + Q_2) = e (P, Q_1) e (P, Q_2).

3.1.21. Miller's Algorithm

We calculate n(P) − n(O) by the double-and-add technique, and finding a divisor equivalent to n(P) − n(O) in this way is called Miller's algorithm.

Algorithm: TatePairing

INPUT: Elliptic curve E: $y^2 = x^3 + bx + c$, P ∈E [n] with $n = \sum_{i=0}^{t} c_i 2^i$, Q

OUTPUT: $e\ (P, Q)$

```
1. $\mathrm{f} \leftarrow 1, \mathrm{t} \leftarrow \lfloor \log_2 \mathrm{n} \rfloor, \mathrm{S} \leftarrow \mathrm{P}, \mathrm{R} \leftarrow$ a random point of $\mathrm{E}, \mathrm{R} \neq O, \mathrm{Q} + \mathrm{R} \neq O$
2. For i $\leftarrow$ t $-$ 1 down to 0
3. $\mathrm{f} \leftarrow \mathrm{f}^2 \frac{\mathrm{u}_{\mathrm{S,S}}(\mathrm{Q+R})\mathrm{v}_{2\mathrm{S}}(\mathrm{R})}{\mathrm{v}_{2\mathrm{S}}(\mathrm{Q+R})\mathrm{u}_{\mathrm{S,S}}(\mathrm{R})}$
```

```
S ← 2S
If $c_i$ = 1
    $f \leftarrow f\frac{u_{S,P}(Q+R)v_{S+P}(R)}{v_{S+P}(Q+R)u_{S,P}(R)}$
    S ← S + P
Return f
```

Algorithm: v

INPUT: P, Q

OUTPUT: $v_P(Q)$

```
If P = $O$
    Return 1
Return $x_Q - x_P$
```

Algorithm: tangent_u

INPUT: P, Q on an elliptic curve E: $y^2 = x^3 + bx + c$

OUTPUT: $u_{P,\,P}$ (Q)

```
If P = $O$
    Return 1
If $y_P$ = 0
    Return v(P, Q)
$m \leftarrow \frac{3x_P^2+b}{2y_P}$
Return $y_Q - y_P - mx_Q + mx_P$
```

Algorithm: u

INPUT: P_1, P_2, Q

OUTPUT: $u_{P_1,P_2}(Q)$

```
If $P_1$ = $O$
    Return v($P_2$ , Q)
If P2 = $O$ or P1 + P2 = $O$
    Return v($P_1$ , Q)
```

```
5. If $P_1 = P_2$
6. Return tangent_u ($P_1$ , $Q$)
7. $m \leftarrow \frac{y_{P_2} - y_{P_1}}{x_{P_2} - x_{P_1}}$
8. Return $y_Q$ - $y_{P_1}$-m$x_Q$ + $mx_{P_1}$
```

3.1.22. Cryptographic Key and Parameters

A cryptographic *key* is a value that defines the operation of an encryption or decryption algorithm. Values that are used for all users of a system are called *parameters* instead. While conventional public key algorithms have two keys, private key and public key, IBE algorithms typically have a set of public parameters.

3.2. Identity Based Encryption

In development of a conventional public key system that uses digital certificates to administer public-keys, a public private key pair is created randomly by a user, or an agency working in support of a user, such that the public-key comprises of all the parameters needed for applying it

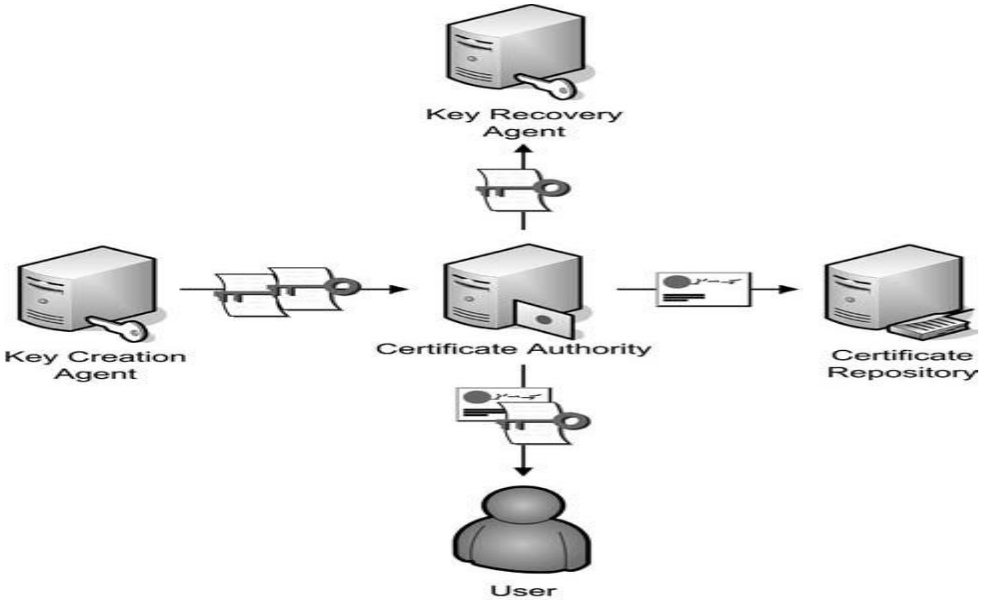


Figure 3.1: Key-Generation in a public-key system.

in cryptographic calculations. Random key generation is not strictly needed by public key algorithms which are used in these systems, rather it is required by the prevailing standards which outline how to use these algorithms. After public key is created, both public-key and the identity of the key owner, is signed digitally by a CA to create a digital certificate which is then used to transfer and control the key. The private key owner then receives a copy of the certificate and a copy of it is stored in a certificate repository which is accessible by anyone who need to get a user’s public key. In applications in which it is necessary to recover lost or unavailable private keys, the private keys are also securely logged by a key recovery agent. If an agent created the private key on behalf of a user, which usually happens when keys are generated in a centralized manner to ensure logging of copies to allow the recovery of lost or unavailable keys, the CA also deliver the private key to the key owner. This is shown in Figure 3.1.

In a conventional public key framework, the ID of a person is usually cautiously confirmed before issuing a digital certificate to him, a procedure that is normally somewhat costly. The procedure of creating public private key pairs is computationally costly. Generating two 512 bit prime numbers which are appropriate to generate a 1024 bit RSA private-key is indeed possible, but creating bigger primes gradually becomes more costly. Generating two 7680 bit primes which are appropriate to create a 15360 bit RSA private-key is certainly not a task that can be performed easily by computers being used presently, still we need large key size to ensure secure transmission of 256 bit AES keys that are widely utilized presently. Because verifying identities of users and creating keys is expensive, digital certificates are frequently allotted with reasonably extended validity times, frequently between 1-3 years. Because of the fairly extended validity time of the public-keys handled by digital certificates, it becomes compulsory to verify the validity of the key in a certificate before applying it. This is shown in Figure 3.2. There have been many solutions proposed for validating public-keys, still the existing technologies being used are relatively unproven and experience practical difficulties with large no. of users.

To obtain a public-key which is enclosed in a digital certificate, a sender asks the public warehouse where he/she can find the certificate and recovers the certificate. As a public-key can be legal for longer time, it becomes important to inspect the validity of public-key for before making use of it. This is usually done by examining a record of void certificates or by receiving the validity of a certificate by asking an online service. After all the essential validity inspection is completed, the sender then encrypt message by applying the public-key for the public-key possessor. Because receiver holds the private-key he/she is able to decrypt this message. This is shown in Figure 3.2.

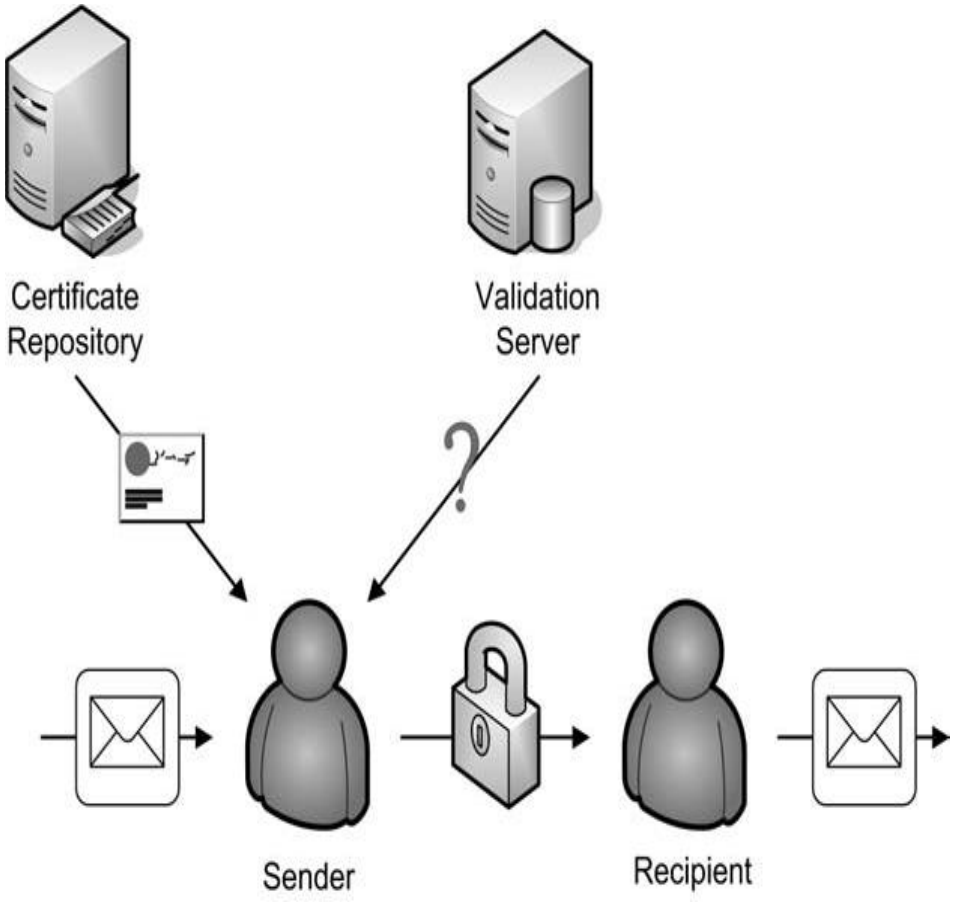
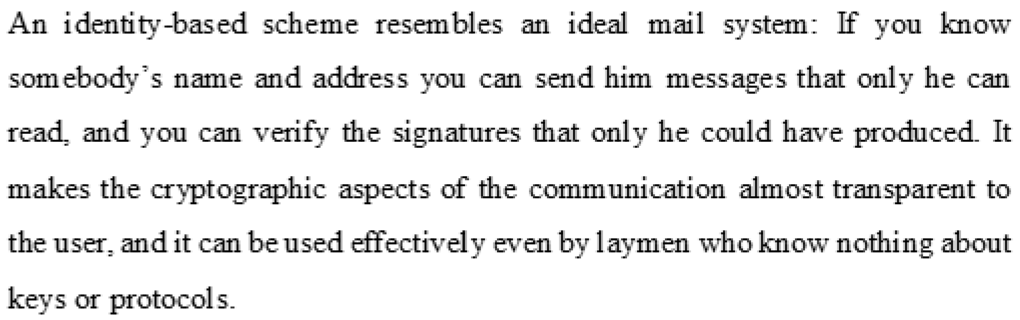


Figure 3.2: Validation and usage of a public key in a traditional PKS.

In 1984, IBE was introduced by Adi Shamir, when he described a rough outline of the properties that such a system should have and how it could be used, although he was unable to find a secure and feasible technology that worked as he described. He seemed to see the advantages of IBE to be related to its ease of use relative to other technologies when he described IBE in this way:



> An identity-based scheme resembles an ideal mail system: If you know somebody's name and address you can send him messages that only he can read, and you can verify the signatures that only he could have produced. It makes the cryptographic aspects of the communication almost transparent to the user, and it can be used effectively even by laymen who know nothing about keys or protocols.

An IBE scheme is similar to conventional public-key scheme, but is also fairly dissimilar in a number of ways. While conventional public-keys comprise the greater part of the parameters expected to utilize the key, IBE system requires a trusted third party to deliver a list of public parameters. A client would then use these parameters to compute the IBE pubic key of any client and utilize it to encrypt data to that client. This process is shown in Figure 3.3.

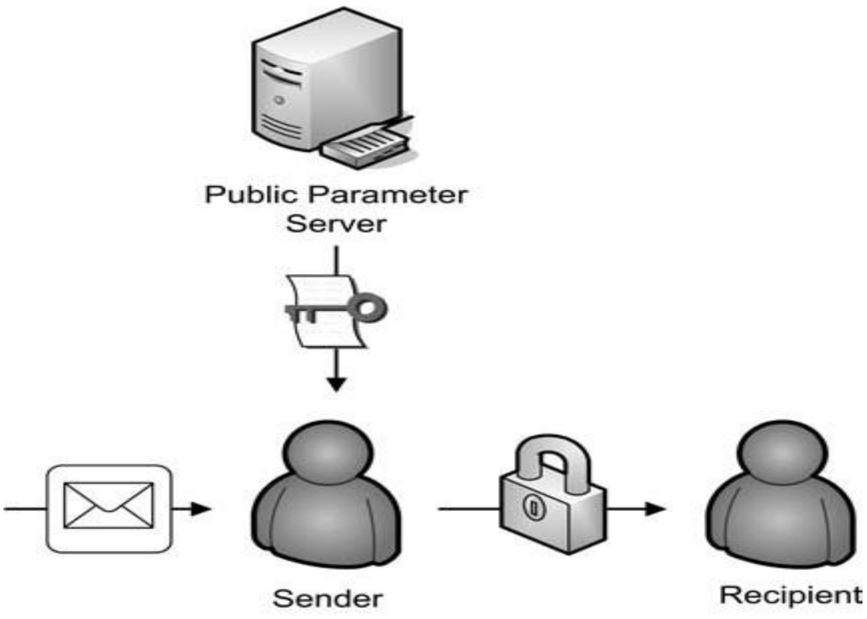


Figure 3.3: Encrypting with an IBE system.

The recipient of IBE-encrypted information then authenticates in some way to a trusted third party PKG which computes the IBE private key corresponding a specific IBE public key. The PKG typically uses confidential information called a *master secret*, plus the user's identity, to calculate such a private key. After this private key is calculated, it is securely distributed to the authorized user. This is shown in Figure 3.4.

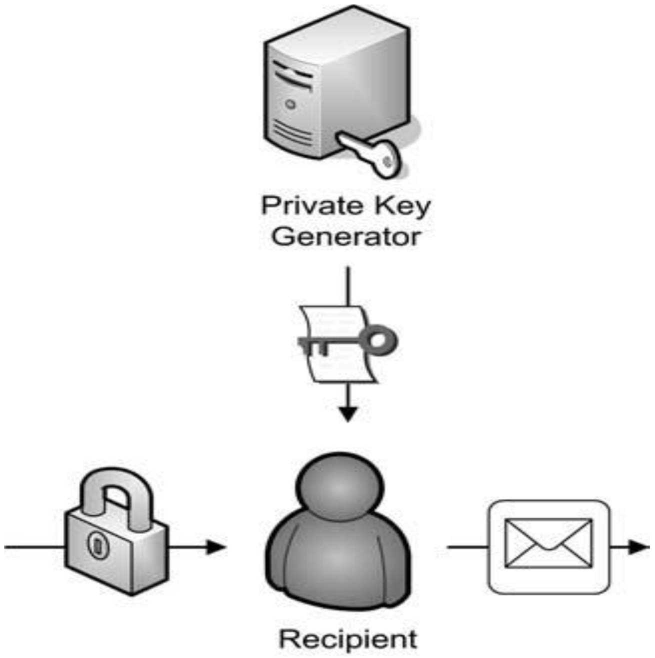


Figure 3.4: Decrypting with an IBE system.

In a traditional public-key scheme, we can summarize the algorithms involved in the creation and use of a key pair as key generation, encryption, and decryption. Two additional algorithms, certification and key validation, are often used in many implementations of such schemes. To fully specify the operation of such a scheme we need to define the operation of each of these algorithms. In the key generation step, one key of the pair is created randomly and the other key in the pair can be calculated from it. After this, the public key and the identity of its owner is digitally signed by a CA to create a digital certificate. Encryption is performed using the certificate. Decryption is performed using the private key.

In an IBE system too, there are also four algorithms. These are called setup, extraction, encryption, and decryption. Setup is the algorithm with which the parameters needed for IBE calculations are initialized, including the master secret that a PKG uses to calculate IBE private keys. Extraction is the algorithm for calculating an IBE private key from the parameters established in the setup step, along with the identity of a user, and uses the master secret of the PKG to do this. Encryption is performed with an IBE public key that is calculated from the parameters from the setup step and the identity of a user. Decryption is performed with an IBE private key that is delivered by the PKG.

3.3. Assumptions

In the case of IBE, we have assumptions that are different than those that we make for traditional public key technologies. Anyone can calculate an IBE private key from a user's identity with the correct IBE public parameters, but we need to assume that users receive the correct set of IBE public parameters. If we can trick a user into using the incorrect public parameters, we can trick them into sending messages that can easily be decrypted. We also need to assume that the IBE PKG is authenticating users appropriately before granting IBE private keys to them. If the PKG can be deceived into giving the IBE private key of some other user then we can decrypt messages that are encrypted with the IBE public key of that user.

Both traditional public-key technologies and IBE are based on the assumption about the intractability of certain number-theoretical calculations. If these calculations are sufficiently difficult for an adversary to perform, then we can reasonably assume that they cannot perform the calculations, and that our system is reasonably secure.

3.4. Boneh_Franklin IBE

The Boneh-Franklin basic scheme uses a shared secret that can be calculated by both the sender and receiver of a message to encrypt a plaintext message.

3.4.1. Setup of Parameters

To implement Boneh-Franklin IBE we need a security parameter that defines the level of bit strength that the encryption will provide. Then we need to define groups G_1 and G_T and a pairing $\hat{e}$: G_1 x $G_1 \rightarrow G_T$. To do this we pick E /F_q with embedding degree k, and a prime p such that p | #E(F_q). We also require that p^2 | #E(F_q) to ensure that the subgroup of order p that we will hash identities into is unique. The parameter p is the order of the groups G_1 and G_T, and G_T is a subgroup of $F^*_{q_k}$.

We then randomly pick a point P ∈ $E(F_q)[p]$ and let $G_1 = \langle P \rangle$ and $G_T = \langle \hat{e}(P, P) \rangle$, which are cyclic groups of prime order p. Next, we pick a random integer s $\in Z_p^*$ and to calculate sP. To map an identity ID to a point Q_{ID} we also need a cryptographic hash function H_1: {0, 1}* → G_1. To encrypt a message of n bits using Boneh-Franklin IBE we also need another cryptographic hash function H_2: G_T → $\{0, 1\}^n$ that hashes elements of G_T into a form that we can combine with the plaintext message, which is a bit string of length n. These elements form the public parameters and master secret. The integer s is the master secret; all other values comprise the public parameters.

The values of p, q, and E, are implicit in the definition of the group G_1. Because of this public parameters reduce to a much smaller list, and we define public parameters of a IBE system to be BFParams = (G_1 , G_T, $\hat{e}$, n, sP, H_1 , H_2) without introducing any ambiguity.

3.4.2. Extraction of the Private Key

Once the public parameters are determined, the private key associated with the identity ID is calculated by mapping identity to a point on curve E by calculating $Q_{ID} = H_1$ (ID) and multiplying point Q_{ID} by master secret s to generate the private key sQ_{ID} . This is summarized in Table 3.1.

Table 3.1: Private Key for Boneh-Franklin IBE System

Element	**Type**	**Comments**
sQ_{ID}	Point on elliptic curve	Private key corresponding to identityID, QID = H_1 (ID)

3.4.3. Encryption in IBE

To encrypt the message *M* ∈{0, 1}*n* to the recipient with identity *ID*, the sender follows the following steps.

1. Generates a random integer r ∈ Z_p^* and calculates rP.
2. Calculates $Q_{ID} = H_1(ID)$ from the recipient's identity ID to calculate

$$K = H_2(\hat{e}(rQ_{ID}, sP)) \qquad \textbf{Equ. (3.16)}$$

3. Sets the ciphertext corresponding to the pair CT = (CT_1, CT_2) where CT_1 = rP and CT_2 = M ⊕ K.

3.4.4. Decryption in IBE

When the recipient receives the ciphertext CT = (rP, M $\oplus$ $H_2(\hat{e}(rQ_{ID}, sP))$ = (CT_1, CT_2) he performs the following steps.

1. Calculates K = $H_2(\hat{e}(sQ_{ID}, CT_1))$ from the ciphertext component CT_1 and private key sQ_{ID}.
2. Calculates M = CT_2 $\oplus$ K.

The plaintext M can be recovered from it because the sender calculates K as

$$K = H_2(\hat{e}(rQ_{ID}, sP)) = H_2(\hat{e}(Q_{ID}, sP)^{rs}) \qquad \textbf{Equ. (3.17)}$$

and recipient calculates K as

$$K = H_2\big(\hat{e}(sQ_{ID}, CT_1)\big) = H_2(\hat{e}(Q_{ID}, P)^{sr}) \qquad \textbf{Equ. (3.18)}$$

3.5. Attribute Based Encryption

Most encryption use an “all or nothing” approach to decrypting ciphertext. If the information is encrypted it becomes extremely difficult for the user to share their information in encrypted form at different granularity level. If you have the key, then you can decrypt the message at any time and have full access to its contents. This can be a problem for time sensitive or forensic applications that might need to limit access to the encrypted information.

Attribute based encryption that is also known as ABE is a very recent form of public key encryption where the ability to encrypt/decrypt a file is based on a universe of attributes that a set of users may or may not have. ABE is extremely valuable for segmenting different sets of information such that not every user within an organization has access to all information simply because they're part of a group. ABE makes it possible to implement many interesting access control mechanisms using cryptography. Traditionally, everyone with the right key can decrypt and thus access the encrypted file. ABE is not that different. However instead of using the public key to encrypt the files, it uses attribute(s) or a key based on attributes to encrypt the files. It reduces the number of key used and thus make encryption and decryption process faster.

Since now everyone in, let’s say, the group "IT support staff" can access the file, there is no need to encrypt the files again and again with all public keys of every staff member of that group. This leads to immense space savings. Furthermore, at least in a PKI based version of ABE, it is possible to expand the keys of subordinates with those of their superior, resulting in the supervisor being able to access the files of his staff but not the other way around. Again with ABE a whole group can be "superior" to another one, removing the need to save differently encrypted copies of the same file on the disk.

ABE has great applications in multi-level security situations where you need to segment information within the same strata of employee level. It's a great way of implementing Multi-Level Security (or MLS), with applications in military organizations (not allowing everyone with Top Secret or SCI clearance to know how to build a nuclear weapon), finance firms (installing a Chinese wall between different investment groups within a large financial services org such as an investment bank to stop insider trading), and healthcare (making sure only doctors treating certain patients have access only to their information).

For example: Let's say you have a universe of the below attributes:

{A, B, C, D}

ABE encrypts a file such that one requires a certain set of logical operations to unlock a file. A file may be encrypted with the following key:

k = (A^B) V C

This means that only a user who holds A and B attributes OR holds a C attribute can decrypt that file.

What does this really mean? Let's pretend we're talking about an investment bank who's currently working on IPO'ing a big startup. For all sensitive information related to this project, you want to make sure that only two groups of people have access:

i. Someone who is of analyst level or higher (A) who is on the investment banking team (B)

ii. A compliance analyst from the SEC who is overseeing this process (C).

The key above is such that you need to be a minimum analyst level AND are on the investment banking team to view the IPO documents (A ^ B), OR you're from the SEC (C) and reviewing the process for compliance with various regulations. You can't be an analyst+ in another group and see the documents, but the documents are still transparent to this outside and necessary observer.

CHAPTER 4

PROPOSED WORK

This chapter presents the system model and the proposed algorithm to be used in the system. The system can be utilized to upload information in encrypted form on the cloud which can only be decrypted by the designated recipient or a group of recipients.

4.1. System Model

The system for the scheme is modelled as shown in Figure 4.1. In contrast to the original IBE scheme, the revocation of compromised clients is handled by an external entity called KU-CSP. The three major components of the system are the Private Key Generator, the KU-CSP and the Public Parameters Server.

4.1.1. Public Parameter Server

To utilize an IBE system, a client requires a trusted third party called Public Parameter Server to deliver a list of public parameters. A client would then use these parameters to compute the IBE pubic key of any client and utilize it to encrypt data to that client.

4.1.2. Private Key Generator (PKG)

A PKG is a trusted external entity which computes the IBE private key corresponding to a specific IBE public key of a user. The PKG typically uses confidential information called a master secret, plus the user's identity, to calculate such a private key. After this private key is calculated, it is securely distributed to the authorized user.

4.1.3. KU-CSP

The KU-CSP can be imagined as a public cloud hosted by an external entity to provide essential computing facilities to PKG as a standard service over the web. A KU-CSP gives an approach to lessen PKG storage and calculation cost by providing an adaptable, even impermanent augmentation to existing setup. In the event of revocation, rather than asking again for private keys from PKG, unrevoked clients need to approach the KU-CSP for refreshing a lightweight part of their private keys.

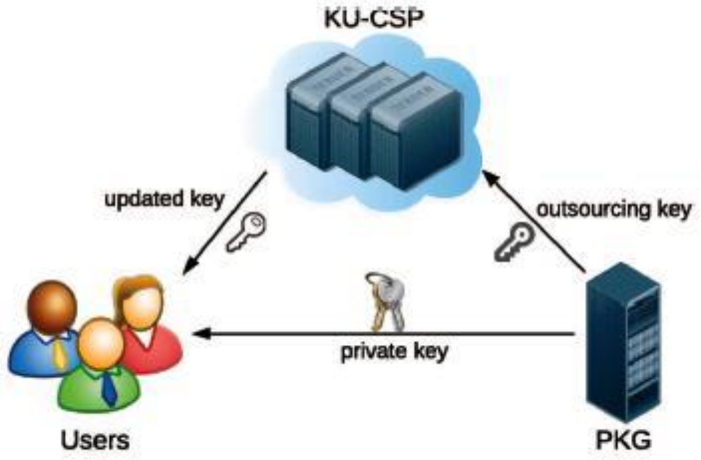


Figure 4.1: System Model

4.2. Algorithm

To further extend the security of IBE, the properties of ABE method is combined with revocable IBE. The algorithms for key generation, encryption, decryption, revocation and key update are modified as per the new approach. We introduced the hybrid method for efficient revocation and security improvement. . In our scheme, KU-CSP handles all the processing required during key update and key issuing while PKG and clients perform a constant number of simple computation steps locally. The component KU-CSP is present for realization of compromised user's revocation. Basically, KU-CSP provides an approach to lessen storage and calculation cost of PKG by facilitating dynamic or impermanent expansion to setup even though it is hosted far away from PKG. This solution solves the problem of efficient key revocation using IBE.

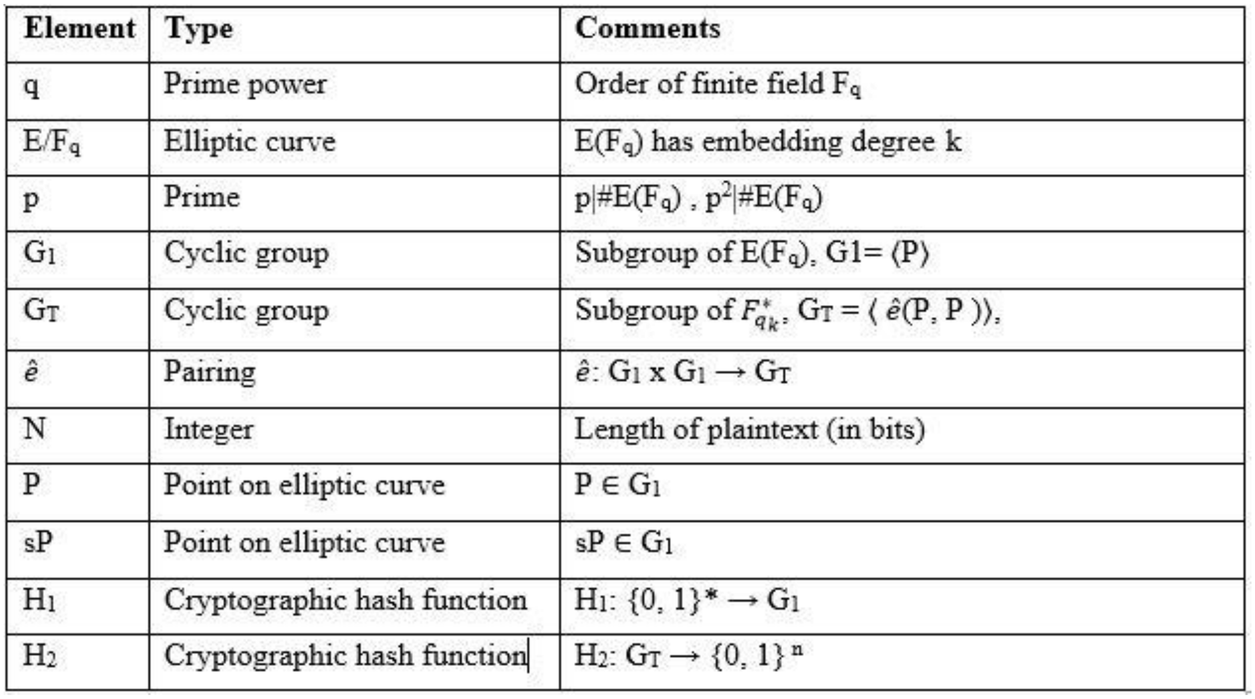
To solve another problem of enhancing the security of efficient key revocable IBE method, we contributed the properties of ABE method while encryption and decryption process. By considering the proposed approach the original functions of IBE are modified by including the time component. Note that three lists AL, TL and RL are utilized in our definition, where AL is a list of attributes, TL is a list of old and new time period, RL is a list of revoked users' identities.

The proposed scheme consists of six phases.

1. Setup (PP)

The PKG executes this algorithm. It requires a single input which is a set of public parameters and returns the public key P_{pub} and master secret MS =s as output. The Table 4.1 and Table 4.2 summarizes the public parameters used in the system.

Table 4.1: Public Parameters of Proposed System



Element	Type	Comments
q	Prime power	Order of finite field F_q
E/F_q	Elliptic curve	$E(F_q)$ has embedding degree k
p	Prime	$p \mid \#E(F_q)$, $p^2 \mid \#E(F_q)$
G_1	Cyclic group	Subgroup of $E(F_q)$, $G1= \langle P \rangle$
G_T	Cyclic group	Subgroup of $F^*_{q_k}$, $G_T = \langle \hat{e}(P, P) \rangle$.
$\hat{e}$	Pairing	$\hat{e}$: G_1 x $G_1 \rightarrow G_T$
N	Integer	Length of plaintext (in bits)
P	Point on elliptic curve	$P \in G_1$
sP	Point on elliptic curve	$sP \in G_1$
H_1	Cryptographic hash function	H_1: $\{0, 1\}^* \rightarrow G_1$
H_2	Cryptographic hash function	H_2: $G_T \rightarrow \{0, 1\}^n$

Table 4.2: Master Secret for Proposed System

Element	Type	Comments
MS or s	Integer	s$\in Z_p^*$

In this phase we follow the following steps:

Step 1: Select a random generator $P \in G_1$

Step 2: Select a random integer $s \in Z_p^*$, where s is the master key of the system

Step 3: Set public key P_{pub} = sP.

Step 4: Deliver the master secret key to KU-CSP via a secure channel

2. KeyGen(MS, ID,AL,RL)

The PKG executes this algorithm for each user. It requires four inputs namely, MS – the master secret, ID- an identity, AL – the attribute list, and RL – the revocation list.

In this phase the following steps are followed:

Step 1: Check if ID ∈ RL, then abort the algorithm.

Step 2: Calculate Q_{ID} = H_1(ID|AL) for user with identity ID.

Step 3: Calculate the private key of user as key D_{ID} = s.Q_{ID}

Step 4: Calculate the initial time key as $T_{ID_i} = s.H_1(ID, T_i)$.

Step 5: Insert the user with {ID,s, T_{ID_i}} into list of users.

Step 6: Deliver the private key to the user via a secure channel.

3. KeyUpdate(ID, RL, T_i)

The KU-CSP executes this algorithm. It requires three inputs namely, RL- the revocation list, ID - an identity, and Ti - the time period i.

In this phase the following steps are followed:

Step 1: Check if ID ∈ RL, then output ϕ.

Step 2: Calculate $R_{ID_i} = H_1(ID, T_i)$..

Step 3: Calculate the user's time updated key as T_{ID_i}= s. $R_{ID_i} \in G_1$ for time period T_i

Step 4: Deliver the updated time key to the user via a public channel.

Step 5: The user update his/her private key as $D_{ID_i} = D_{ID} + T_{ID_i}$ for time period T_i.

4. Encrypt (M, ID, P_{pub} , AL, T_i)

This algorithm is run by the sender. It requires five inputs namely, M – a message, ID – the identity of receiver ,P_{pub} – the public key of receiver , AL – the attribute list, and T_i - time period i.

In this phase the following steps are followed:

Step 1: Sender selects a random number, $r \in Z_p^*$.

Step 2: Sender calculates $Q_{ID_i} = Q_{ID} + R_{ID_i} = H_1(ID|AL) + H_2(ID, T_i)$.

Step 3: Sender calculates CT_1 = r.P and CT_2= M$\oplus$ $H_2(\hat{e}(Q_{ID_i}, P_{pub}))$.

Step 4: The ciphertext of the message M is CT= (CT_1, CT_2).

The Figure 4.2 shows the process of encryption.

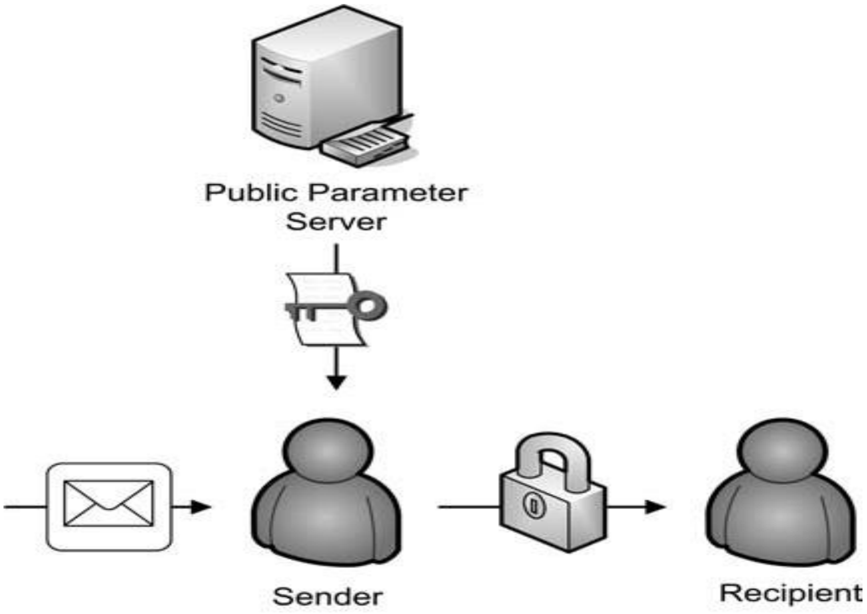


Figure 4.2: Encrypting a message in the system.

5. Decrypt (DID, CT)

It is executed by recipient and it requires two input namely, DID - the private key, and CT - the ciphertext encrypted at time period T_i, with identity ID, and attribute list AL.

In this phase the following steps are followed:

Step 1: If ID $\neq$ ID' or $T_i \neq T_j$, then return ϕ.

Step 2: The receiver uses his private key to calculate message a M = $CT_2 + H_2\left(\hat{e}\left(D_{ID_i}, CT_1\right)\right)$.

The decryption process is as shown in Figure 4.3 below.

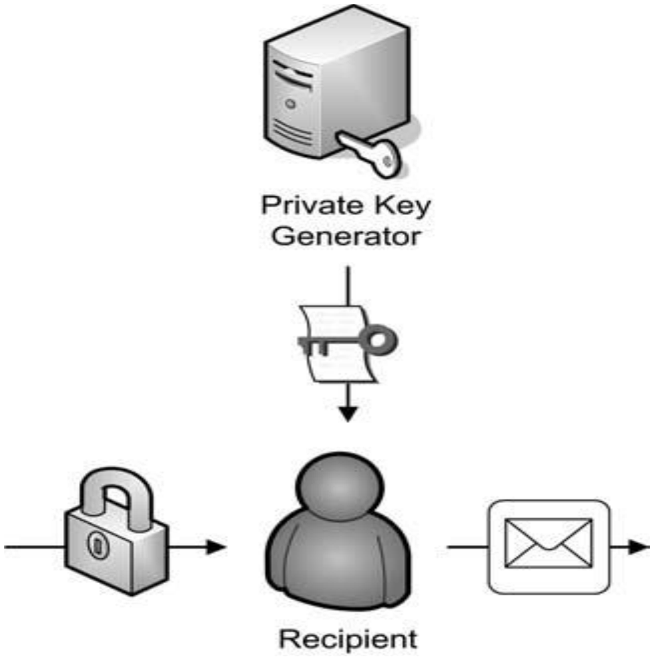


Figure 4.3: Decrypting a message in the system.

6. Revoke ($\{ID_{m1}. . . ID_{mk}\}$, RL, TL)

Revocation algorithm is executed at KU-CSP. It requires three input namely, $\{ID_{m1}$, ID_{m2}, . . , $ID_{mk}\}$ - the set of identities to be revoked, RL - the revocation list, and TL - the time list.

In this phase the following steps are followed:

Step 1: A revoked user sends a KeyUpdate request to KU-CSP

Step 2: The KU-CSP runs the KeyUpdate but since ID ∈ RL, returns ϕ.

Step 3: The KU-CSP updates the revocation list as RL' = RL $\cup$ $\{IDm1, IDm2, .., IDmk\}$.

It also updates the time period T_{i+1} and time list TL'

4.3. Flow Diagram

The Figure 4.4 shows the flow of messages and action taken while performing KeyGen, KeyUpdate and Revocation

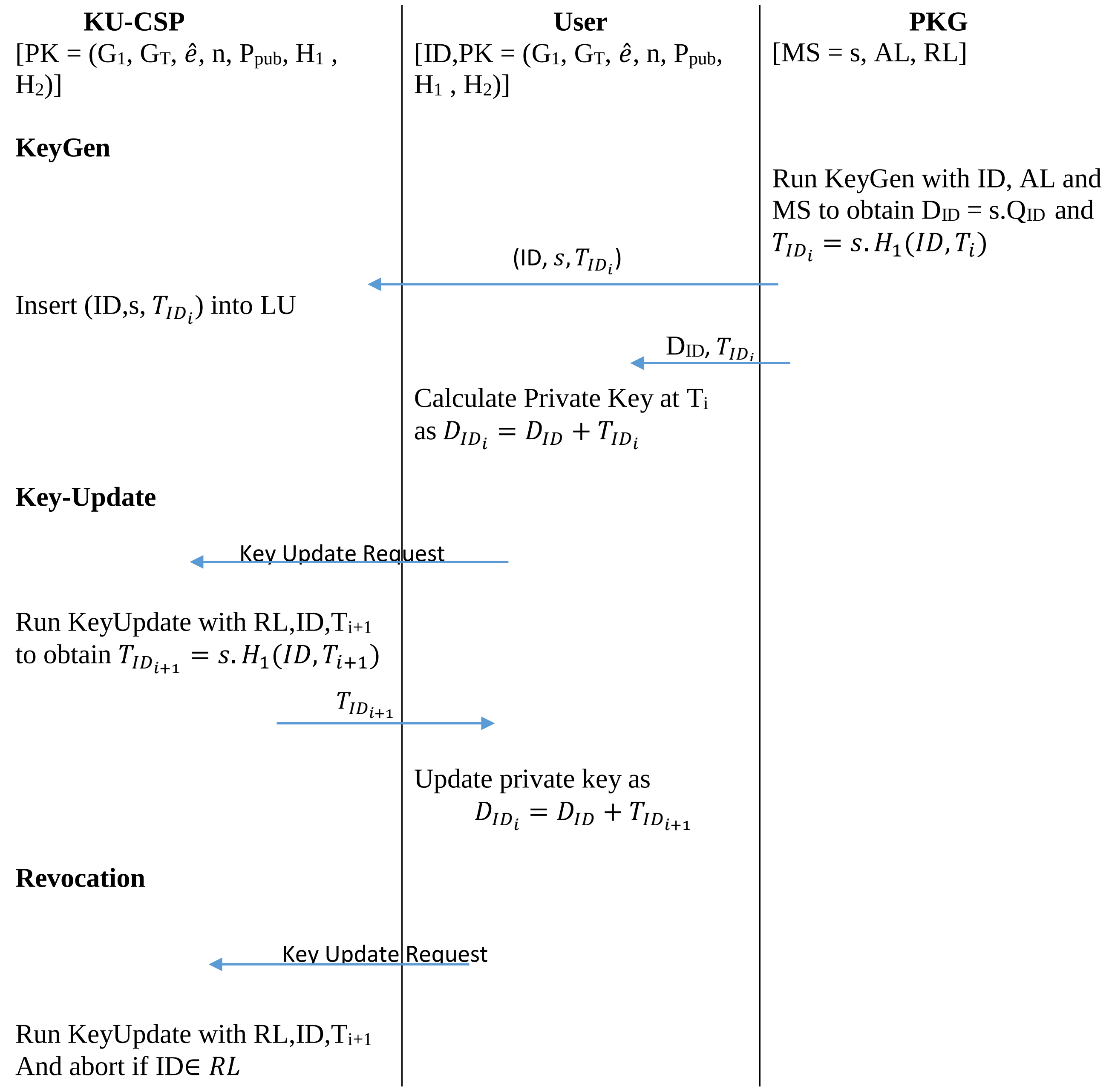
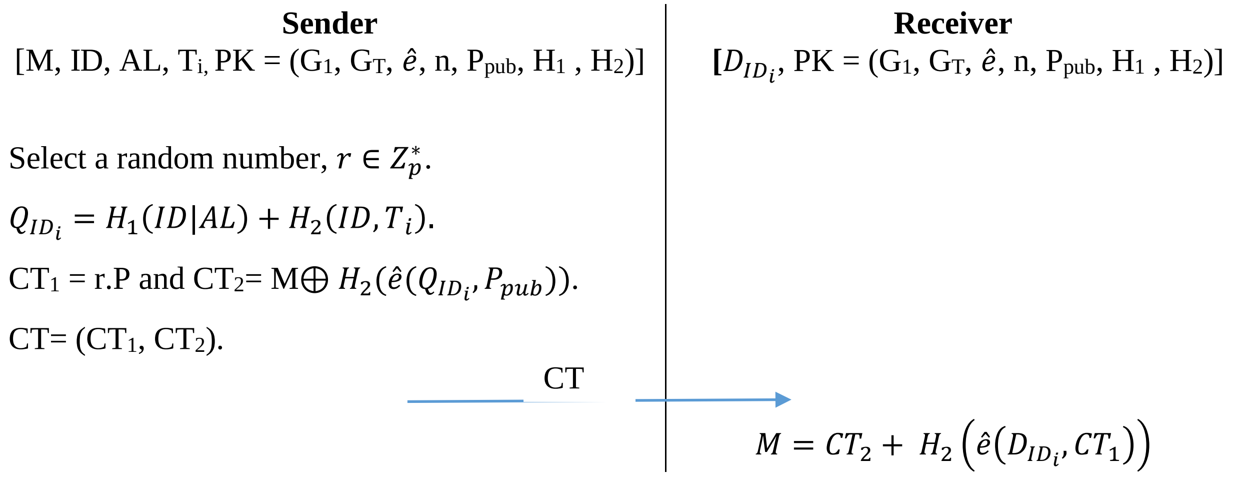

KU-CSP
[PK = (G_1, G_T, $\hat{e}$, n, P_{pub}, H_1 , H_2)]
User
[ID,PK = (G_1, G_T, $\hat{e}$, n, P_{pub}, H_1 , H_2)]
PKG
[MS = s, AL, RL]
KeyGen
Run KeyGen with ID, AL and MS to obtain D_{ID} = s.Q_{ID} and $T_{ID_i} = s.H_1(ID, T_i)$
(ID, s, T_{ID_i})
Insert (ID,s, T_{ID_i}) into LU
D_{ID}, T_{ID_i}
Calculate Private Key at T_i as $D_{ID_i} = D_{ID} + T_{ID_i}$
Key-Update
Key Update Request
Run KeyUpdate with RL,ID,T_{i+1} to obtain $T_{ID_{i+1}} = s.H_1(ID, T_{i+1})$
$T_{ID_{i+1}}$
Update private key as $D_{ID_i} = D_{ID} + T_{ID_{i+1}}$
Revocation
Key Update Request
Run KeyUpdate with RL,ID,T_{i+1}
And abort if ID∈ RL

Figure 4.4: Flow Diagram for KeyGen, KeyUpdate and Revocation

The Figure 4.5 shows the flow of messages and action taken during Encryption and Decryption



Sender

[M, ID, AL, $T_{i,}$ PK = (G_1, G_T, $\hat{e}$, n, P_{pub}, H_1 , H_2)]

Receiver

[D_{ID_i}, PK = (G_1, G_T, $\hat{e}$, n, P_{pub}, H_1 , H_2)]

Select a random number, $r \in Z_p^*$.

$Q_{ID_i} = H_1(ID|AL) + H_2(ID, T_i)$.

CT_1 = r.P and CT_2= M$\oplus$ $H_2(\hat{e}(Q_{ID_i}, P_{pub}))$.

CT= (CT_1, CT_2).

CT

$$M = CT_2 + H_2\left(\hat{e}\left(D_{ID_i}, CT_1\right)\right)$$

Figure 4.5: Flow Diagram for Encrypt/Decrypt

4.4. Security Definition

4.4.1. Insider Adversary

"It is defined as a curious user with identity ID but revoked before time period T_i. Such adversary tries to obtain useful information from ciphertext intended for him/her at or after T_i(e.g. time period $T_{i,,}$ T_{i+1},....) through colluding with other users even if they are unrevoked. Therefore, it is allowed to ask for private key including identity component and updated time component for cooperative users."

4.4.2. BDH Assumption

It is used to verify the security of ID-based encryption scheme. It is stated as follows:

"Given an additive cyclic group G_1 and P, aP, bP, cP $\in$ G_1 for unknown$a, b, c \in Z_q^*$, no probabilistic polynomial time (PPT) algorithm A with non-negligible probability which can compute $\hat{e}$(P, P)^{abc} $\in$ G_2. The successful probability (advantage) of A is presented as Adv_A=Pr[P$\in$$G_1$, $a, b, c \in Z_q^*$|A(P,aP,bP,cP)= $\hat{e}$(P, P)^{abc} $\in$$G_2$], where the probability is over the random choice consumed by A."

4.4.3. Security Concept

We have followed the security requirement of IBE scheme [21] to propose the requirement of our scheme. Our scheme is semantically secure against an adaptive CPA (IND-RID-CPA) if no PPT adversary A has a non-negligible advantage against the challenger B in the IND-RID-CPA game described in [16].

The advantage of adversary A in attacking our scheme can be defined as the function of the security parameter k as follows:

$$\mathrm{Adv}_{\mathrm{A}}(\mathrm{k}) = |\Pr[\alpha' = \alpha] - 0.5| \qquad \textbf{Equ. (4.1)}$$

Where $\alpha \in \{0, 1\}$ is randomly picked by the challenger at time period i to create a target ciphertext C* = Encrypt(ID*, i*, M_β) with plaintext pair (M_0,M_1) generated and given to challenger by adversary A and α' is the guess by A.

4.5. Security Analysis

To show that our scheme is semantically secure against adaptive CPA (IND-RID-CPA) for the outside adversary and the revoked user the following two theorems are given.

Theorem 1: Assume that the random oracles are represented by three hash functions H_0, H_1, and H_2. Now supposing that the BDH problem is hard in groups generated by G, our scheme is a semantically outsider-secure IBE scheme (IND-O-RID-CPA). Conclusively, suppose that there exists an outside adversary A that has advantage ε(k) against our scheme. Assume that A makes at most q_{H_i}> 0 probes to hash functions H_j (j = 0, 1, 2), q_{KG}> 0 KeyGen probes, and q_{KU} > 0 KeyUpdate probes. Then there exists an algorithm B that solves the BDH problem in groups generated by G with advantage at least

$$Adv_{G,B}(k) \geq \frac{2\varepsilon(k)}{e(1+q_{KG})q_{H_2}} \qquad \textbf{Equ. (4.2)}$$

where e is the base of the natural logarithm.

Proof. The BDH parameters $\langle q, G_1, G_2, \hat{e}\rangle$ produced by G and a random instance of the BDH problem $\langle P, aP, bP, cP\rangle$ for these parameters are given as input to the algorithm B, i.e. P is random in G_1^* and a, b, c are random in Z_q^* , where q is the order of G_1,G_2. Let the solution to this BDH problem be D = $\hat{e}$(P, P)^{abc} ∈ G_2. By computing the probability of algorithm B termination in the simulation we can prove the above theorem.

Let the total number of KeyGen probes made by algorithm A be q_{KG}. Then B does not terminate in Phase 1 or 2 with probability $\delta^{q_{KG}}$. Similarly it does not terminate in the challenge step with probability 1−δ. Therefore, B does not terminate in the simulation with probability $\delta^{q_{KG}}$(1−δ). At δ_{opt} = 1−1/(q_{KG} + 1) it maximizes the above probability. Using δ_{opt}, B does not terminate with probability ≥ 1 / e(1 + q_{KG}). The probability that Algorithm B yields the right answer D is ≥ $2\varepsilon \ / \ q_{H_2}$. Suppose there exists an outside adversary A (IND-O RIDCPA) with advantage $\varepsilon(k)$ against our scheme. Then it is possible to build an algorithm B which can solve the BDH problem in groups generated by G with advantage≥ $2\varepsilon(k)/e(1 + q_{KG})q_{H_2}$, as required.

Theorem 2

Assume that the random oracles are represented by three hash functions H_0, H_1, and H_2. Now supposing that the BDH problem is hard in groups generated by G, our scheme is a semantically insider-secure IBE scheme (IND-I-RID-CPA). Conclusively, suppose that there exists an inside adversary A that has advantage ε(k) against our scheme. Assume that A makes at most q_{H_i}> 0 probes to hash functions H_j (j = 0, 1, 2), q_{KG}> 0 KeyGen probes, and q_{KU} > 0 KeyUpdate probes. Then there exists an algorithm B that solves the BDH problem in groups generated by G with advantage at least

$$Adv_{G,B}(k) \geq \frac{2\varepsilon(k)}{e(1+q_{KU})q_{H_2}} \qquad \textbf{Equ. (4.3)}$$

where e is the base of the natural logarithm.

Proof. The BDH parameters $\langle q, G_1, G_2, \hat{e}\rangle$ produced by G and a random instance of the BDH problem $\langle P, aP, bP, cP\rangle$ for these parameters are given as input to the algorithm B, i.e. P is random in G_1^* and a, b, c are random in Z_q^* , where q is the order of G_1,G_2. Let the solution to this BDH problem be D = $\hat{e}(P, P)^{abc} \in G_2$. By computing the probability of algorithm B termination in the simulation we can prove the above theorem.

The analysis is similar to those of Theorem 1. B does not terminate with probability ≥ 1 / e(1 + q_{KU}). And the probability that Algorithm B yields the right answer D is $\geq 2\varepsilon \ / \ q_{H_2}$. Suppose there exists an inside adversary A (IND-I RIDCPA) with advantage $\varepsilon(k)$ against our scheme. Then it is possible to build an algorithm B which can solve the BDH problem in groups generated by G with advantage $\geq 2\varepsilon(k)/e(1+q_{KU})q_{H_2}$, as required.

CHAPTER 5

IMPLEMENTATION

In this chapter, we will discuss the experimental setup of the research work done. Section 5.1 gives a brief description of programming tools, software and various libraries used. Section 5.2 dives the details of the system used for development and execution of the system. Section 5.3 show some runtime snapshots of the system.

5.1. Programming Tools and Software Used

The proposed scheme is implemented using .NET 4.5 platform and SQL Server 2016 Database. Google drive is used for storing the encrypted data. Microsoft Visual Studio 2013 IDE is used for development of the scheme.

The following external libraries are used in our development work:

1. **Google.Apis**
2. **Google.Apis.Auth:** to authenticate our application to Google drive services.
3. **Google.Apis.Core:** to integrate our application with Google services.
4. **Google.Apis.Drive:** to store, delete, and access files and folders on Google drive
5. **BouncyCastle:** to perform big number calculations on elliptic curves

5.2. System Specification

The system on which the scheme is implemented has the following specifications

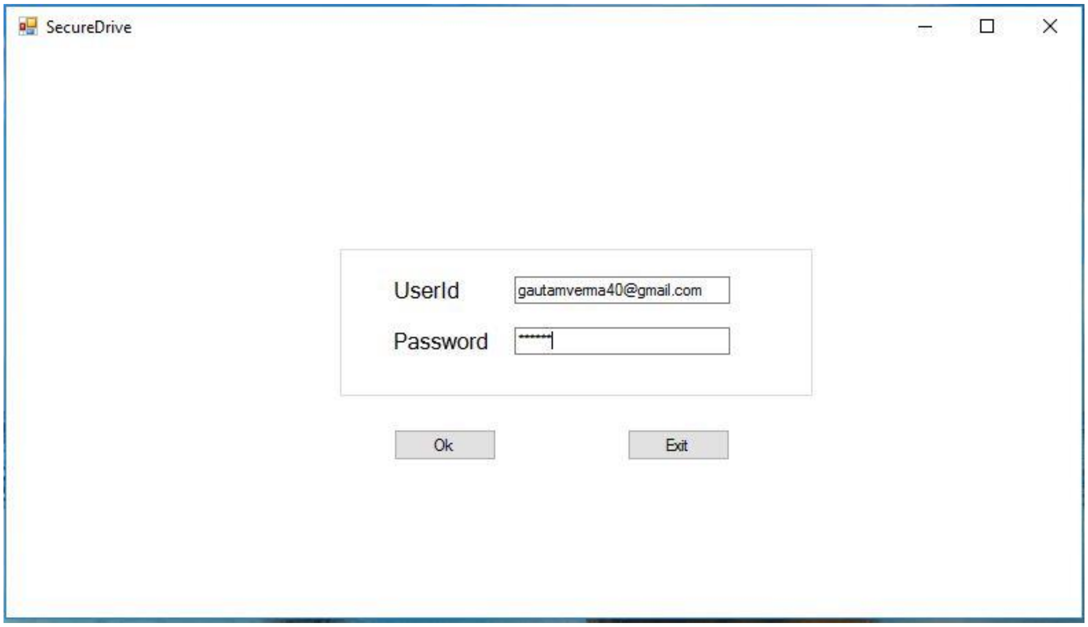
Processor: Intel Core i5, 2.1 GHz

Memory: 4GB

OS: Windows 10

5.3. Output

The Figure 5.1 shows the login screen which is used by user to login to the system.

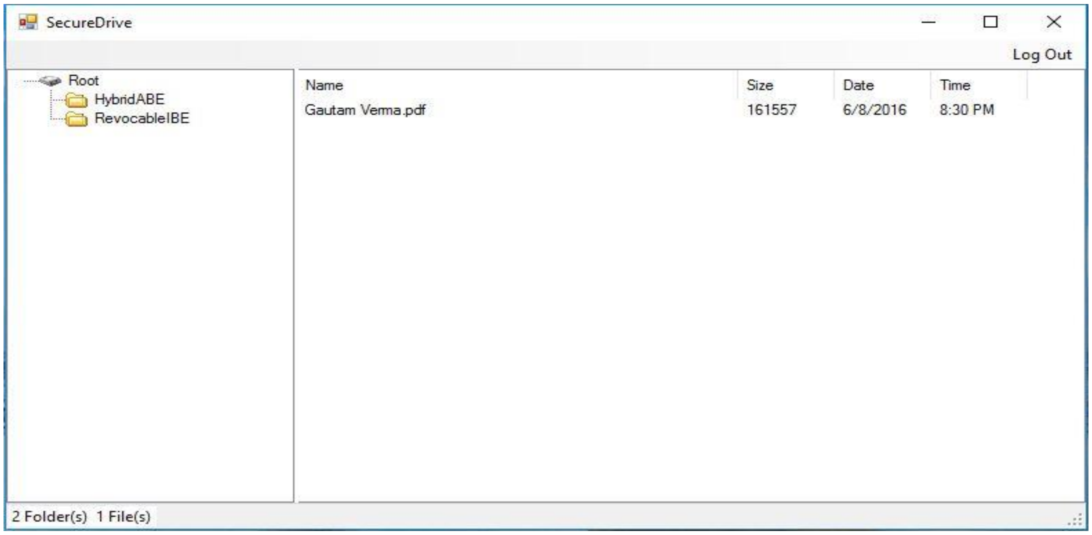


UserId gautamverma40@gmail.com

Password ******

Figure 5.1: Login Screen

The Figure 5.2 shows the file explorer that displays the files stored on Google Drive.

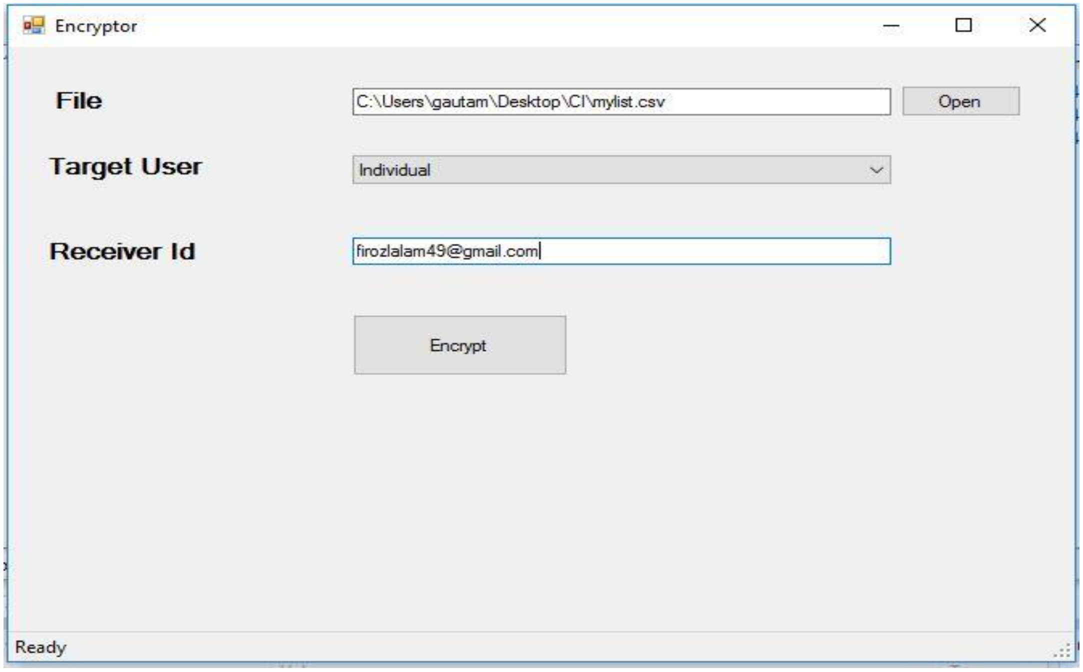


Name	Size	Date	Time
Gautam Verma.pdf	161557	6/8/2016	8:30 PM

2 Folder(s) 1 File(s)

Figure 5.2: File Explorer for Cloud Storage

The Figure 5.3 shows how the system is used to encrypt a file for a single recipient.



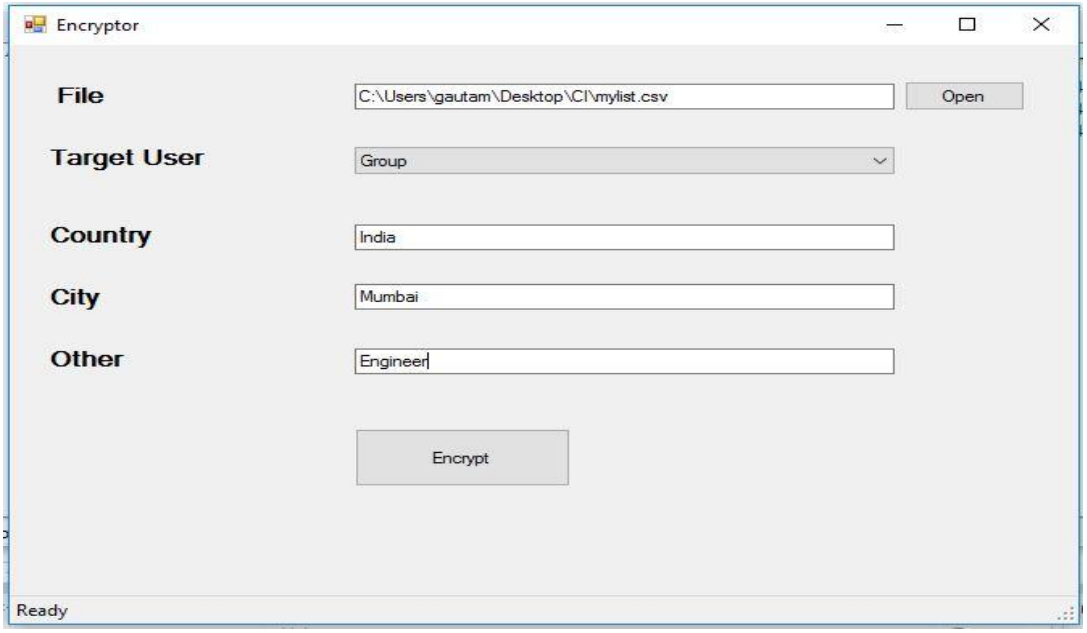
File: C:\Users\gautam\Desktop\CI\mylist.csv

Target User: Individual

Receiver Id: firozlalam49@gmail.com

Figure 5.3: Encryption for single recipient

The Figure 5.4 shows how the system is used to encrypt a file for a group of recipients.



File: C:\Users\gautam\Desktop\CI\mylist.csv

Target User: Group

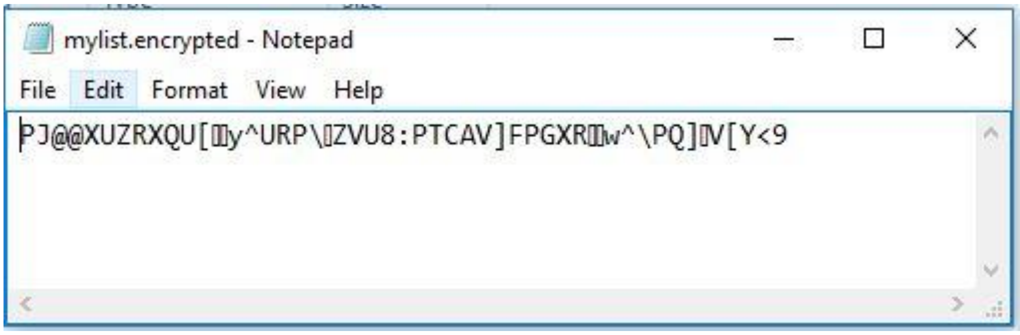
Country: India

City: Mumbai

Other: Engineer

Figure 5.4: Encryption for group of recipients

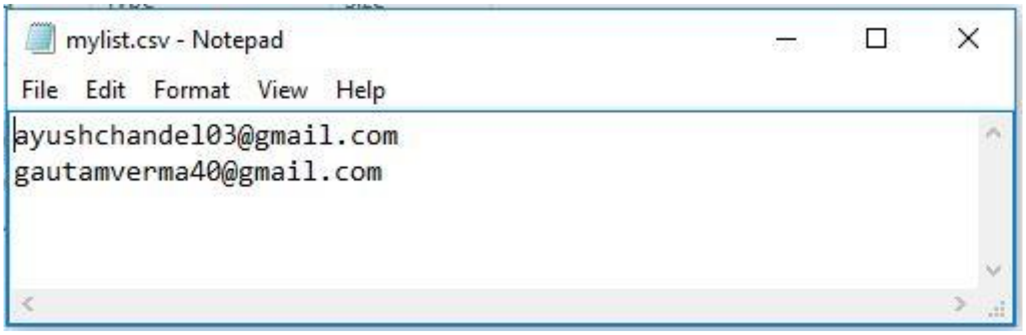
The Figure 5.5 shows the content of an encrypted file.



```
PJ@@XUZRXQU[□□y^URP\□ZVU8:PTCAV]FPGXR□□w^\PQ]□V[Y<9
```

Figure 5.5: Encrypted File

The Figure 5.6 shows the content of an encrypted file after decryption.



```
ayushchandel03@gmail.com
gautamverma40@gmail.com
```

Figure 5.6: Decrypted File

CHAPTER 6

RESULTS AND ANALYSIS

In this Chapter, we will give an intensive test assessment of the technique proposed in Chapter 4. Note that in each one of the assessments, the groups G_1 and G_2 are chosen with length of 256 bit and 512 bit separately.

6.1. Performance Analysis of Whole Construction

Primarily, our goal is to compare the aggregate time taken during each phase of our technique with that of the original IBE [21] to assess the effectiveness of our technique.

Table 6.1 shows how much time it cost to run a specific phase of both techniques. Since we have taken into account the revocability problem it was obvious that our technique will take additional time. The setup module of our technique is similar to that of the IBE technique in [21]. Also the key-issuing phase of IBE technique [21] is comparatively shorter than that of our technique. The factor responsible for this is the insertion of time-period part into private key of every client to permit periodic update with revocation, due to which extra calculations are required in our technique to initialize this module. It is not astounding that our technique also takes marginally longer time for encryption and decryption than the IBE technique [21] because the time part is present. Extra encryption/decryption is carried out by client for time part, instead of simply encrypting/decrypting the identity part.

To briefly summarize, in comparison to the first IBE technique [21] our technique accomplishes both revocability and encipherment/decipherment without incurring huge cost.

Table 6.1: Efficiency Comparison for Various Phases

Phase	Our Scheme	IBE Without Revocation
Setup	87.34 ms	82.32ms
Key-Issuing	43.06 ms	22.23 ms
Encryption	43.54 ms	29.94 ms
Decryption	22.72 ms	12.82 ms
Key_Update	15.2 ms	

6.2. Performance Analysis for Revocation

Secondly, to demonstrate a broad comparison between our technique and revocable IBE technique we reenacted the multi-user revocation situation. We examined by taking key-refresh phase and the key-issuing phase into account.

6.2.1. Key-Issuing Phase

In Figure 6.1, we show the replying time for a single key generation request by changing the maximum number of users in the scheme. In IBE scheme [18], a binary tree is used to manage all the users, and a single user is represented by a leaf node of the tree. PKG has to carry out calculation on each and every node in the path from the target leaf node to root node during key-issuing. So it is quite obvious that the replying time of this scheme is in proportion of O $(\log_2 M)$ where M is the number of users in the scheme. To issue a single key, the efficiency of [18] is logarithmically increasing, while the efficiency of our scheme is constant.

Similarly, in Figure 6.2 we demonstrate the efficiency of our scheme on the basis of the size of the private key. While our scheme has constant size of private key, the size of private key in scheme [18] grows with the number of the users because of the same cause highlighted above.

Our scheme results in improvement in size of private key and efficiency. Also the scheme [18] does not allow changing the number of users while our scheme allows dynamic number of users. In other words, since the previous scheme [18] uses binary tree to represent users it need to fix the maximum number of users in system. So it is hard to increase number of user then the specified bound once the maximum number is fixed. Our scheme is flexible at it provisions dynamic controlling of users.

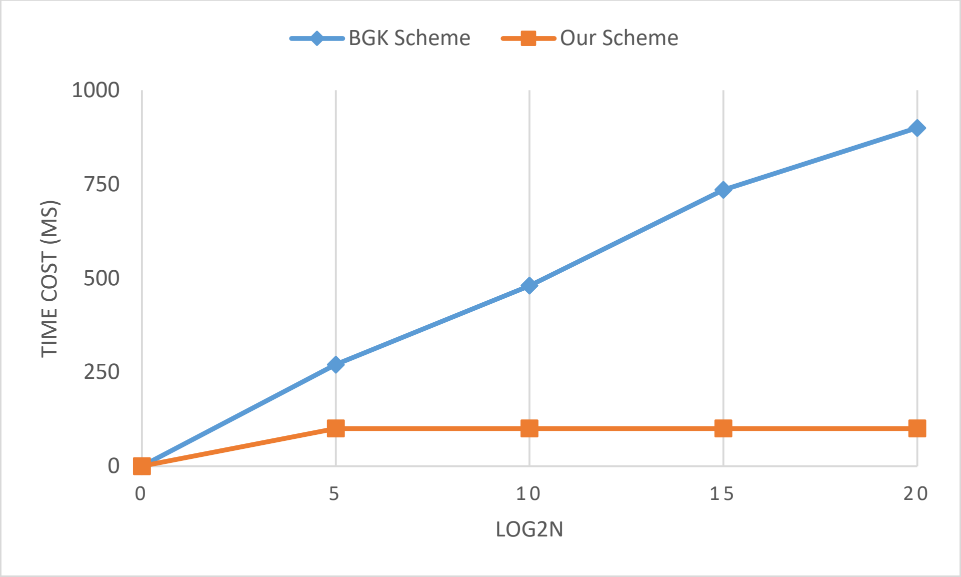


Figure 6.1: Responding time for Single Key-Generation Request

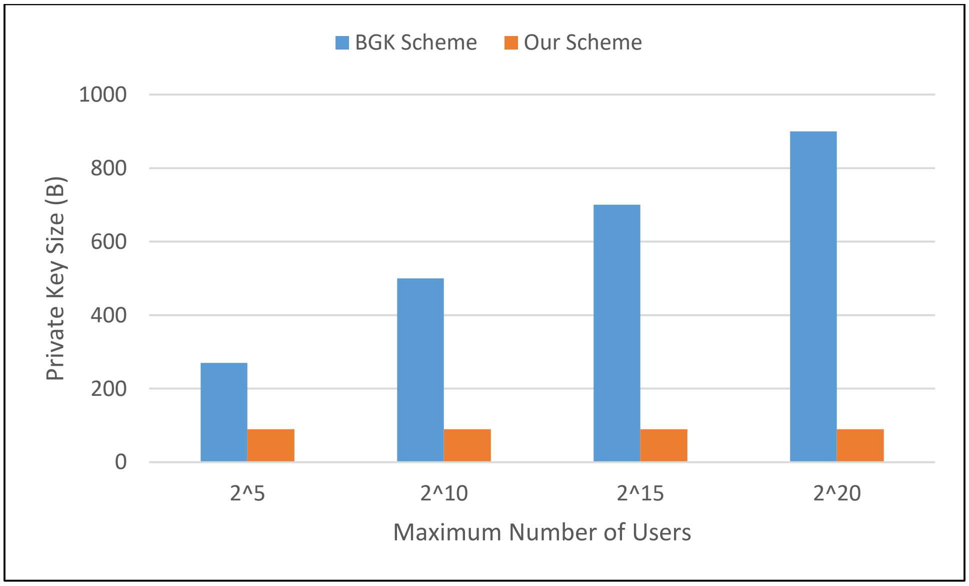


Figure 6.2: Private Key Size

6.2.2. Key-Update Phase

To evaluate the aggregate time of updating private keys for non-revoked users we arbitrarily select 5% to 75% of users to be revoked. To simplify the test, we will simply show an example of user revocation in Figure 6.3 and evaluate the aggregate time in updating key at PKG. We can see that the efficiency graph of IBE system [18] is parabolic in shape, and in our assessment efficiency reaches the lowermost value at the 25% revocation ratio. This is due to the fact that it is the gap that the leaf nodes to be revoked has a huge number but small aggregation degree, which necessitates updating lots of intermediate nodes for updating keys. On the other hand, in our system, behavior like this is prevented, and PKG requires just an insignificant uniform time. In general, our system achieved this fixed efficiency in updating keys irrespective of the number of users because we have outsourced the revocation to KU-CSP, but in BGK scheme [18] the time cost increases with increase in the number of users. In view of that, we also demonstrate the time taken for private keys renewal at KU-CSP for all the unrevoked users in our system with revocation ratio varying from 5% to 75% as shown in Figure 6.4. Nevertheless, it should be pointed out, that such a time cost is increasing with the increase in number of users in each case. However, in contrast to the calculation performed at PKG in [18], in our system these calculations are performed at cloud with plenty of resources. Additionally, for each key-update request of a user we calculated the communication cost which is 87 ms. Observe that these overhead at cloud provider contains the verification and transmission time.

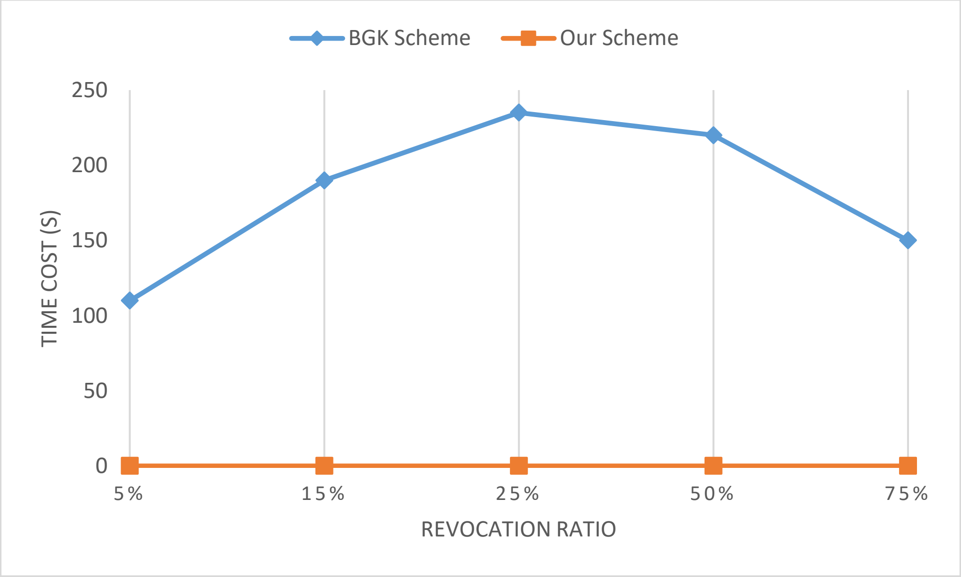


Figure 6.3: Key Update at PKG

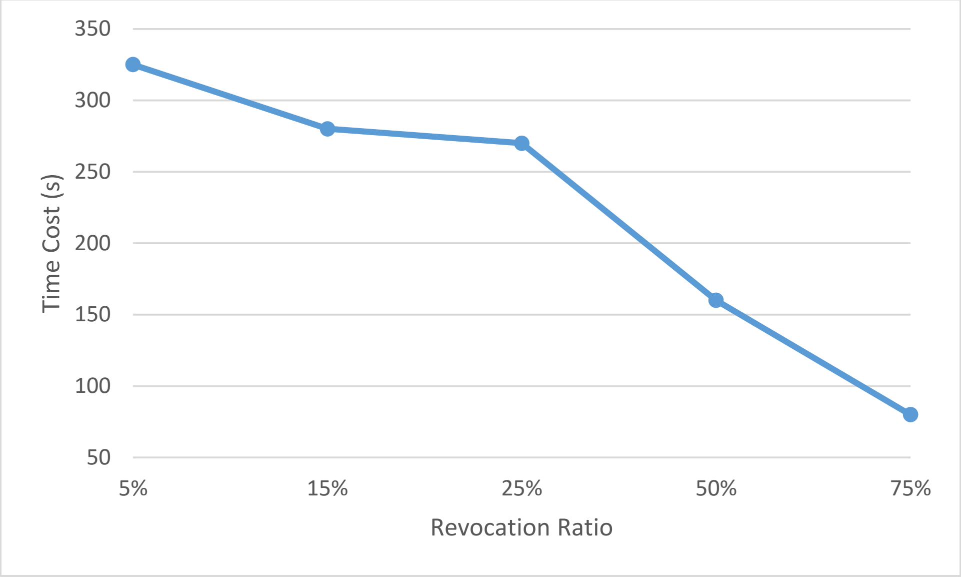


Figure 6.4: Key-Update at KU-CSP

6.3. Performance Analysis for Outsourcing

Thirdly, we will weigh the efficiency of our system against the straight revocable system [21] which uses an equivalent revocation idea as ours but does not take into consideration outsourcing to demonstrate the efficiency of outsourcing calculation in our system. Remember that in [21] Boneh et al. proposed that senders utilize the identities of recipients appended with present time period to encrypt data for recipient and all the users update their private keys regularly. In contrast to the work done in [21], to realize efficient revocation we delegated the calculation cost at PKG to KU-CSP. The outcomes of outsourcing is demonstrated in Figure 6.5.

It can be seen clearly that the calculation cost at PKG in Boneh's revocable system [21] and KU-CSP in our system is more or less equal. This is for the reason that we constructed identity element and time element in our system, and original private key form in [21] has a comparable form with each element in our system. Therefore, the calculation required for updating time element and re-issuing private key during the key update process are approximately equal. We need to highlight that such calculations are normally handled by KU-CSP with plentiful assets so it will not gravely influence the effectiveness of our scheme.

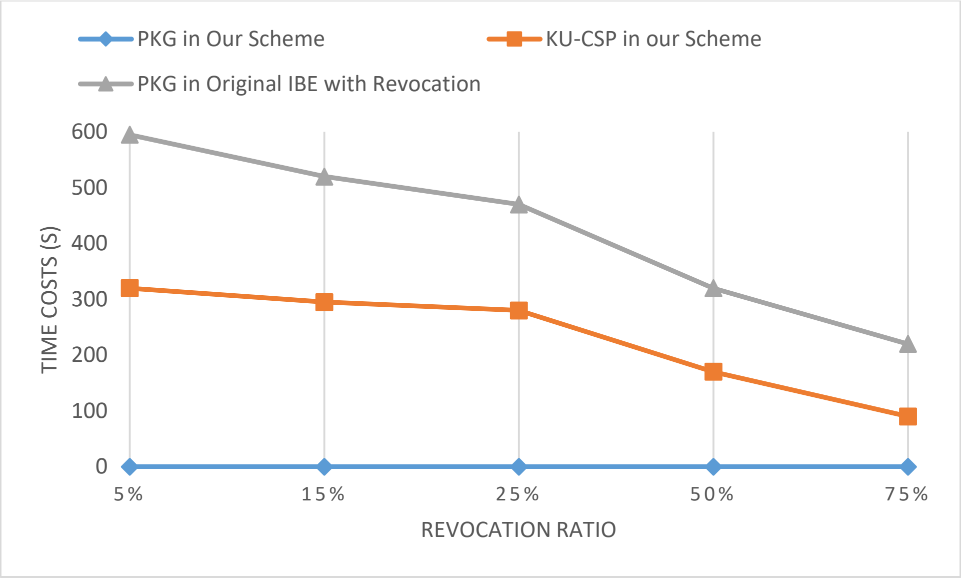


Figure 6.5: Key Update time

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis the main goal is to overcome the current research problem of efficient revocation while improving the security level of IBE method. We proposed the H-IBE method based on outsourcing computation into the Attribute based IBE method. In addition to this, we proposed the revocable technique in which the revocation functionalities are assigned to CSP. The functions keygen, encrypt, decrypt, revoke and key-update are designed, modified and implemented in this thesis. The performance is evaluated to claim the efficiency of proposed method. The revocation efficiency is improved as compared to existing method.

In this thesis, we propose outsourcing computation into IBE to deal with the important problem of identity revocation, and suggest a revocable scheme in which CSP handles all the actions related to the revocation process. The introduction of KU-CSP in the system has helped in making the suggested scheme full-fledged: 1) Both calculation performed at PKG and size of the private key at user are possible with constant efficiency; 2) There is no need for the PKG to be online after delivering the revocation list to KU-CSP. In other words to update his/her key user needs to contact with KU-CSP and not with PKG; 3) Public channel can be used between KU-CSP and user during the key-update process; 4) It results in storage saving because a single encrypted file can be used by a group of users. Finally, to make evident the effectiveness of our suggested scheme we provided broad experimental outcomes.

For future work, we suggest to work on in detail practical analysis and testing to check the possibilities of further improvements.

CHAPTER 8

REFERENCES

[1] Q. Xing et al., "Unbounded Revocable Hierarchical Identity-Based Encryption with Adaptive-ID Security," IEEE 18th International Conference on High Performance Computing and Communications, NSW, pp. 430-437, 2016.

[2] Y. Jiang and M. Du, "Provable Security Analysis on Unbounded Hierarchical Identity-Based Encryption and Attribute-Based Encryption," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, pp. 510-513, 2016.

[3] Jianchang Lai, Yi Mu, Fuchun Guo, "Efficient identity-based online/offline encryption and signcryption with short ciphertext", International Journal of Information Security, pp. 1-13, 2016.

[4] Damien Vergnaud, M. Zheng, Y. Xiang, H. Zhou, "Comment on a strong provably secure ibe scheme without bilinear map" in Journal of Computer and System Sciences, pp. 125-131, 2016.

[5] J. Wei, W. Liu, X. Hu, "Forward-secure identity-based signature with efficient revocation", Int. J. Comput. Math., vol. 93, pp. 1-23, 2016.

[6] D Kalyani, R Sridevi, "Survey on Identity based and Hierarchical Identity based Encryption Schemes", International Journal of Computer Applications, vol. 134, no. 14, pp. 0975-8887, January 2016.

[7] J Seo, K Emura, "Adaptive-ID Secure Revocable Hierarchical Identity-Based Encryption", IWSEC 2015 LNCS 9241, pp. 21-38, 2015.

[8] J.-H. Seo and K. Emura, “Revocable hierarchical identity-based encryption: history-free update, security against insiders, and short Ciphertexts,” Proc. CT-RSA’15, LNCS, vol. 9048, pp. 106-123, 2015.

[9] S. Park, K. Lee, and D.H. Lee, “New constructions of revocable identity-based encryption from multilinear maps,” IEEE Transactions on Information Forensics and Security, vol.10, no. 8, pp. 1564 - 1577, 2015.

[10] J. Li, Y. Shi, and Y. Zhang,”Searchable ciphertext-policy attributebased encryption with revocation in cloud storage,” International Journal of Communication Systems, article in press (DOI: 10.1002/dac.2942), 2015.

[11] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, “Identity-based encryption with outsourced revocation in cloud computing,” IEEE Trans. On Computers, vol. 64, no. 2, pp. 425-437, 2015.

[12] C. Wang, Y. Li, X. Xia, and K. Zheng, “An efficient and provable secure revocable identity-based encryption scheme,” PLoS ONE, vol. 9, no. 9, article: e106925, 2014.

[13] J.-H. Seo and K. Emura, “Efficient delegation of key generation and revocation functionalities in identity-based encryption,” Proc. CT-RSA’13, LNCS, vol. 7779, pp. 343-358, 2013.

[14] J.-H. Seo and K. Emura, “Revocable identity-based encryption revisited: security model and construction,” Proc. PKC’13, LNCS, vol. 7778, pp. 216-234, 2013.

[15] S. Hohenberger and B. Waters,”Attribute-based encryption with fast decryption,” Proc. PKC’13, LNCS, vol. 7778, pp. 162-179, 2013.

[16] Y.-M. Tseng. and T.-T. Tsai, “Efficient revocable ID-based encryption with a public channel,” Computer Journal, vol.55, no.4, pp. 475-486, 2012.

[17] B. Libert and D. Vergnaud, “Adaptive-ID secure revocable identity-based encryption,” Proc. CT-RSA’09, LNCS, vol. 5473, pp. 1-15, 2009.

[18] Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryption with efficient revocation,” Proc. CCS’08, pp. 417-426, 2008.

[19] V. Goyal, O. Pandey, A. Sahai, and B. Waters,”Attribute-based encryption for fine-grained access control of encrypted data,” Proc. ACM CCS, pp. 89-98, 2006.

[20] Sahai and B. Waters, “Fuzzy identity-based encryption,” Proc. Eurocrypt’05, LNCS, vol. 3494, pp. 557-557, 2005.

[21] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” Proc. Crypto’01, LNCS, vol. 2139, pp. 213-229, 2001.

[22] D. Boneh, X. Ding, G. Tsudik, and C.-M. Wong, “A Method for fast revocation of public key certificates and security capabilities,” Proc. 10th USENIX Security Symp., pp. 297-310. 2001.

[23] X. Ding and G. Tsudik, “Simple identity-based cryptography with mediated RSA,” Proc. CT-RSA’03, LNCS, vol. 2612, pp. 193-210, 2003.

[24] B. Libert and J. J. Quisquater, “Efficient revocation and threshold pairing based cryptosystems,” Proc. PODC2003, pp. 163-171, 2003.

[25] J. Baek and Y. Zheng, “Identity-based threshold decryption,” Proc. PKC’04, LNCS, vol. 2947, pp. 262-276, 2004.

[26] H.-S. Ju, D.-Y. Kim, D.-H. Lee, H. Park, and K. Chun, “Modified ID-based threshold decryption and its application to mediated IDbased encryption,” Proc. APWeb2006, LNCS, vol. 3841, pp. 720-725, 2006.

[27] Lewko A and B. Waters, “New techniques for dual system encryption and fully secure hibe with short ciphertexts,” Proc. TCC’10, LNCS, vol. 5978, pp. 455-479, 2010.

[28] Shamir, “Identity-based cryptosystems and signature schemes,” Proc. Crypto’84, LNCS, vol. 196, pp. 47-53, 1984.

[29] Ryu Geumsook et al., Unbounded Hierarchical Identity-Based Encryption with Efficient Revocation. Information Security Applications, Springer International Publishing, 2015.

[30] Y. Mao, "Fully Secure Fuzzy Identity-Based Encryption for Secure IoT Communications", Computer Standards & Interfaces, 2015.

[31] H. Qian, J. Li, Y. Zhang, and J. Han, ”Privacy preserving personal health record using multi-authority attribute-based encryption with revocation,” International Journal of Information Security, vol. 14, no. 6, pp. 487-497, 2015.

[32] Y. Zhou, D. Feng, W. Xia, M. Fu, F. Huang, Y. Zhang, C. Li, "Secdep: A user-aware efficient fine-grained secure deduplication scheme with multi-level key management", 2015 31st Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-14, 2015.

[33] Graham Enos, Yuliang Zheng, "An id-based sign-cryption scheme with compartmented secret sharing for unsigncryption", Information Processing Letters, vol. 115, no. 2, pp. 128-133, 2015.

[34] Guiyi Wei, Jun Shao, Yang Xiang, Pingping Zhu, Rongxing Lu, "Obtain confidentiality or/and authenticity in big data by id-based generalized signcryption", Information Sciences, vol. 318, pp. 111-122, 2015.

[35] Minghui Zheng, Yang Xiang, Huihua Zhou, "A strong provably secure ibe scheme without bilinear map", Journal of Computerand System Sciences, vol. 81, no. 1, pp. 125-131, 2015.