Towards Secure Identity-based Cryptosystems for Cloud Computing

By

Waleed Khalid Hassan Al-Jumyli

School of Computing

The University of Buckingham

United Kingdom

A Thesis

Submitted for the degree of Doctor of Philosophy in Computer Science to the school of Computing in the

University of Buckingham

January 2020

ABSTRACT

The convenience provided by cloud computing has led to an increasing trend of many business organizations, government agencies and individual customers to migrate their services and data into cloud environments. However, once clients' data is migrated to the cloud, the overall security control will be immediately shifted from data owners to the hands of service providers. When data owners decide to use the cloud environment, they rely entirely on third parties to make decisions about their data and, therefore, the main challenge is how to guarantee that the data is accessible by data owners and authorized users only.

Remote user authentication to cloud services is traditionally achieved using a combination of ID cards and passwords/PINs while public key infrastructure and symmetric key encryptions are still the most common techniques for enforcing data security despite the missing link between the identity of data owners and the cryptographic keys. Furthermore, the key management in terms of the generation, distribution, and storage are still open challenges to traditional public-key systems.

Identity-Based Cryptosystems (IBCs) are new generations of public key encryptions that can potentially solve the problems associated with key distribution in public key infrastructure in addition to providing a clear link between encryption keys and the identities of data owners. In IBCs, the need for pre-distributed keys before any encryption/decryption will be illuminated, which gives a great deal of flexibility required in an environment such as the cloud. Fuzzy identity-based cryptosystems are promising extensions of IBCs that rely on biometric modalities in generating the encryption and decryption keys instead of traditional identities such as email addresses.

This thesis argues that the adoption of fuzzy identity-based cryptosystems seems an ideal option to secure cloud computing after addressing a number of vulnerabilities related to user verification, key generation, and key validation stages. The thesis is mainly concerned with enhancing the security and the privacy of fuzzy identity-based cryptosystems by proposing a framework with multiple security layers. The main contributions of the thesis can be summarised as follows.

1. Improving user verification based on using a Challenge-Response Multifactor Biometric Authentication (CR-MFBA) in fuzzy identity-based cryptosystems that reduce the impacts of impersonators attacks.
2. Reducing the dominance of the "trusted authority" in traditional fuzzy identity-based cryptosystems by making the process of generating the decryption keys a cooperative process between the trusted authority server and data owners. This leads to shifting control over the stored encrypted data from the trusted authority to the data owners.
3. Proposing a key-validity method that relies on employing the Shamir Secret Sharing, which also contributes to giving data owners more control over their data.
4. Further improving the control of data owners in fuzzy identity-based cryptosystems by linking the decryption keys parameters with their biometric modalities.
5. Proposing a new asymmetric key exchange protocol based on utilizing the scheme of fuzzy identity-based cryptosystems to shared encrypted data stored on cloud computing.

Dedicated to

My father's soul and my family

ACKNOWLEDGEMENTS

*First and foremost, I would like to thank **Allah Almighty the Most Gracious and Merciful** for giving me the strength, knowledge, ability and opportunity to undertake my PhD study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible, and it would not have seen the light.*

*I wish to acknowledge the support and great love of my big family, **my mother**, **my brothers**, **Mohammed** and **Ahmed**, and all my **sisters**. They kept me going on and this work would not have been possible without their supporting.*

*I would like to express my gratitude and appreciation for **my wife**; **my sons**, **Aws and Anas**; **my beloved daughter**, **Tuqa** whose guidance, support, patience and encouragement has been invaluable throughout this study.*

*I wish to express my sincere appreciation to my supervisor, **Dr Hisham Al-Assam**, who has the substance of a genius: he convincingly guided and encouraged me to be professional and do the right thing even when the road got tough. Without his persistent help, the goal of this thesis would not have been realized.*

*Special thanks go to **Mrs Sharon Salerno, Mrs Jayne Kelly** and **Mrs Alison Wood**, Administrator of Psychology and School of Computing for their support and assistance.*

*I have to thank all my friends who have supported and encouraged me during my study. Special thanks go to **Dr Nasiru Ibrahim**, **Dr Omar Al-okashi**, **Mr Tahseen Al-baidhani**, **Mr Qahtan Al-qaisi**, **and Mr Tarek Kabel.***

*Finally, Special thanks for the financial support from **the Government of Iraq / Ministry of Higher Education and Scientific Research Iraq**, **University of Anbar**, as well as the all the staff of the **Iraqi Cultural Attaché in London** who granted and support my study.*

ABBREVIATIONS

Word		Abbreviation
1-NN	:	Simple Nearest Neighbour
ABAC	:	Attribute-based Access Control
ACL	:	Access Control List
ACM	:	Access Control Mechanism
AWS	:	Amazon Web Services
BCs	:	Biometric Cryptosystems
BDK	:	Bob's Decryption Key
BL	:	Biometric Lock
BSs	:	Biometric Systems
BVF	:	Binary Features Vector
CA	:	Certificate Authority
CBDS	:	Cloud-based Data Storage
CDH	:	Computational Diffie-Hellman
CM	:	Challenge Message
CP	:	Complementary Part
CPA	:	Central Point of Attack
CP-ABC	:	Ciphertext-Policy ABC
CSP	:	Cloud Service Provider
CV	:	Challenge Vector
DAC	:	Discretionary Access Control
DDH	:	Decisional Diffie-Hellman
DH	:	Diffie-Hellman
DHKE	:	Diffie-Hellman Key Exchange
DKG	:	Decryption Key Generation
DLP	:	Discrete Logarithm Problem
DMBDH	:	Decisional Modified Bilinear Diffie-Hellman
DOs	:	Data Owners
DWT	:	Discrete Wavelet Transform
EC	:	Elliptic Curves
EC2	:	Amazon Elastic Compute Cloud
ECC	:	Error Code Correction
ECDLP	:	Elliptic Curve Discrete Logarithm Problem
EER	:	Error Equal Rate
FAR	:	False Accept Rate
F-IBCs	:	Fuzzy- IBCs
FRR	:	False Reject Rate
F-SID	:	Fuzzy Selective-Identity
FVs	:	Feature Vectors
FVSs	:	Fuzzy Vault Systems
GCE	:	Google Compute Engine
GCF	:	Google Cloud Platform
IaaS	:	Infrastructure as a Service
IBCs	:	Identity-based cryptography systems
ICT	:	Information and Communication Technology
IT	:	Information Technology
KBSs	:	Key Binding Systems

KCG	:	Key Centre Generator
KE-BIBC	:	Key Exchange Using Biometric Identity Based Cryptography
KGSs	:	Key Generation Systems
KP-ABC	:	Key-Policy ABC
LL	:	Low Low (lowest-pass wavelet subband)
MAC	:	Mandatory Access Control
MPPs	:	Master Public Parameters
MSPs	:	Master Secret Parameters
NIST	:	National Institute of Standards and Technology
Ob	:	Object
ORL	:	Olivetti Research Lab
ORP	:	Orthonormal Random Projection
OTCR-MFA	:	One-Time Challenge-Response Multifactor Authentication
OTRS	:	One-Time Random Secret
PaaS	:	Platform as a Service
PAP	:	Policy Administration Point
PDP	:	Policy Administration Point
PDP	:	Policy Decision Point
PEP	:	Policy Enforcement Point
PHRs	:	Personal Health Records
PIN	:	Personal Identification Number
PIP	:	Policy Information Point
PKEs	:	Public Key Encryption systems
PKG	:	Private Key Generator
PKG-BKE		PKG base Key Escrow
PKI	:	Public Key Infrastructure
PL	:	Permission List
RAs	:	Role Assignment
RAu	:	Role Authorization
RBAC	:	Role-Based Access Control
RS	:	Reed Solomon
RM	:	Response Message
ROI	:	Region of Interest
RV	:	Response Vector
SaaS	:	Software as a Service
SID	:	Selective-Identity
SLAs	:	Service Level Agreements
Su	:	Subject
TA	:	Transaction Authorization
XACML	:	Extensible Access Control Markup Language
XML	:	Extensible Markup Language
$CBio_u$	:	Cancellable Biometric Template of the user u

TABLE OF CONTENTS

List of Figures

List of Tables

DECLARATION

*I hereby declare that all the work in my thesis entitled "**Towards Secure Identity-based Cryptosystems for Cloud Computing**" is my own work except where due reference is made within the text of the thesis.*

*I also declare that, to the best of my knowledge, none of the material has ever previously been submitted for a degree in the **University of Buckingham** or any other university.*

Waleed Khalid Hassan Al-Jumyli

Chapter 1
Introduction

1.1 Research Background

The rapid growth of the Internet has led to a revolution in the delivery of information and communications technology (ICT) services and the development of scalable digital infrastructures. The development of software has long required combined skills in programming, design, and management of systems. In the event an integral development, application servers and databases were expected to support the different projects, which means that there is a need to install, configure and maintain these parts. Such tasks can take a long time to design and build creative technology. What if we can get it out or at least pass the burden to a service provider, thus eliminating the burdens of capital investments and providing these services through a common IP-based infrastructure? That is cloud computing [1].

Cloud computing offers various services, such as applications, development tools, storage solutions, servers and network services that rely on "on-demand services" that allow customers to take advantage of them under the "pay-for-use" model. It enables companies to pay only for the services they consume from the cloud, and this model relaxes companies from cost concerns, hardware failures, software upgrades, outdated technologies, as well as migration, cost and the availability of appropriate information and communication technology skills. Cloud computing is a web-based technology, which stores and maintains user data away in cloud providers' area, for example, Salesforce.com, Google Compute Engine (GCE), Google Cloud Platform (GCF), Amazon Web Services (AWS), Microsoft Azure[2]; all are provided by third trusted parties, called Cloud Service Providers (CSPs).

Cloud-based data storage application is considered as one of the essential services offered by cloud computing (see Figure 1.1). It provides scalable storage space with cost-effectiveness compared with those could be obtained in traditional IT.

Regarding data owners, the transfer of data to the cloud computing environment would also lead to serious challenges— once the data is transferred to cloud computing, the full control over that data will also be shifted from data owners to CSPs. Hence, the data owners cannot figure out who can access, use or tamper their stored data. In other words, CSPs have full control over the services provided. Figure 1.1 describes the usage of storage service for different types of cloud storage applications [3].

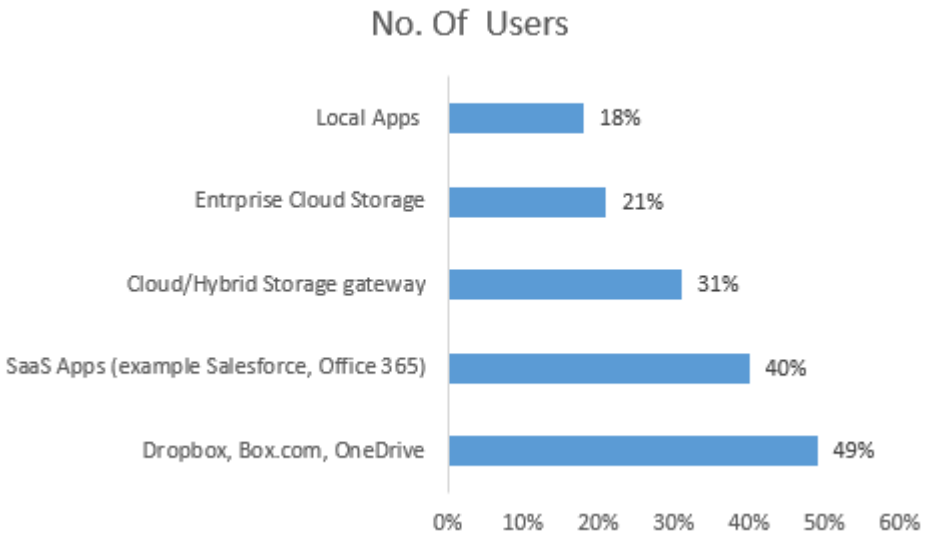


Figure 1.1: Usage rate for different cloud storage application "adopted from [3]"

Although all good things can be offered to cloud users, cloud environment also has emerging security and privacy concerns, which makes it hard for some businesses with sensitive data storage to adopt cloud computing. M. Zhou et al. [4] pointed out that a significant factor of deciding whether to opt for cloud services depends primarily on the security and privacy issues. Data may involve sensitive information such as user's identities, personal health records (PHRs), bank information, and so on, and this raises the concerns of disclosure or access by unauthorised parties. According to a study carried out by Cyber Security in 2011 about 209 international companies that found about 37% of the data had been breached by malicious attacks and the average cost of one record up to $222[2]. The study showed that Amazon's Zappos has also compromised, and their data was breached, which affected 24,000,000 users, included the disclosure of names, e-mail addresses, telephone numbers, in addition to exposing important banking information. There were also approximately 440 million compromised customers records (about 200GB) at Cloud Data Management Company Veeam Software Inc. and became publicly known over the internet[5].

Further details about cloud data incidents recorded in mid-2012, when Dropbox issued a warning to its customers and confirmed the need to change the password immediately. The report said details of more than 68 million user accounts had been compromised. It also reported that the data involved email addresses as well as the hash algorithm that utilised to safeguard the user' passwords[6]. However, the following figure adopted from the Identity Theft Resource Centre's report (ITRC) shows who data breaches were split (percentages) across different sectors up to 2019 [7][8][9].

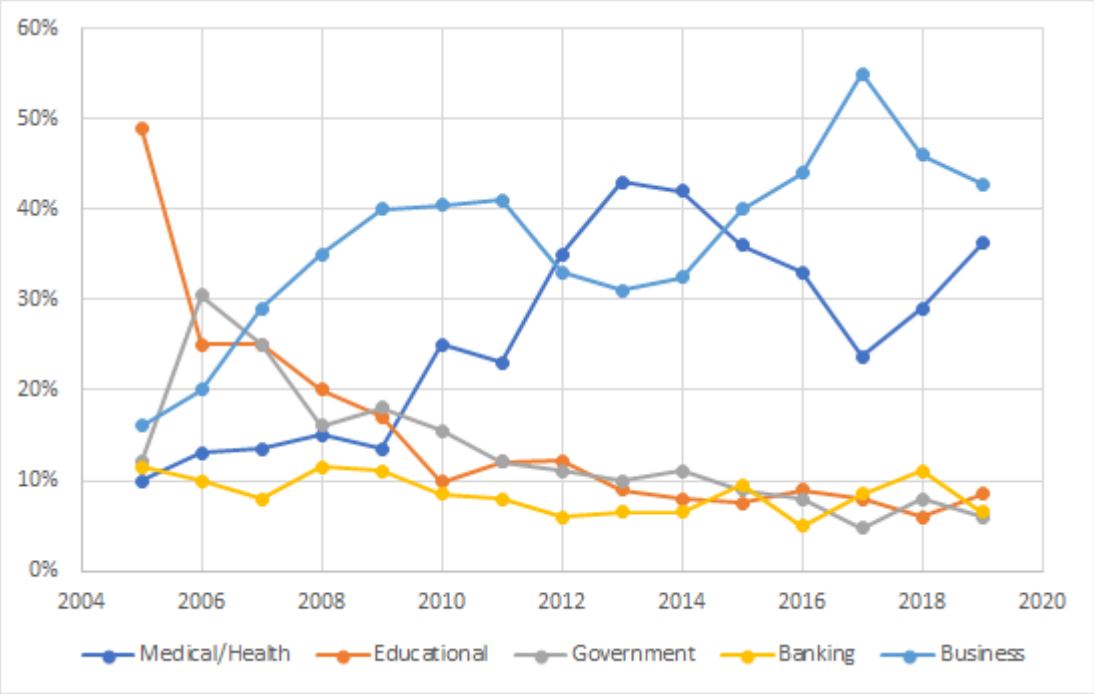


Figure 1.2: The percentage of Data compromised by Industrial sector" adopted from [7]"

As shown in the above figure, the percentage of a data breach in business in 2017 was significantly higher than in 2016 [7][8][9].

Meanwhile, since data and services are not on the same organisation's site, adding further concerns and complexities regarding the nature of the access control mechanisms [10]. Access control is a set of techniques that data owners rely on to preserve their data from any attempt may conduct to access by unauthorised entities. In general, it is difficult to guarantee that the cloud user gains proper access control over their data like traditional data storage. The common practice in securing data on the cloud suggests that the most appropriate approach is to encrypt the data before migrating it from the user's site to the cloud. The presence of data in an encrypted form protects it from any tampering that may occur by an intruder or services providers. This approach has been adopted in the vast majority of studies that aim to protect the data of cloud users[11][12][13][14][15][16].

1.2 Problem Statement of My Research

Traditionally, the cryptography community has defined two main types of conventional cryptography, symmetric and asymmetric keys also known as public key encryption systems (PKEs). How securely store and exchange of the encryption/decryption keys among the entities is the major challenge involved in using symmetric/asymmetric keys, especially in the open environment, for example, cloud computing. For the session-key exchange of a lot of web services, the PKEs offer an appropriate practical solution. The need for a trusted third party (e.g. certificate authority) is not only the key challenge of PKEs solutions but also the missing link between the data owner and the encryption keys. The latter is arguably more critical where accessing data demands to be connected with the identity of the owner. Likewise, adopting trusted couriers or secure channels is a key requirement of the existing available key exchange protocols. These protocols usually can be subjected to a man-in-the-middle attack, as well as various other attacks. New generations of PKEs, known as identity-based cryptosystems (IBCs), have been introduced to overcome the problem of not linking the key to identity. IBCs were suggested in [2] as an advanced scheme of PKEs that ignored the needed of certificate authority as long as the keys are derived from the user's identity (e.g., email addresses, telephone numbers, driver licenses). IBC schemes traditionally have three main entities, a sender, a recipient, and a third trusted authority known as Private Key Generator (PKG). Two main parameters, master public parameters (MPPs) and master secret parameter (MSP), are generated by PKG. MPPs will be known by all, while MSP is known only by the PKG. The essential ingredients to produce decryption key are MSPs in addition to the users' identities and, therefore, these parameters are so important. An alternative advanced model of IBCs was proposed by Sahai and Waters in [17] to adopt users' biometrics instead of the traditional users' identities, which known as fuzzy identity-based cryptosystems (F-IBCs).

In this thesis, we argue that although F-IBC could potentially be an excellent solution to secure cloud applications, the way in which PKG manages the MSP inherited a range of challenges that can be summarized as follow:

1. F-IBCs-based impersonating Attacks (External): It can be argued that existing F-IBC systems have a serious security vulnerability related to releasing decryption keys without proper user authentication. In fact, security relies on the assumption that biometrics can be only presented by genuine users/owners, which is

unrealistic. The fact that F-IBCs deal with public biometrics data of a user allows impersonators to attack the system by making use of the availability of public biometric (e.g. Social media platforms) of a genuine user to be used for receiving the corresponding decryption keys of the user.

2. IBC-based Key-escrow (Internal Attack)[18]: A key-escrow is a property related to traditional PKEs that allows authorised individuals, e.g., persons, officials of an enterprise and the government, to decrypt the encrypted messages under certain circumstances and according to the retrieval criteria via one or more trusted entities that stores the recovery keys [19] [20]. However, the key-escrow in IBCs is inherited due to the exclusive management of MSPs by PKG. It, therefore, gives PKGs the ability to decrypt all the encrypted data even without the data owners' permissions. Also, PKGs can grant access and read encrypted data to any unauthorised parties without the consent of the data owners.
3. The central point of attacks (External attack): This challenge is related to the way that MSP is stored. Because all MSPs are stored on a single database (PKG's database) in IBCs and F-IBCs, this could lead to a domino effect. Therefore, once an attacker compromises this site, the whole encrypted message is decrypted.
4. Lack of control: Once the data owners used IBCs, they will lose their control over their data. They have no control on who and when their data will be accessed.

At present, the questions this thesis tries to address are

- Can we integrate the IBC and users' biometric data to bring the control back again to data owners by providing an effective access control mechanism that determines who can get access to the data?
- Can we use the recipient's biometrics data to support F-IBCs in such a way that withstands the impersonator's attacks?
- In terms of MSP management, can we make these sensitive parameters under user management by employing biometric cryptosystem techniques to solve the central point of attacks in addition to the IBC-based key-escrow problems?

1.3 Thesis Motivation

Users, typically, go on to adopt a cloud-based application to serve their digital technologies because of the positive features comparing to those provided by traditional

IT. The typical (i.e., unencrypted form) storage of data in cloud-based data storage makes it easily read, update or tamper with by dishonest service providers or by external attackers who can penetrate the server—thus a security challenge could rise for that reason. Besides, a privacy risk could be carried out if the stored data are sensitive personal information. As stated in section 1.1, various literature studies recommended storing data in an encrypted form, requiring the use of traditional PKEs. There are, however, also some challenges highlighted earlier and will be discussed in detail in the upcoming chapters (3, 4 and 5) related to the adoption of traditional PKEs. Alternative promising public-key encryption schemes IBCs and F-IBCs were introduced [17], [18], [21] to address most of PKEs' problems. However, the control on stored data has been transferred from CSPs in cloud computing to a trusted party (a member of IBCs and F-IBCs) called PKG. Furthermore, the primary component of releasing the decryption key is also under the PKG's management.

Biometrics or biometric recognition systems simply refer to the process of recognizing individual automatically by adopting either their physiological or behavioural attributes[22], [23]. Instead of using something that the individual may possess, e.g., ID card, or may need to remember such as a password, the biometric systems are used to generate individuals' unique identities—where it unforgettable as the password and cannot be lost or stolen like the ID card. In IBCs, the biometrics is introduced to produce another version of IBCs that addressed the user verification requirements. F-IBC rely on users' public biometrics data to generate their keys pairs (i.e., encryption and decryption keys)[17].

To sum up, the motivation for this thesis involves improving the security of F-IBCs to be used in protecting data stored on cloud computing and preserving users' privacy.

1.4 Research Aim and Objectives

The thesis aims to improve the overall security and privacy of cloud computing by developing practical techniques that can offer users a reasonable control over the security of their data based on integrating the users' biometric data with IBC scheme.

The following points summarised the objectives of this thesis:

- Investigating and evaluating recent studies on security and privacy concerns that may arise as a result of reliance on cloud computing to find out what contemporary solutions have been followed to overcome or alleviate the concerns of cloud users.
- Conducting a study on modern encryption systems and evaluating their advantages over tradition encryption systems regarding improving the security and privacy of cloud computing - are they sufficient to support the security and privacy of cloud users or not?
- Implementing an F-IBC and evaluating the feasibility and the challenges related to applying such kind of encryption systems in practical real-life scenarios.
- Cryptanalyzing the existing IBC systems and focusing on the fuzzy identity-based cryptosystems (F-IBCs) to find out how one can exploit the user's biometric data and IBC to achieve the aim of the thesis.
- Bringing the control of cloud-based data storage applications back to the cloud user rather than the PKG based by utilising user's biometric.

1.5 The Main Contributions of the Thesis

It is important to emphasise that most of our proposed solutions are centre around giving more control to data owners in F-IBCs and, on the other hand, reducing the dominance of the PKG, which traditionally has full control. The contributions of the thesis can be summarised as follows.

- Improving the security of existing F-IBCs by proposing counter-measures against attacks by impersonators in the decryption-key releasing phase based on a challenge-response authentication mechanism to solve users' verification challenge.

- Proposing solutions that offer data owners more control over their data stored on a remote site (e.g., cloud computing).
  - Making the process of generating decryption keys, in F-IBCs, a cooperative process between the data owners and PKG. Hence, the decryption keys will consist of two parts, one issued by the data owners and the other by PKG.
  - Imposing a new key-validity for the existing F-IBCs using Shamir Secret Sharing method. Our contribution relies on a polynomial equation of first degree as well as the Lagrange Interpolation equation to give the data owner control over the validity of the keys
  - Proposing a modified version of F-IBCs in which the key ingredient of the decryption key is controlled by data owners and bound with their biometrics using biometric cryptosystem techniques.
- Developing an effective access control system through which data owners can decide who can access and decrypt their cloud-based data.
- Protecting users' privacy by adopting cancellable biometrics in F-IBCs instead of using the raw version of users' biometric data.

1.6 PUBLICATIONS

1. **W. K. Hassan** and H. Al-Assam, "Key exchange using biometric identity-based encryption for sharing encrypted data in the cloud environment," Mob. Multimedia/Image Process. Secure. Appl. 2017, vol. 10221, no. May, p. 102210J, 2017.
2. H. Al-Assam, **W. Hassan**, and S. Zeadally, "Automated Biometric Authentication with Cloud Computing," Biometric-Based Phys. Cybersecurity Syst., pp. 455–475, 2019.

1.7 THESIS OUTLINE

In general, the thesis consists of 8 chapters, which are as follows:

- Chapter Two aims to give the reader the relevant background and fundamental concepts in Cloud Computing and Cryptography (mathematical concepts).
- Chapter Three presents an introduction and review of Attribute-based Access Control (ABAC) generations and identity-based cryptosystems (IBCs).
- Chapter Four presents the first proposed solution called "One-Time Challenge-Response Multifactor Authentication for fuzzy identity-based cryptosystems. It aims to solve the problem of a user's verification that accompanying the adoption of F-IBCs in which precedes the delivery of decryption keys.
- Chapter Five presents the second proposed solution with regard to the key management and a key revocability in F-IBCs. This solution supports the data owners by making them directly participate in the process of generating the decryption keys. Also, we employee Shamir Secret Sharing to enforcing keys revocability.
- Chapter Six discusses the third proposed solution that aims to shift the control over the encrypted data to the data owners rather than PKG. It binds the core elements of issuing the decryption keys in F-IBCs (called Master Secret Parameter (MSP)) to the user's biometrics.
- Chapter Seven deals with the last contribution of this thesis. It uses F-IBCs as a key exchange between two parties to securely exchange a symmetric key used to encrypt data and stored on Cloud computing.
- Chapter Eight concludes the research carried out in the thesis and suggested future works will be in chapter eight.

CHAPTER 2

BACKGROUND AND FUNDAMENTAL CONCEPTS IN CLOUD COMPUTING AND CRYPTOGRAPHY

This chapter aims to give the reader the relevant background and fundamental concepts in Cloud Computing and Cryptography. First, a discussion on Cloud Computing, its definition and proliferation over the last few years is presented in Section 2.1. Four fundamental aspects of Cloud Computing are presented in Sections2.1.1-2.1.4. The main characteristics such as; on-demand service, elasticity are presented in Section 4). The delivery models of Cloud Computing such as; Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are presented in Section 2.1.2. The basic deployment models are presented in Section 2.1.3 and Section 2.1.4 presents security challenges faced by Cloud Computing.

Second, a detailed description of the fundamental Cryptographic (mathematical) concepts relevant to identity-based cryptosystems is presented in Section 2.2. Groups, Elliptic Curve (EC) and Bilinear Mappings are discussed in Section Chapter 2, 2.2.4 and 2.2.5 respectively. These discussions are supplemented with examples for better understanding of the concepts. Finally, the chapter concludes with a summary in Section 2.3.

2.1 Cloud Computing

Over the last few years, cloud computing has become one of the fastest-growing IT fields due to the services it provides to individuals and businesses. Cloud computing, as defined by the National Institute of Standards and Technology (NIST), is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [24]. Cloud Service Providers (CSPs) are the key players in cloud computing, who are responsible for providing clients with a wide range of services. These services vary from software levels such as Microsoft Office 365 and Google Docs to complete infrastructure level such as Amazon Elastic Compute Cloud (EC2) [25].

For appropriate understanding, four aspects of cloud computing are presented:

1) Characteristics of cloud computing,
2) Delivery models,
3) Deployment models,
4) Security challenges in cloud computing.

2.1.1 Characteristics of Cloud Computing

The convenience provided by cloud computing has led to an increasing trend of many business organisations, government agencies and customers to migrate their services and data into cloud environments. This trend can be attributed to the following five characteristics [25]:

- *On-demand self-service*: A client can immediately get computing resources (e.g., CPU time, applications, network storage, etc.) without a need for human intervention at the CSP side.
- *Broad network access*: Cloud resources are network accessible from different clients' applications installed on various platforms such as smartphones, tablets, PCs, and laptops.
- *Resource pooling*: The CSPs aggregate their resources to meet clients' need by appropriating multi-tenant approaches based on physical as well as virtual resources which can be dynamically added or withdrawn based on clients' requirements. The

pooling factor means that the clients do not need to know where the resources are coming from or where the data is physically stored.

- *Rapid elasticity*: The capabilities of cloud services should be flexible enough to rapidly shrink or expand to meet the requirements of different clients at different times.
- *Measured* service: CSPs have the ability to measure any resources used by each tenant (client) using charge-per-use mechanisms.

2.1.2 DELIVERY MODELS

Cloud services are typically delivered to clients using pre-packaged combinations of IT resources provided by CSPs. They are delivered on one of the following three cloud service models [26].

Software as a Service (SaaS): This model of delivery is also called "on-demand software". The software and associated data are centrally hosted on CSP's servers (i.e. instead of using the Clients' machine) where no maintenance or upgrades are required. In this model, clients have no control or management permission over the underlying cloud infrastructure. Typical examples of SaaS include Google Docs, Dropbox, and Microsoft Office 365.

Platform as a Service (PaaS): This model of service is typically used by application developers. It provides access to computing platforms that include operating systems, programming languages, software tools, databases, web servers, etc. In this model, the clients have control only over the deployed applications. Some examples of PaaS include Google AppEngine, Microsoft Azure, and Apache Stratos.

Infrastructure as a Service (IaaS): This delivery model supplies clients with computing resources (physical or more often virtual) processors, storage, firewalls, load balancers, virtual local area networks. Therefore, the clients are not only able to deploy and execute various software, but also have control over the operating systems, storage, processing power, and networking components. Amazon's EC2 is an example of IaaS.

2.1.3 Deployment models

The deployment models described in section 2.1.2 can be deployed in different environments. Deployment models define ownership and the size of cloud resources, and most importantly restrict who can access them. Currently, four basic models of deployment have been identified [26].

- **Private Cloud Computing**: The cloud infrastructure and services are offered exclusively to one enterprise, and it might be owned, managed as well as operated by the enterprise, a third party or a combination of both. This deployment model not only gets the optimal use of existing in-house resources, but it also provides better data security and privacy. It should be noted that the cloud environment in this model might be located in or outside of the premises of the enterprise.
- **Community cloud computing**: The cloud infrastructure is shared by a group of clients or organisations to provide shared policies, values, and security procedures. The ownership, management, and operation of this model are given to one or more members of the group.
- **Public Cloud Computing**: The cloud infrastructure is open for public use. The ownership and management are given to business, academic institutes, government bodies, and so on.
- **Hybrid Cloud Computing**: More than one deployment models can be combined to form a hybrid cloud environment to meet clients' needs.

It can be argued that each type of service and deployment model meets the demands of some business more than others. For example, while a large enterprise might benefit from the private cloud, smaller corporations will most likely opt for a public cloud for cost consideration. Figure 2-1 illustrates a typical cloud computing service layers along with their cost and timeline impact.

2.1.4 Security Challenges in Cloud Computing

Although cloud computing offers considerable advantages over other traditional IT solutions, it poses serious security concerns. In fact, security and privacy are essential factors for an enterprise when deciding on whether to migrate their data, applications, and other relevant services to cloud environments. Typically, Service Level Agreements (SLAs) between clients and CSPs tend to include details on how to access and utilise

cloud services, service duration, and data storage and management when the contract ends [24]. However, the main challenge is how to guarantee that the data is accessible by authorised users only.

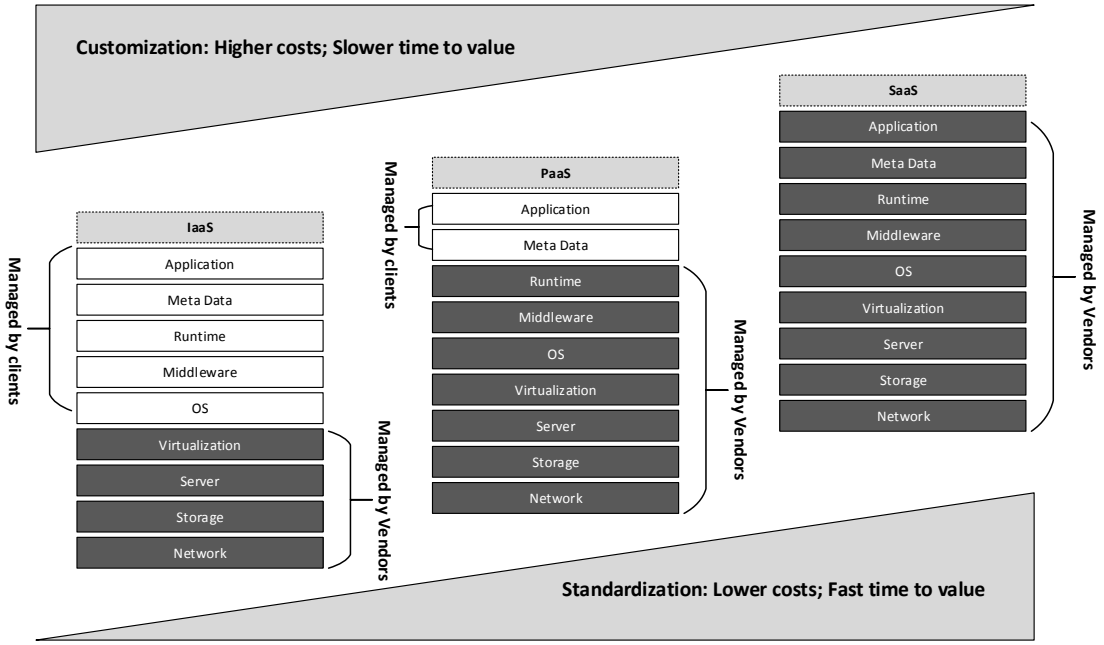


Figure 2.1: Typical Layers of Cloud Computing Services (adapted from [25])

When data owners decide to use the cloud environment, they rely entirely on third parties to make decisions about their data. Therefore, it is imperative for data owners to have the right technologies or methods to prevent CSPs from utilising such data without their permission. Both technical and non-technical methods have to provide effective means to fulfil this goal [27][28]. A wide range of possible solutions has been proposed to implement different mechanisms to prevent unauthorised access to cloud data even by untrusted CSPs [18], [29]–[33]. In general, to address clients' concerns about security and privacy of the cloud environment, the following three essential challenges must be addressed [27]:

- **Outsourcing**: In the traditional IT environment, clients can exercise full control over their data. However, they usually lose all means of physical control over the data once it is migrated to cloud environments, which is a key security concern. To overcome this problem, clients need to ensure that the cloud services providers are trustworthy and are capable of meeting the requirements related to secure data storage, correctness and integrity of cloud data and computation at all times and maintaining clients' privacy.

- **Multi-tenancy**: Cloud environments can share their resources and services among multiple clients simultaneously. Both the virtual machines provided by CSPs and the cloud data of different clients are eventually located on a single physical machine based on a particular resource allocation policy. Hence, a legitimate cloud client can potentially act as an adversary by exploiting some holes in the policies to gain unauthorised access to the data of other users.
- **Big data and intensive computation**: Cloud environment requires dealing with large volumes of data supported by powerful processing capabilities. Hence, traditional security techniques might be difficult to apply to such data because of the quantity of high computation and communication overheads. For instance, to guarantee the integrity of remotely stored data, it is computationally infeasible to hash the whole data. Consequently, new strategies and protocols are needed to overcome such difficulties.

2.2 Mathematical Background

Initially, to make encryption systems presented in the following chapter intelligible, it is imperative to clarify the mathematical concepts and notations that have been adopted. For this, this section has been drawn up to list the most important definitions and theories that are relevant to identity-based cryptosystems (IBCs). In addition, some examples are prepared.

2.2.1 Group

In general, a set of elements with the binary operation is said to be a group once a certain of conditions are met. The conditions are— closure, associativity, identity, inverse as well as commutativity [34][35]. The binary operation could be carried out between any two elements to produce another element. Mathematically, the group is defined as:

Definition 2.1: Group *- A Group is a set of elements* $\mathbb{G}$ *and a binary operation (∘) takes place on two elements in* $\mathbb{G}$*. To say this set is a group, the following should be achieved.*

a) **Closure** - The group operation (∘) should be *closed*, i.e., for all $\alpha, \beta \in \mathbb{G}$, the result of $\alpha \circ \beta$ is also in $\mathbb{G}$.

b) **Associativity** - The group operation ∘ is associative, i.e., for all $\alpha, \beta, \gamma \in \mathbb{G}$, $\alpha \circ (\beta \circ \gamma) = (\alpha \circ \beta) \circ \gamma$.

c) **Identity** - The set has to have a neutral (or identity) element *e* such that for all $\alpha \in \mathbb{G}$, $\alpha \circ e = e \circ \alpha = \alpha$.

d) **Inverse** - Each element $\alpha \in \mathbb{G}$ has to have an inverse element $\alpha^{-1} \in \mathbb{G}$, that is $\alpha \circ \alpha^{-1} = e$.

e) **Commutativity** - A group $\mathbb{G}$ has to be an abelian (or commutative) group. This is achieved if for all $\alpha, \beta \in \mathbb{G}$, $\alpha \circ \beta = \beta \circ \alpha$.

Furthermore, multiplicative and additive are the principal operations that can be applied to groups. Subtraction and division operations are turned to addition and multiplication operations, respectively (e.g., $a - b = a + (-b)$, and $a/b = \alpha * b^{-1}$, where $b \neq 0$).

It should be emphasized that the group $\mathbb{G}$ without a *finite number* of elements $\mathbb{Z}_k^*$ does not support the encryption systems.

Theorem 2.1. *The set $\mathbb{Z}_k^*$ which comprises of all integers $\{0, \dots, k-1\}$ and has gcd (i, k) =1 constructs an abelian group under multiplication modulo k with the neutral element is e=1.*

The following example demonstrates the group conditions presented above.

Example 2.1. Let k=7, $\mathbb{Z}_7^*$ consists of the elements {1, 2, 3, 4, 5, 6}. Table 2-1 shows the multiplicative table of $\mathbb{Z}_7^*$:

Table 2-1: The results of the multiplicative operation in K=7

X *mod 7*	**1**	**2**	**3**	**4**	**5**	**6**
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

To guarantee $\mathbb{Z}_7^*$ is a group, the above group conditions (see definition 2. 1) must be met. Table 2-1 exhibits that the first condition (i.e., closure) has been accomplished since it has been getting all the elements (numbers) of $\mathbb{Z}_7^*$. Besides, both third and fourth conditions are also satisfied since all elements of $\mathbb{Z}_7^*$ appearing randomly distributed in both rows and columns. Because of each element at row *i* and column *j* (element (*i*, *j*)) is equal to the element at row *j* and column *i* (element (*j*, *i*)), this implies that the fifth condition is also met.

Notwithstanding, the *Extended Euclidean* algorithm can be employed to calculate a^{-1} of an element$a \in \mathbb{Z}_k^*$. To check the validity of the second condition, let us check any three elements $\in \mathbb{Z}_7^*$, for instance, 3, 5 and 6 then:

$$(3 * 5) * 6\ (mod 7) \equiv 15 * 6 (mod 7) \equiv 1 * 6 (mod 7) \equiv 6 (mod 7)$$

$$3 * (5 * 6)\ (mod 7) \equiv 3 * 30 (mod 7) \equiv 3 * 2 (mod 7) \equiv 6 (mod 7)$$

2.2.2 Cycle Group

Abstractly, finite constructions must be present to carry out the encryption/ decryption process. Therefore, it is important to highlight these concepts which are a key element in building cryptography systems.

Definition 2.2. Order of an Element *- A smallest positive integer n of an element λ in a group ($\mathbb{G}$,$\circ$) such that $\lambda = \underbrace{\lambda \circ \lambda \circ \ldots . \lambda}_{n\ times} = 1$ where the number 1 refers to the neutral element of the group $\mathbb{G}$.*

For further clarification, see example below.

Example 2.2. *Assume the following group $\mathbb{Z}_{11}^{*}$, what is the order of the element a=3. Because the operation is a multiplicative group, It should continue to calculate the exponents of a till the result be equal to 1(i.e., equal to the neutral element).*

$a^1 \equiv 3^1 \pmod{11} \equiv 3 \pmod{11}$
$a^2 \equiv 3^2 \pmod{11} \equiv 9 \pmod{11}$
$a^3 \equiv 3^3 \pmod{11} \equiv 27 \pmod{11} \equiv 5 \pmod{11}$
$a^4 \equiv a^3 . 3^1 \pmod{11} \equiv 5 . 3 \pmod{11} \equiv 15 \pmod{11} \equiv 4 \pmod{11}$
$a^5 \equiv a^4 . 3^1 \pmod{11} \equiv 4. 3 \pmod{11} \equiv 12 \pmod{11} \equiv 1 \pmod{11}$

As it is shown, the neutral element produced at the exponent $a=5$. For this, the order of α =3 is 5. If it requires to calculate the rest of the exponents, the process must continue.

That is,

$a^6 \equiv a^5 .3^1 \pmod{11} \equiv 1. 3 \pmod{11} \equiv 3 \pmod{11}$
$a^7 \equiv a^6 .3^1 \pmod{11} \equiv 3. 3 \pmod{11} \equiv 9 \pmod{11}$
$a^8 \equiv a^7 .3^1 \pmod{11} \equiv 9. 3 \pmod{11} \equiv 27 \pmod{11} \equiv 5 \pmod{11}$
$a^9 \equiv a^8 .3^1 \pmod{11} \equiv 5. 3 \pmod{11} \equiv 15 \pmod{11} \equiv 4 \pmod{11}$
$a^{10} \equiv a^9 .3^1 \pmod{11} \equiv 4. 3 \pmod{11} \equiv 12 \pmod{11} \equiv 1 \pmod{11}$

The results are then repeated (i.e., cyclic every five times) and the order of an element $a=3$ is five which involves {1, 3, 4, 5, 9}.

Definition 2.3. Cyclic Group *- A given group $\mathbb{G}$ that has an element α is usually indicated as a Cyclic Group if the maximum order of the element α is equal to the order of $\mathbb{G}$.*

Definition 2.4. Group Generator *- All elements which have maximum order are termed as generators or primitive elements.*

Let us go back again to the example 2.2 and check whether the element $\alpha = 2$ is a primitive element (or generator) of $\mathbb{Z}_{11}^*$. Since the elements of the group $\mathbb{Z}_{11}^*$ are {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, then the cardinality of $|\mathbb{Z}_{11}^*| = 10$. The elements are generated by the power of element $\alpha = 2$ can be shown in the following table:

Table 2-2: Check the generator group of $\mathbb{Z}_{11}^*$

i	a^i	$a^i (mod\ 11)$
1	2	2
2	4	4
3	8	8
4	16	5
5	32	10
6	64	9
7	128	7
8	256	3
9	512	6
10	1024	1

By looking to the right column (α^i (mod 11))— we can see all the elements of $\mathbb{Z}_{11}^*$ are generated so that the order (a) = $|\mathbb{Z}_{11}^*|$. As a consequence, a is a primitive element (or a generator of $\mathbb{Z}_{11}^*$) and the group $\mathbb{Z}_{11}^*$ is a cyclic group. Constructing cryptosystems rely primarily on a relation between the group elements and the exponents.

A set of characteristics concerning the cyclic groups are essential to encryption systems. The following two theorems point out these characteristics.

Theorem 2.2*. For each prime number p, the group ($\mathbb{Z}_p^*$,$\circ$) be an abelian finite cyclic group.*

Hence, these groups are the building blocks of forming Discrete Logarithm systems as can be shown in section 2.2.3.

Theorem 2.3. Cyclic subgroup *- For a given cyclic group ($\mathbb{G}$,$\circ$) each element $a \in \mathbb{G}$ which has an order$(\alpha) = \beta$ deems to be a generator of a cyclic subgroup of β.*

To revert to $\mathbb{Z}_{11}^*$, $\alpha = 3$ is a generator of the cyclic group $\beta = \{1,3,4,5,9\}$—thus, β is a cyclic subgroup of the group $\mathbb{Z}_{11}^*$.

2.2.3 Discrete Logarithm Problem

The security of numerous encryption systems' constructions relies on Discrete Logarithm hardness (or problem) (DLP). The cyclic group in prime fields $\mathbb{Z}_p^*$ is an example of binding the DLP to the encryption systems. Not that the DLP has also been strongly adopted in *hardness assumptions* that measure how systems are secured against certain attacks.

Definition 2.5. Discrete Logarithm Problem in $\mathbb{Z}_p^*$*- For a given generator element $\alpha \in \mathbb{Z}_p^*$ of a finite cyclic group $\mathbb{Z}_p^*$ and element $\beta \in \mathbb{Z}_p^*$, the DLP is the problem of calculating an integer n such that $\alpha^n \equiv \beta \pmod p$.*

The integer *n* refers to the *discrete logarithm of* β to the base α, and it can be written as follows: $n = log_\beta \pmod p$.

2.2.4 Elliptic Curves

An elliptical curve (EC) depicts a special polynomial equation which principally bases on generalising the DLP. It is extremely important to work alongside with cyclic group environment that assumes the hardness of calculating the DLP over this group. For this, it adds a good one-way characteristic, says Paar and Pelzl in [34]. Moreover, it remains to know that to perform the cryptographic purpose; the curve should be executed over prime fields or Galois Fields of prime order $GF(P)$. So, modulo *P* does all operations.

Definition 2.6. Elliptic curve over $\mathbb{Z}_p$ *- An elliptic curve over $\mathbb{Z}_p$ is a set of all pairs (x, y) $\in \mathbb{Z}_p$ that achieves the following equation: $y^2 \equiv x^3+ax+b \pmod p$, where a, b $\in \mathbb{Z}_p$, p>3 with an obligatory condition that $4a^3 + 27b^2 \neq 0 \pmod p$.*

For example, the following elliptic curve $y2 \equiv x^3 - 3x + 3$ over real number $\mathbb{R}$ can be observed in Figure 2.2.

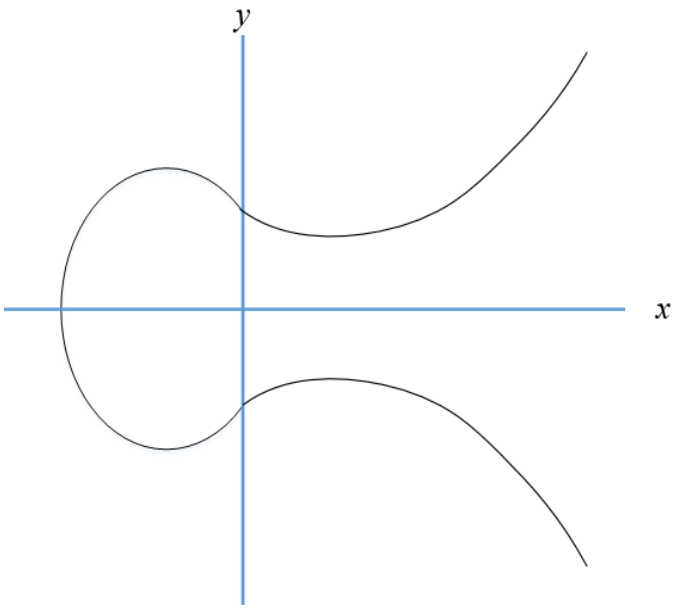


Figure 2.2: Elliptic curve over real number R

2.2.4.1 Executing Group Operation over EC

The addition operation (+) is a binary operation that is widely being carried out in the group. It is worth noting that all the group operations, identified in section 2.2.1, can also be realized with the Elliptic Curve (EC), and therefore, the EC can serve the encryption/ decryption operations[34]. For this reason, it is necessary to give the reader further clarification on its fundamental operations. However, EC adopts two basic operations—*point addition* and *point doubling.*

Assume the following two points P and Q with their coordinates (x_1, y_1) and (x_2, y_2), respectively.

- **Point Addition P+Q:** The following formula is applied to carry out this operation.

$$EC = P + Q, where\ \mathrm{P} \neq Q. \qquad (2.1)$$

Thus, the procedure, depending on their coordinates, is as follows.

$$P + Q = (x_1, y_1) + (x_2, y_2) = (x_r, y_r) \qquad (2.2)$$

The point addition on an elliptic curve over real numbers $\mathbb{R}$ can be fulfilled by connecting P and Q via a straight line which, in turn, intersects EC in another

point. From this new point, drop a vertical line then after, drop from this point a vertical in parallel to the y-axis and intersects with the *x-axis*. The corresponding intersection point of this line with the elliptic curve represents the new point generated from the point addition operation (see Figure 2.3).

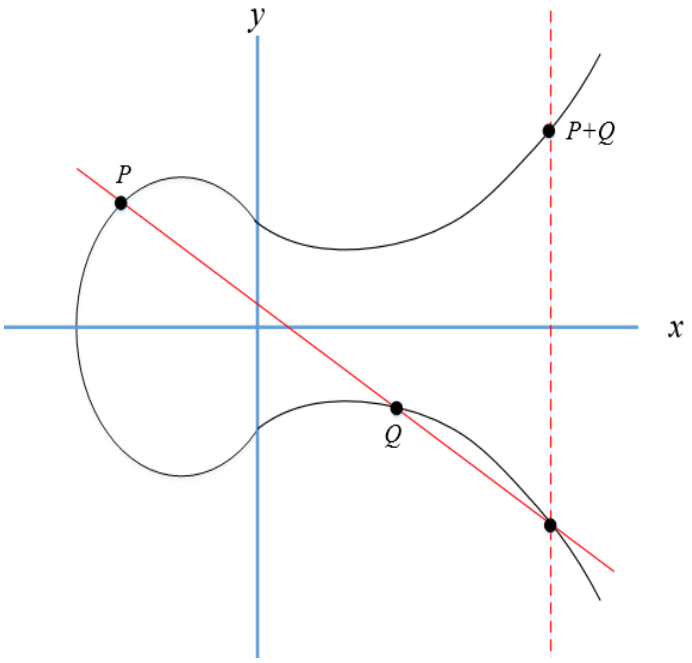


Figure 2.3: Point addition on an elliptic curve over R

- **Point Doubling P+P:** It can be calculated based on the following statement:

$$EC = P + P = 2P$$
$$or$$
$$EC = Q + Q = 2Q, \text{ where } P = Q. \qquad (2.3)$$

The point doubling ordinarily constitutes by drawing a tangent line through Q to get the second point which intersects the EC. Figure 2.4 depicts the point doubling has been calculated on EC based on real numbers $\mathbb{R}$. Not that, EC can use different fields in addition to the real numbers $\mathbb{R}$. Therefore, in order to serve the cryptography requirements, EC will be examined in a prime field.

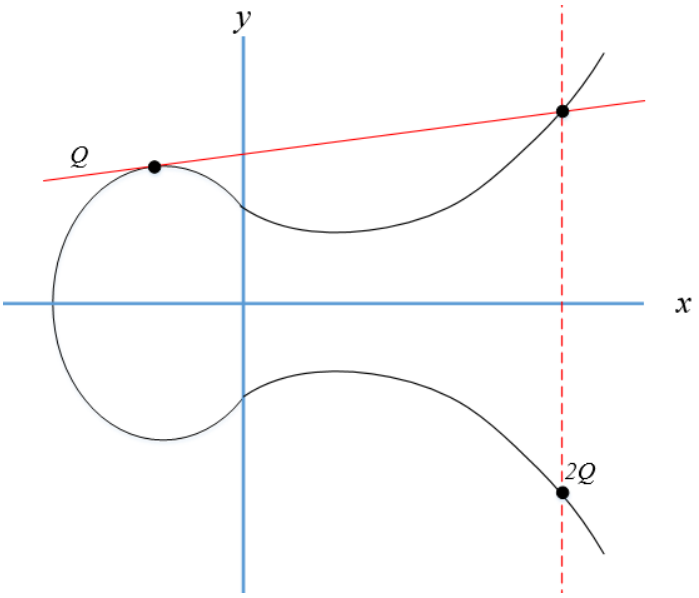


Figure 2.4: Point doubling on an elliptic curve over R

Nonetheless, the following expressions describe the process of calculating the group operations (i.e., point addition and doubling):

$$\begin{aligned} x_r &= \lambda^2 - x_1 - x_2 \quad (mod\ p) \\ y_r &= \lambda(x_1 - x_r) - y_1 \quad (mod\ p) \end{aligned} \tag{2.4}$$

Where we can calculate the slope λ of a line using the following formula.

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \text{ mod} p, & if\ P \neq Q \ldots\ldots\ldots (point\ adition) \\ \dfrac{3x_1^2 + a}{2y_1} \text{ mod} p, & if\ P = Q \ldots\ldots\ldots (point\ doubling) \end{cases} \tag{2.5}$$

Furthermore, to establish the finite group, we should examine whether the conditions described in the group definition (see section 2.1) can be achieved on the elliptic curve. Another key thing to remember that, there is no point can accomplish the neutral element condition, it usually defines a point at infinity (∞) to represent it, i.e., $P + \infty = P$.

Indeed, this point can be seen at infinity either (+ ∞) or (- ∞) trends the y-axis (see figure 2-5). Additionally, the inverse of point *P* is *-P*, this implies if $P = (x, y)$ then $-P = (x, -y)$ such that $P + (-P) = \infty$. Next, to compute *-P*, the *tangent* and *chord* method is utilized.

Accordingly, when an elliptic curve over a prime field *GF (p)* is required, *-P* can be calculated immediately using the following formula:

$$-P \equiv (c, -y) \pmod{p}, \text{ where } -y \equiv P - y \pmod{p} \qquad (2.6)$$

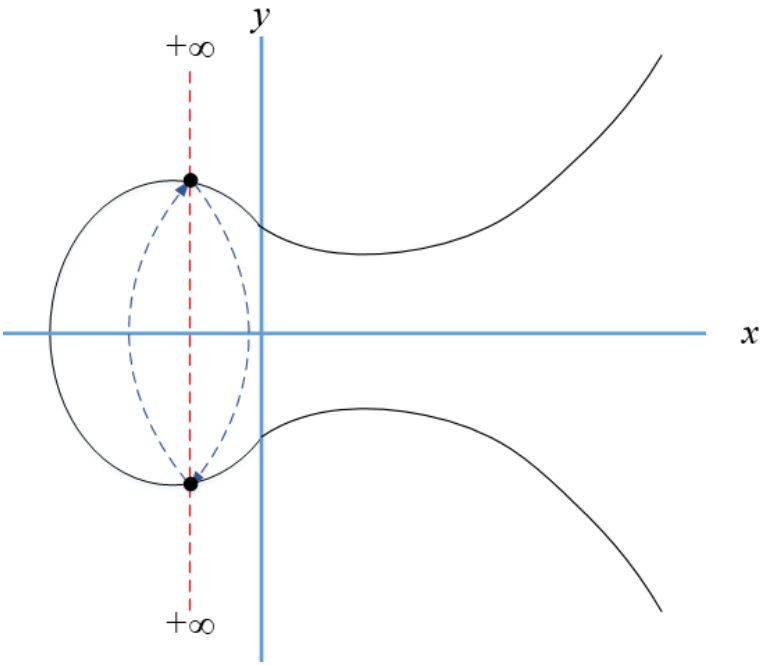


Figure 2.5: How to compute a neutral element over the EC

Demonstrating this can be deeply understood by the following example.

Example 2.3. *Consider the following curve equation over the field* $\mathbb{Z}_{29}$

$$EC: y^2 = x^3 + 3x + 7 \bmod 29$$

Assume we have the following points (4, 5), (10, 14). As we can see, there are two different points, thus, the operation will inevitably be a point addition:

$$(4, 5) + (10, 14) = (x_r, y_r)$$

The slope λ can be found as follows:

$$\lambda = ((y_2 - y_1) / (x_2 - x_1)) \pmod{29} = ((14-5)/ (10-4)) \pmod{29}$$

$$= 9/6 \pmod{29}$$

$$= 9*6^{-1} \pmod{29}$$

$$= 9*5 \pmod{29}$$

$$= 16 \pmod{29}$$

Next step, find the coordinates (x_r, y_r) of a new point:

$$x_r = \lambda^2 - x_1 - x_2 \; mod 29 = (16)^2 - 4 - 10 \; (mod\ 29) = 10 \; (mod\ 29)$$

$$y_r = \lambda(x_1 - x_r) - y_1 mod 29 = 16\ (4 - 10) - 5 \; (mod\ 29) = 15 \; (mod\ 29)$$

Thus, the new point = (x_r, y_r) = (10, 15).

At present, it is necessary to check whether the new point's coordinates are really over the EC by applying the new point over the curve equation:

$$EC: y^2 = x^3 + 3x + 7 \; mod 29$$

$$(15)^2 = (10)^3 - 3*10 + 7 \; (mod\ 29)$$

$$255 = 1037 \; (mod\ 29)$$

$$22 = 22 \; (mod\ 29), \text{ where } 22 \in \mathbb{Z}_{29}$$

Example 2.4. *Consider the following single point (6, 3), and the following elliptic curve over the field* $\mathbb{Z}_{37}$*:*

$$EC: y^2 = x^3 - 5x + 8$$

For the sake of calculating the doubling of this point, initially, we must compute the slope λ as follows:

$$\lambda = (3x_1^2 + a) / (2y_1) \; (mod\ 37)$$

$$= (3 * 6^2 - 5) / 2 * 3 \; (mod\ 37)$$

$$= 103 / 6 \; (mod\ 37)$$

$$= 103 * 6^{-1} \; (mod\ 37) = 103 * 31 \; (mod\ 37) = 11 \; (mod\ 37)$$

To find the coordinates of the new point (x_r, y_r), the same procedure has been used for point addition will be also used as follows:

$$x_r = \lambda^2 - x_1 - x_2 \; mod\ 37 = (11)^2 - 6 - 6 \; (mod\ 37) = 35 \; (mod\ 37)$$

$$y_r = \lambda(x_1 - x_r) - y_1 \; mod\ 37 = 11\ (6 - 35) - 3 \; (mod\ 37) = -322 \; (mod\ 29) = 11 \; (mod\ 29).$$

Hence, the new point = (x_r, y_r) = (35, 11).

2.2.4.2 Discrete Logarithm Problem in Elliptic Curves

This section aims to show the relationship between DLP and EC, and how it works concerning encryption systems. Giving some examples gives flexibility in understanding this relationship and thus how it serves the encryption/ decryption processes.

Theorem 2.4. *Points over elliptic curves alongside a neutral element have to be a cyclic subgroup, and thus, all points over elliptic curves can shape a cyclic group under particular properties.*

The following example is established to clarify theorem 2.4 and how to have a cyclic group characteristic in an elliptic curve.

Example 2.5. *Consider the example 2.4. It requires to find all points that exist on the intended curve.*

The slope and the coordinates of the new points will be calculated using the same procedure used with the previous example. As a result, the new point is Q + Q = (6, 3) + (6, 3) = (35, 11) = 2Q. Table 2-3 demonstrates the points that have been generated and are located on *EC:* $y^2=x^3 - 5x + 8$ depending on the point Q = (6, 3).

Table 2-3: The points (elements) have resulted over the EC

Q + Q	2Q	(6,3)	(6,3)	(35,11)	19Q + Q	5Q	(8,6)	(6,3)	(16,19)
2Q + Q	3Q	(35,11)	(6,3)	(34,25)	20Q + Q	6Q	(16,19)	(6,3)	(22,1)
3Q + Q	4Q	(34,25)	(6,3)	(8,6)	21Q + Q	7Q	(22,1)	(6,3)	(20,8)
4Q + Q	5Q	(8,6)	(6,3)	(16,19)	22Q + Q	8Q	(20,8)	(6,3)	(20,29)
5Q + Q	6Q	(16,19)	(6,3)	(22,1)	23Q + Q	9Q	(20,29)	(6,3)	(22,36)
6Q + Q	7Q	(22,1)	(6,3)	(20,8)	24Q + Q	10Q	(22,36)	(6,3)	(16,18)
7Q + Q	8Q	(20,8)	(6,3)	(20,29)	25Q + Q	11Q	(16,18)	(6,3)	(8,31)
8Q + Q	9Q	(20,29)	(6,3)	(22,36)	26Q + Q	12Q	(8,31)	(6,3)	(34,12)
9Q + Q	10Q	(22,36)	(6,3)	(16,18)	27Q + Q	13Q	(34,12)	(6,3)	(35,26)
10Q + Q	11Q	(16,18)	(6,3)	(8,31)	28Q + Q	14Q	(35,26)	(6,3)	(6,34)
11Q + Q	12Q	(8,31)	(6,3)	(34,12)	29Q + Q	15Q	(6,34)	(6,3)	∞
12Q + Q	13Q	(34,12)	(6,3)	(35,26)	30Q + Q	Q	∞	(6,3)	(6,3)
13Q + Q	14Q	(35,26)	(6,3)	(6,34)	31Q + Q	2Q	(6,3)	(6,3)	(35,11)
14Q + Q	15Q	(6,34)	(6,3)	∞	32Q + Q	3Q	(35,11)	(6,3)	(34,25)
15Q + Q	Q	∞	(6,3)	(6,3)	33Q + Q	4Q	(34,25)	(6,3)	(8,6)
16Q + Q	2Q	(6,3)	(6,3)	(35,11)	.	.	.	.	.
17Q + Q	3Q	(35,11)	(6,3)	(34,25)	.	.	.	.	.
18Q + Q	4Q	(34,25)	(6,3)	(8,6)	.	.	.	.	.

Table 2-4 exhibits that for each point has been produced, there is an inverse point; for instance, the inverse of point (6, 3) is (6, 34), and vice versa. The following table gives each point, and the corresponding inverse point over the curve*: y2 = x3 -5x + 8*.

Table 2-4: The elements and their corresponding inverse upon the curve: y2 = x3 -5x + 8

(x, y)	(x, -y)
(6,3)	(6,34)
(35,11)	(35,26)
(34,25)	(34,12)
(8,6)	(8,31)
(16,19)	(16,18)
(22,1)	(22,36)
(20,8)	(20,29)
(20,29)	(20,8)
(22,36)	(22,1)
(16,18)	(16,19)
(8,31)	(8,6)
(34,12)	(34,25)
(35,26)	(35,11)
(6,34)	(6,3)

2.2.4.3 Generate Public Key Encryption in EC domain.

The section intends to clarify the way of generating the well-known encryption keys regarding the EC in cryptography. Typically, public key encryption or asymmetric keys have two critical keys called public and private keys.

Abstractly, let us go back again to $EC: y^2 = x^3 - 5x + 8$ over the group field $\mathbb{Z}_{37}$ and the point Q= (6, 3) of a prime order p.

The cyclic subgroup of $EC(\mathbb{Z}_{37})$ can be generated based on the element Q is:

$$\langle Q \rangle \underbrace{\{\infty, Q, 2Q, 3Q, 4Q, 5Q, 6Q, 7Q, 8Q, 9Q, 10Q, 11Q, 12Q, 13Q, 14Q, 15Q\}}_{times}$$

It should be noted that the prime p, the point Q, as well as the elliptic curve equation EC together, represent the public parameters. On the other hand the integer, i of (iQ) represents the private key. Hence, the private key ($sk = i$) will be selected uniformly at random from the interval [1 p-1], whereas the corresponding public key will be $pk = iQ$. For more explanation, pk is a point over an elliptic curve with coordinates $pk = (x_{pk}, y_{pk})$ while the private key is only an integer.

It is imperative to establish the *Discrete Logarithm* (*DL*) cryptosystems, know the order of the group. *Hasse's* theorem (or *Hasse bound*) was developed to provide an approximate number of points that exist on an elliptic curve on a finite group.

Theorem 2.4.3. *Hasse's bound: The number of points on a given elliptic curve EC over a finite group with p elements is restricted by* $p + 1 - 2\sqrt{p} \leq \#\mathcal{E} \leq p + 1 + 2\sqrt{p}$, *where* $\#\mathcal{E}$ *refers to the number of points on EC.*

The problem of determining a value of *i* from given public parameters and *pk* is the main idea of an elliptic curve discrete logarithm problem ECDLP. However, the ECDLP can be detected by the following definition:

Definition 2.7. Elliptic Curve Discrete Logarithm Problem (ECDLP) *- For a given an elliptic curve EC and two elements Q and T such that Q is the primitive element. The Discrete Logarithm Problem DLP is computed the integer i, where* $1 \leq i \leq \#\mathcal{E}$, *such that:*

$$\underbrace{Q + Q + Q + \cdots + Q}_{times} = iQ = T$$

2.2.4.4 Diffie-Hellman Key Exchange using Elliptic Curves

Diffie-Hellman Key Exchange DHKE can also be carried out using an elliptic curve. To understand the role of an elliptic curve in establishing DHKE,

Figure *2.6* explains the steps of DHKE between Alice and Bob using the elliptic curve.

Alice	***Bob***
• Choose $sk_A = \alpha \in \{2,3,\ldots, \#\varepsilon\}$	• Choose $sk_B = \beta \in \{2, 3,\ldots, \#\varepsilon\}$
• Calculate $pk_A = \alpha Q = A = (x_A, y_A)$	• Calculate $pk_B = \beta Q = B = (x_B, y_B)$
	$A \longrightarrow$
	$\longleftarrow B$
• $\alpha B = \alpha\, \beta Q = (x_{AB}, y_{AB})$	• $\beta A = \beta\alpha Q = (x_{AB}, y_{AB})$

Figure 2.6: Diffie-Hellman Key Exchange Protocol using EC

Firstly, both Alice and Bob have to select two long integers α and β as their secret keys. Depending on a random primitive element $Q = (x, y)$ was chosen as elliptic curve parameter, Alice and Bob then find their corresponding public keys pk_A and pk_B. Ultimately, Alice and Bob exchange their public keys with each other. By leverage from the associative property, they can get the same joint secret key $\alpha\beta Q$ which is, of course, an element over the elliptic curve. For this, the $\alpha\beta Q$ can be exploited to release a session key. Let us go back again to the previous example 2- 5 with simple integers: Assume the following elliptic curve $EC: y^2 \equiv x^3 - 5x + 8 (\mathrm{mod} 37)$ of order $\#\mathcal{E} = 15$ and a primitive point Q = (6, 3) (see

Figure *2.7*).

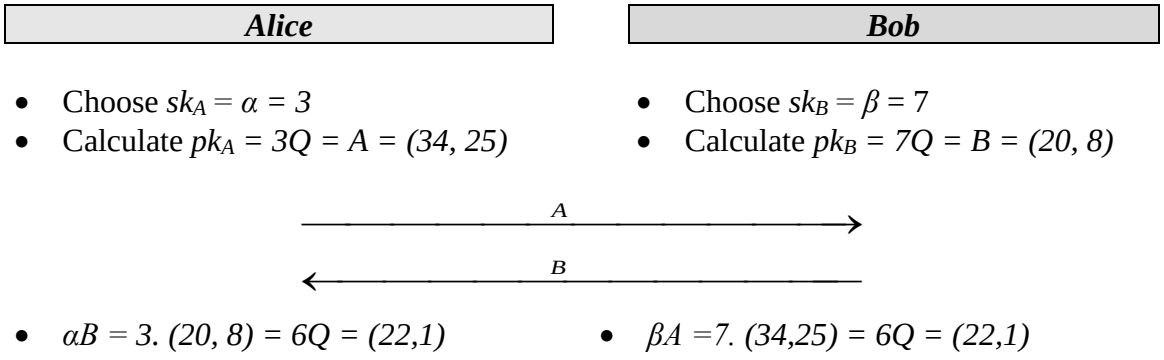


Figure 2.7: Diffie-Hellman Key Exchange protocol using EC:y^2≡x^3-5x+8 (mod37)

The shared secret key is the element (22, 1) on the elliptic curve that can be resulted using doubling and addition in both sides.

2.2.5 Bilinear Mapping

A. Menezes et al. pointed out that bilinear mapping was initially used in cryptography for the simulation of an attack algorithm[36] to break the elliptic curve. The idea of the algorithm mainly based on reducing the elliptic curve logarithm problem to a discrete logarithm problem in the multiplicative group over a finite field. Moreover, Koblitz and Miller referred to the way of constructing the public key using the group of points on an EC (further details about EC listed in section 2.2.4) that can be represented over the finite field [37], [38].

In the end, it is vital to highlight that the bilinear mapping had brought a significant turning point in the world of public key encryption systems when Boneh and Franklin exploited it in constructing the first practical model of identity-based

cryptosystems (IBCs) [18]; and, therefore, they solved an open challenge presented by Shamir[21].

Definition 2.8- *A given two distinct cyclic groups $\mathbb{G}_1$and $\mathbb{G}_2$ of a prime order p, and two distinct generators (primitives) elements $\langle g_1 \rangle$ and $\langle g_2 \rangle$ of $\mathbb{G}_1$. To form the bilinear map* $\hat{\boldsymbol{e}}$*:* $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, $\hat{\boldsymbol{e}}$ *must be an admissible map which is realized by relying on the following three conditions.*

- $\forall\, \alpha, \beta \in \mathbb{Z}_p$ *compute* $\hat{e}\left(g_1^{\alpha}, g_2^{\beta}\right) = \hat{e}(g_1, g_2)^{\alpha\beta} = \hat{e}\left(g_1^{\beta}, g_2^{\alpha}\right)$.
- $\hat{e}$ *should be no* $-$ $degenercy, i.e. \hat{e}(g_1, g_1) \neq \hat{e}(g_2, g_2) \neq \hat{e}(g_1, g_2) \neq 1$.
- $\hat{e}$ *is efficiently computable.*

Definition 2.9- *For a given finite cyclic group of order p,*$\langle g \rangle$ *is a primitive element of group* $\mathbb{G}_1$ *and* $\boldsymbol{h} \in \mathbb{G}_1$ *computes the unique* $\alpha \in \mathbb{Z}_p$ *such that* $\boldsymbol{h} = g^{\alpha}$ *and* $0 \leq \alpha \leq p\text{-}1$ *designates the discrete logarithm (DL) of* $\boldsymbol{h}$ *to base g and can be written as* $\alpha = Dlog_g\, \boldsymbol{h}$.

2.3 Chapter Summary

Cloud computing has seen significant growth over the last few years chiefly because it provides users with on-demand availability of computer system resources, in particular, data storage, networks, operating systems, and even virtual machines. This chapter has introduced general concepts and background knowledge in Cloud Computing and Cryptography. First, the characteristics, delivery and deployment models were introduced. Then, the security challenges faced in cloud computing such as; ownership, access control were presented. Second, fundamental mathematical concepts and theorems were presented which form the foundation of cryptography and identity-based cryptosystem (IBC) in particular.

The next chapter will focus on identity-based cryptosystems (IBCs), which is the main focus of the thesis and where the research contributes.

Chapter 3

Identity-Based Cryptosystems and Attribute-Based Access Control

In this chapter, we present an introduction and review of identity-based cryptosystems (IBCs) and Attribute-based Access Control (ABAC). In Chapter 1, we observed that IBC and ABAC are ideal candidates to protect users' data especially in the cloud at two levels – security and privacy. Chapter 1 also highlighted some of the vulnerabilities surrounding existing IBCs and ABACs. Therefore, this chapter begins by introducing access control systems and their models in Sections 3.2 and 3.3. Identity-based cryptography is presented in Section 3.4 in which we explain the standard and fuzzy identity-based cryptosystems, the security evaluation of fuzzy identity-based cryptosystems, the challenges of IBC infrastructure, and the Attribute-based cryptography (ABC). Finally, the chapter is summarised in Section 3.5.

3.1 INTRODUCTION

In Chapter 2, we explained the services related to cloud computing, such as data storage, network access, on-demand service and others, and highlighted that cloud-based data storage is one of the most important services that cloud computing provided to consumers. However, the rapid and extensive use of internet services such as cloud environment makes users' sensitive data protection paramount especially the security and user's privacy. Therefore, when the users decide to deal with the cloud service, it is imperative to implement proper mechanisms or techniques that preserve the security and privacy of their data.

As indicated in chapter 2, the cloud service providers (CSPs) are the only entities with absolute control over the capabilities (services) of cloud computing. In the case of Cloud-based data storage service, it implies transferring the control of the data or any other products from data owners (DOs) – users - to CSP—thus, highlighting the data security and privacy concerns. For remote, anonymous storage sites (e.g., Cloud Computing), it is crucial to provide an appropriate mechanism which assures the DOs of the safe storage of their data and safeguard against any manipulation from CSP, the site administrator, or even against unauthorized users.

Access control techniques are mainly established to guarantee that only an authorised user can get access to a particular data or system. The access control works as a policy imposed by administrations in order to enforce the systems' restriction, for example, allow or deny access as well as some other activities such as monitor all access requests made by users.

To safeguard the privacy of DOs, F. Li [39] pointed out that there are three critical solutions that need to be taken into account: 1) The use of appropriate technologies, 2) the adoption of efficient access control approaches which monitors the usage of client's data, and 3) the data must be stored in a non-readable form, which can be achieved through encryption - before transferring to the cloud storage.

Until recently, the cryptographic community was aware of two distinct models of encryption systems: symmetric and asymmetric (or public key) encryptions. In symmetric encryption, the same key is used in the encryption as well as the decryption processes. While in asymmetric encryption, two keys are used for each process: a public key is used in the encryption process and a secret key (or private key) for the decryption process.

In the mid-1980s, a new model of public key encryption emerged when Shamir stated in his paper that the user identity could be used to generate an encryption key and a decryption key [21]. The first practical implementation of Shamir's design was published by Boneh and Franklin [18] in 2003, which they coined as identity-based encryption or identity-based cryptosystem. Boneh and Franklin adopted the users' unique identities, such as emails, phone numbers, passport numbers, and other unique identifiers, to generate their public keys as well as the corresponding private keys. The IBC systems typically consist of three major parties; senders, recipients, and an authority server called the private key generator (PKG) or key centre generator (KCG).

3.2 Access Control Systems

In access control systems, two standard terms are usually linked—Subject (Su) and Object (Ob). The Su refers to a party that plays a decisive role in the implementation of an activity, while the latter points to the *target* in which the Su expects to reach. The target could be data, executable applications, or services. It is necessary to note that Su and Ob in computing terminology refer to software components and/or human users. There are mechanisms that manage access requests by the Su, they are called 'Access Control Mechanism'. Access Control Mechanism (ACM), is "the logical component whose principal role is to coordinate and manage access requests issued from the Su and to make a decision then force the consequence of the decision" [40].

The access control system endeavours to preserve the Ob from any potential unauthorised access by Su. From the perspective of Information Engineering, the concept of access control is related to distinct operations—for instance—read, write, share, edit, discover, execute, or delete any Ob(s) owned by an individual(s) or organisation (s) [40].

Moreover, conducting any of these operations, the Su should have permission (or an authorisation)[39]. In this context, an owner (individual or organisation) should also have the authority to define an access policy that identifies the operation of Su as well as the Ob. If Su meets the requirements of access control defined by the owner, Su should be authorised to carry out the operation.

The researchers in [4][41][6] noted that up-to-date access control systems are classified into two fundamental categories: one, a Su can get access to an Ob based on particular references or capabilities pre-defined with the Ob. Two, a Su can get access to an Ob based on an access control list (ACL). For example, if a person possesses a house key,

he/she is then able to go in. As known, such capability can easily be moved to another entity (or person in the example) to gain access. Alternatively, in ACL, the process of getting the Ob by the Su will rely on the permission list (PL) preset with each Ob. The Su is granted access to particular Ob as long as the Su's identity exists in PL.

3.3 Access Control Models

Generally, there are two main models of access control systems in which either Discretionary and Mandatory (or non- discretionary) access control.

3.3.1 Discretionary Access Control (DAC)

F. Meade [43] defined the discretionary access control (DAC) as "*a set of tools that are adapted to make the access operation of subjects (users) to the objects (targets) primarily depended on their identities and/or the group that they may belong. Access control can be said to be discretionary if the subjects can pass their permissions (perhaps indirectly) to another subject*". Therefore the DAC imposes a mechanism that allows only the Su who has the right identity to access the Ob. The DAC model is common in most operating systems, for example, Windows, Linux, Macintosh and most versions of Unix [44]. By using the DAC mechanism, every user can create a file and decide what type of access privileges to grant to another user. So, when someone requires access to the file, the operating system and based on the installed access privileges will decide whether to accept or reject the access to that file.

On the other hand, in NIST publication, J. T. Force argued that the access policy of the DAC could give the Su access to the data and execute a set of activities[45]. These activities include: 1) conveying data to another subject or object; 2) passing its privileges to another subject; 3) altering security attributes associated with subjects, objects, information system, or even system setting; 5) identifying security attributes to be related with recently-created or revised objects; 6) altering the rules that establish access control. It is essential to note that the Su could execute at least one of the above activities at runtime.

The DAC model by a user X, which gives them the right to access to specific file or data, allow the user X to pass the permission to another user Y without the consent of the data owner. However, there are concerns that there is no reasonable control of the data owners about who accesses and uses their data after migrating it to a specific user. Consequently,

this is a significant dilemma that prevents the adoption of this model of access control systems. For this reason, Ferraiolo et al. [46] pointed out that a proper using the DAC in supporting security processing necessities of industry and civilian government.

3.3.2 Mandatory Access Control (MAC)

In mandatory access control (MAC) [47] or sometimes called non-discretionary access control, the interaction between subjects and objects is based on a set of rules and security attributes which are installed for both Su and Ob. Determining whether accepting or rejecting critical access between the Su and the Ob based on matching their security attributes. Ferraiolo et al. [46] suggested that an entirely appropriate of using the MAC in supporting security processing that requires a multi-level secure military application.

There are two essential schemes that are selected for performing the MAC; role-based access control and attribute-based access control [42].

3.3.2.1 Role-Based Access Control (RBAC)

In the mid of 1990s, the concept of Role-Based Access Control (RBAC) was introduced by [48] when they added the notion of role among the Su and permission. Set of rules were suggested to arrange the interaction between the subjects and the objects that only authorised subjects are given access to the required objects. The main difference between the RBAC and the DAC is access permission cannot be passed by Su to another. In RBAC, the system uses means (as a label) to impose access constraints over the objects. The constraints mainly depend on how the sensitivity of the data or files involved in the Ob. Also, whether the Su can get access to these data of such sensitivity (i.e. whether Su has clearance or not).

Typically, the RBAC systems are applied in enterprises and organisations with more than 500 employees to provide multi-level security. For such a number of workers, it is necessary to adopt roles that cover various job functions, for instance, managers, lecturers, researchers and so on. In the RBAC, permissions for access control are associated with the roles. Therefore, the Su needs to be a member of the convenient role that has access permission to the required Ob. Figure 3.1 illustrates the relationships that can bind the subjects, roles, objects, as well as the transformation procedures [49].

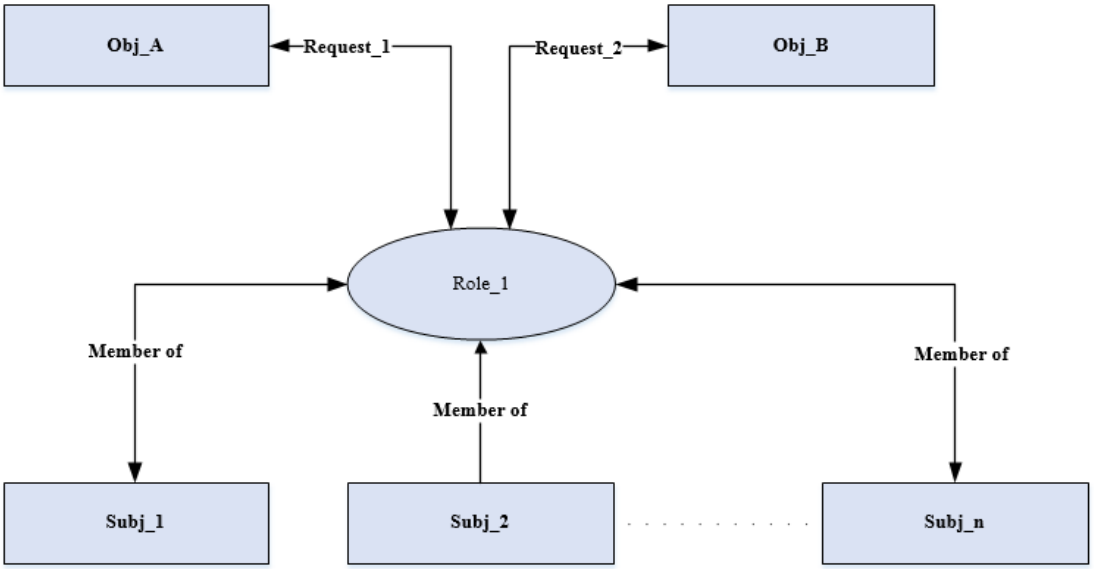


Figure 3.1: Role relationships, "adopted from[39]"

Besides, a typical RBAC system is shown in Figure 3.2.

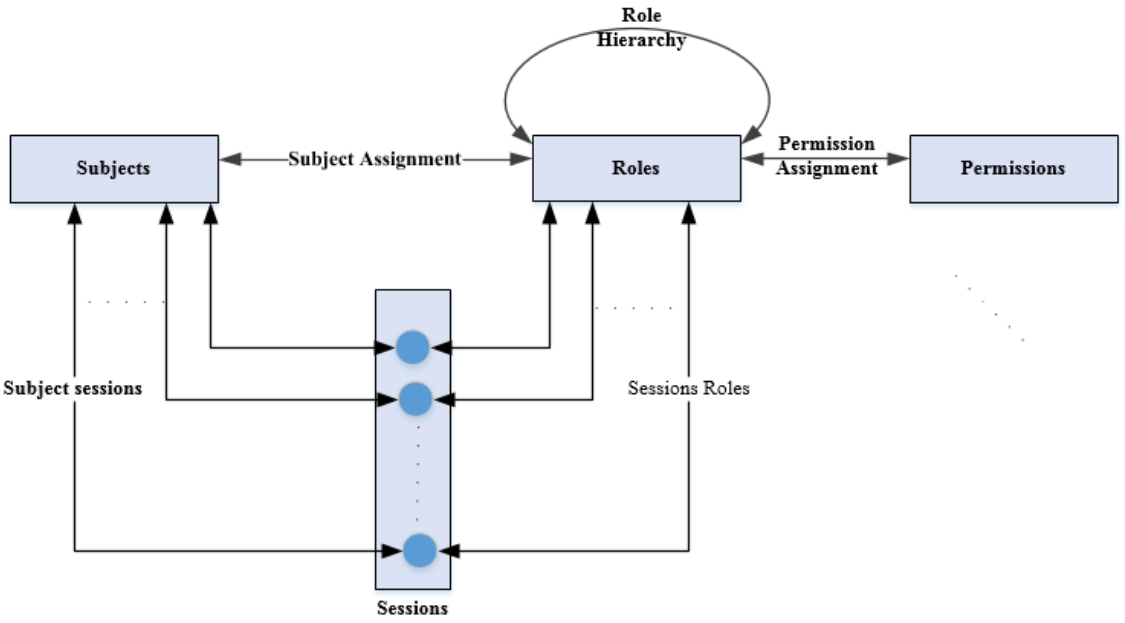


Figure 3.2: A typical RBAC system, " adopted from [39],[12]"

Furthermore, there are three primary rules in the RBAC systems, which are :

i. **Role Assignment (RAs):** The subject should have a dedicated role to gain permission and the following formal describes this rule: *" RAs (s: subject) = { the active role for subject s}"* [39], [42], [46].

ii. **Role Authorization (RAu):** Each subject could be authorised to execute one or more transactions, and it takes the following form: *"RAu(s: subject) = { authorised roles for subject s}*" [39], [42], [46]

iii. **Transaction Authorization (TA):** Each role could be authorised to implement one or more transactions, and a formal description is: "*TA(r: role) = { transaction authorised for role r}*" [39], [42], [46].

In addition to the advantages of using the RBAC, there is also a challenge associated with the implementation of this type of access control. The challenge is that the RBAC supports fine-grained access control based on a particular role for a single Ob.

3.3.2.2 Attribute-Based Access Control (ABAC)

Attribute-based access control (ABAC) is a technique that provides an access policy based on a computational language and a set of distinctive attributes, making it preferable for many companies and organisations [50]. It is vital to say that the access control list ACL and RBAC are deemed in some way as special kinds of ABAC, since they are also focused on the attribute of identity and the attribute of a role, respectively. Commonly, once the user requests access, the ABAC, in turn, constructs an access control decision according to the attributes that are assigned with the requester, the attributes are assigned to the Ob, the environment's conditions, as well as the policies that identified as regards to those attributes and conditions. One typical example of ABAC is the Extensible Access Control Markup Language (XACML) system [39], [51] that was designed to assist the most authorisation systems' requirements. Linguistically, XACML identifies using an Extensible Markup Language (XML) to designate a privacy policy language using the attributes. XACML may also represent as an arbitrary tree of sub-polices. An Ob is depicted by a tree, while the leaves represent the rules [52]. Four main factors, involved in the construction of the XACML system:

1. Policy Administration Point (PAP): It is responsible for establishing the policies or policy set.
2. Policy Decision Point (PDP): It is responsible for evaluating an appropriate policy and making an authorisation decision.
3. Policy Enforcement Point (PEP): It refers to the system entity responsible for executing access control. It releases a decision request to the PDP and implements the access decision obtained from the PDP.
4. Policy Information Point (PIP): It points to the location, where to store the attributes.

Figure 3.3 describes how data flows between the four factors of XACML system.

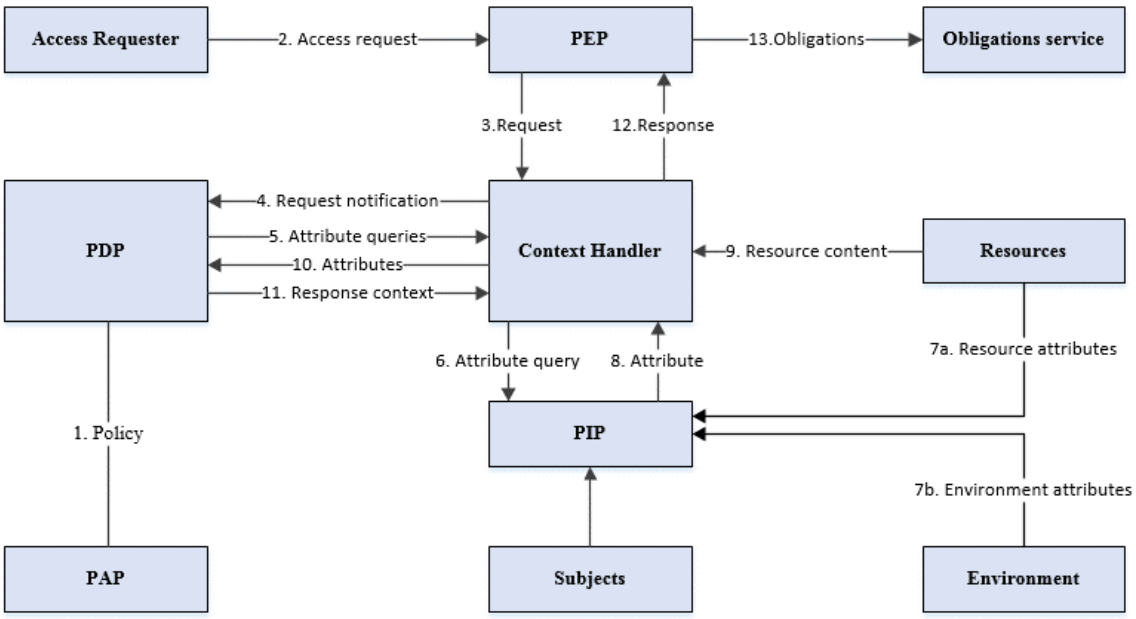


Figure 3.3: An XACML Dataflow, "adopted from[53]

3.4 Identity-Based Cryptosystems

This section introduces a modern type of Public Key Infrastructure, namely Identity-Based Cryptosystem. The section further describes the core components that contribute to the building and implementation of our IBC systems. The next two subsections introduce the two essential schemes of the IBC – standard IBC and fuzzy IBCs.

3.4.1 Standard Identity-Based Cryptosystem

Identity-based cryptosystems (IBC) was first introduced as a proposal to solve the limitations of the PKI by Shamir in [21]. The IBC remained as a theoretical idea until the first practical application carried out by Boneh and Franklin in their paper [18] using "groups" which provide a novel use through an efficiently computable bilinear map. Commonly, the IBC system was presented as an alternative approach to traditional public-key encryption in which the public key of a user is some unique information, which is represented in the form of a string about the user's identity, for example, an email address, telephone number, driver license and so on.

Besides, three significant parties play a crucial role in constructing the IBC systems, which are, the sender, receiver, and the third trusted party called a private key generator (PKG). The PKG is in charge of generating two basic parameters, which are the Master Public Parameters (MPPs) and Master Secret Parameters (MSPs). The MPPs, which are publicly known, contribute in conjunction with the user's identity in generating the public

key (PK). In return, the MSPs will be exclusively known by the PKG, contribute in conjunction with the user's identity and the PK in generating the private key (SK). In case Alice wants to send a secure message to Bob, she can use the text-value of his email address as an encryption key using the MPPs of the system issued by the PKG. To decrypt the ciphertext, Bob needs to interact with the PKG to prove that he is indeed the owner of the email. If the proof is achieved, the MSPs will be released to create the SK.

The basic concept of IBC can be described in Figure 3.4. Alice sends Bob a message encrypted using his identity such as email, phone number even if Bob has no public key certificate. The encrypted message will then be sent via Bob's email. The private decryption key is then retrieved from the PKG. IBC scheme can offer more power to Alice because she can apply more restrictions on her encrypted message by adding more roles to Bob's ID such as {Bob's email, Access time/data, Role}. Furthermore, to decrypt the encrypted message, Bob needs to interact with the PKG to prove his character (i.e., this is my email).

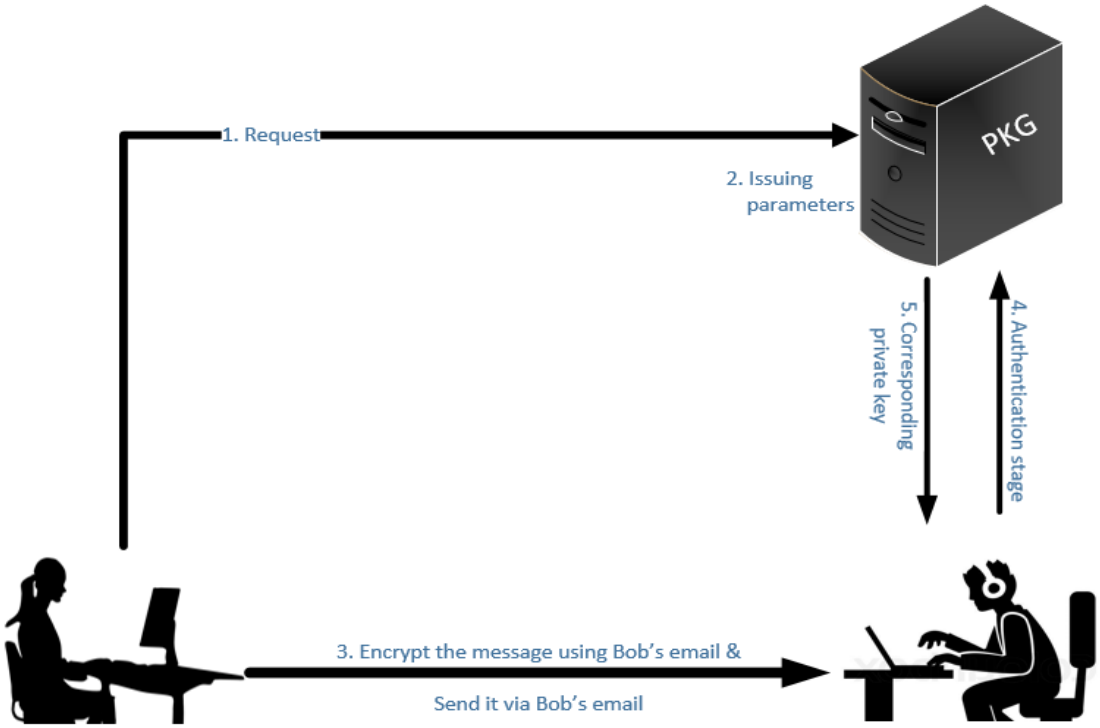


Figure 3.4: Example of identity-based cryptosystem architecture

Four main stages are needed in constructing the IBC [18], as follows.

- *Setup* (1^{λ}): This algorithm accepts a security parameter, λ, and gives back system parameters (or master public parameters) MPPs and the master secret parameter, *MSP*. However, the MPPs describe the message space, *M*, and the ciphertext space,

C. Noteworthy, the MPPs will be known by all, while the *MSP* is only known to the PKG.

- *Key Generation* (MPPs, *MSP*, *id*): This algorithm accepts the MPPs, *MSP*, as well as a public identity *id*, which is an arbitrary string, $\{0,1\}^*$. It gives back the user's private key, *sk*, that corresponds to the *id*.
- Encryption (MPPs, id, *M*): The inputs of this algorithm are the system parameters, the user identity *id*, and the message space *M*. Undoubtedly, the encryption algorithm usually gives the ciphertext $c \in C$.
- *Decryption* (*c*, *id*, *sk*): It accepts the ciphertext,*c*, the user identity, *id*, and the user's private key,*sk*. It then returns the original message $m \in M$.

For a visual representation of the correlation between the IBC's algorithms, Figure 3.5 shows the inputs and outputs for each algorithm.

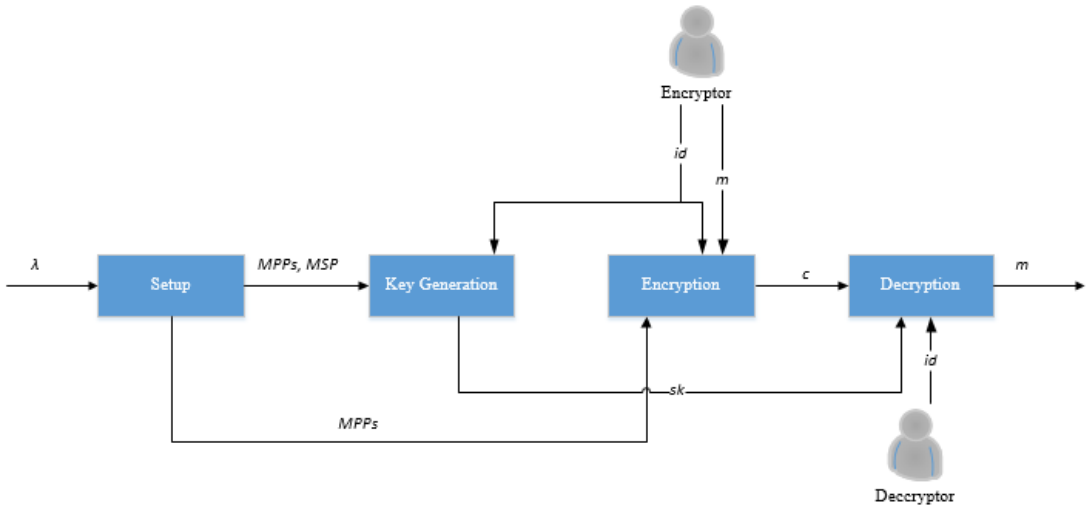


Figure 3.5: Relationship between IBC algorithms

The correctness of the IBC is attained as long as the user private key *sk* created by the key generation algorithm is mainly derived from a given identity, *id*:

$\forall m \in M$: Decryption (MPPs, *id*, *C*,*sk*) = m iff *C*= Encryption (MPPs, *id*, *m*).

3.4.2 Fuzzy Identity-Based Cryptosystem

This section gives a practical understanding of a new generation of the Public Key Infrastructure (PKI) or asymmetric keys systems. The primary challenge associated with using symmetric/asymmetric keys encryption is how to securely store and exchange the keys between different parties in an open environment such as cloud environments. The PKI has been providing a practical solution for session key exchange for many web

services. The critical limitation of the PKI solution is not only the need for a trusted third party (e.g., certificate authority) but also the missing link between the data owner and the encryption keys i.e.make encryption keys immediately linked with users' identities in cloud environments.

Fuzzy identity-based cryptosystems (F-IBCs) has been recently proposed as a new formulation of public/private key infrastructure in which the encryption keys are directly derived from a user's identity [17]. The F-IBC serves in two applications: 1) the Attribute-Based cryptography (ABC) that uses a set of particular attributes for encryption/ decryption process, and 2) the Biometric-Based Cryptography that elects the user's biometric as identity.

3.4.2.1 Attribute-Based Cryptosystems

In traditional public-key encryption, a message is encrypted for a specific receiver using the receiver's public-key whereas ABC revolutionizes this idea by linking the public key with the receiver's descriptive attributes, for instance, position, salary, age, …, etc.., The key feature of ABC is to enable data owners to share encrypted data with a set of individuals who have a matching set of attributes. A threshold to specify the minimum number of required attributes can be used to offer a better level of flexibility on who can access the data. For example, the following attributes {Dept. = Applied computing department, Status = Staff member, Age >=40, Committee Membership= exam committee member} can be used during encryption. At the decryption stage, anyone has who d-attributes (e.g. 3 out of 4 attributes) should be able to decrypt the message. If d=3, then a person with {Dept. =Applied computing, Status=staff member, Age= 42} will be able to decrypt the message. The encryption and decryption keys are generated by a trusted third party based on the set of descriptive attributes as illustrated in Figure 3.6.

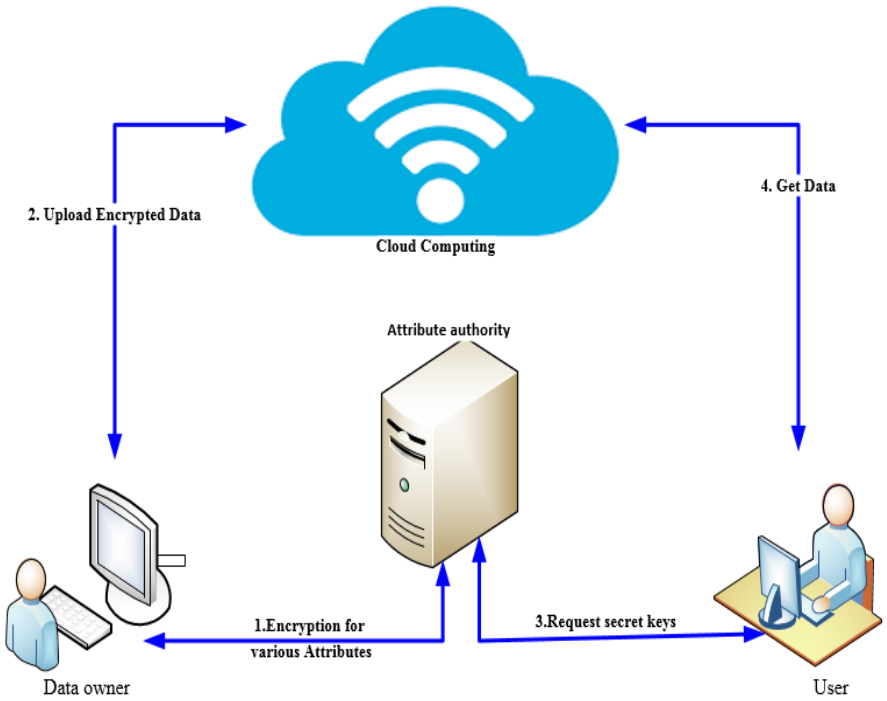


Figure 3.6: General Attribute-Based Encryption Architecture "adapted from [39]"

ABC systems subsequently developed two models which depend on whether the descriptive attributes with the user decryption key (Key-Policy ABC (KP-ABC)) or with the ciphertext (Ciphertext-Policy ABC (CP-ABC)). The content of the two next sub-sections will explain these two models of ABC.

3.4.2.1.1 Key-Policy Attribute-Based Cryptography (KP-ABC)

Goyal et al. [54] proposed the KP-ABC scheme which provides an advanced version of ABC. In KP-ABC, data owners generate master public keys to encrypt the data - such that the corresponding ciphertext is associated with a set of descriptive attributes. Each user's decryption key is associated with an access policy. The association is a tree-like structure that identifies which encrypted message can be decrypted by the key. The users' descriptive attributes are represented as the leaf nodes. Any decryption process will not be accomplished unless the accompanying attributes of the encrypted message match the key access structure.

One application of KP-ABC could be the encryption of Audit Log Entries of a big organisation. Suppose the entries have the following structure {user name, date and time of action, type of action}, and a forensic analyst is assigned the task of carrying out a

particular investigation on the log. If the entries are encrypted using traditional cryptography, the analyst needs a secret key which will enable him/her to decrypt and access all entries. However, in KP-ABE, the analyst would be issued a secret key associated with a specific access structure, through which the corresponding decryption key enables a particular kind of encrypted search such as accessing log entries whose attributes satisfy the conditions {"username = John" OR (access date between 01/01/2017 and 01/06/2017)}. The KP-ABC also makes it unfeasible for multiple analysts to access unauthorised entries from the audit log even if they collaborate and share their keys [54]. Another simple example of KP-ABC is illustrated in Figure 3.7.

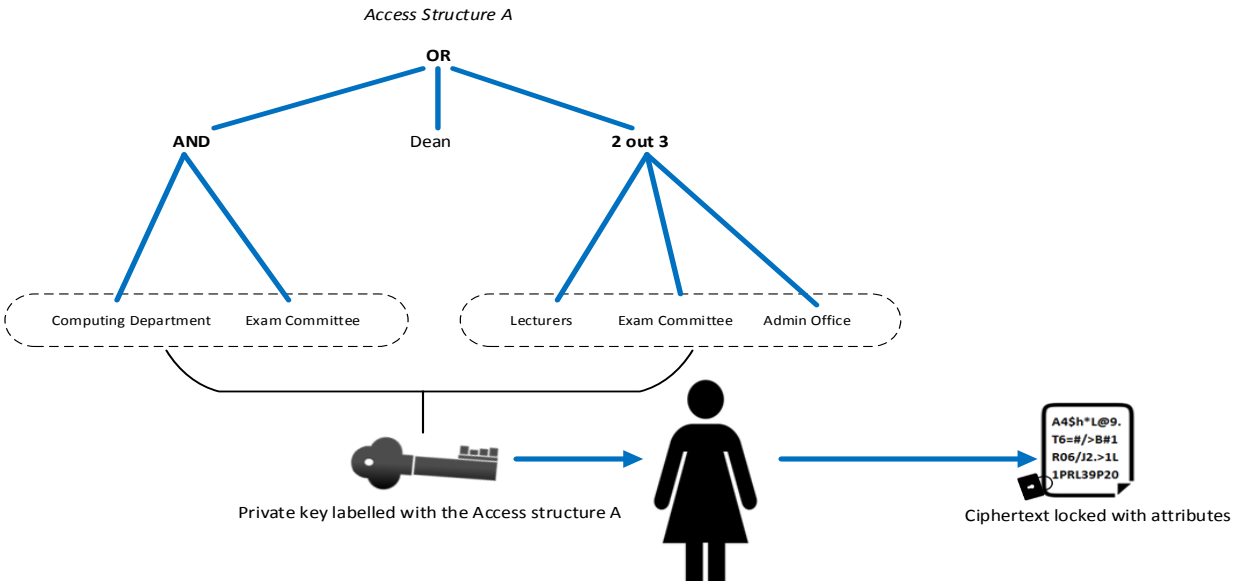


Figure 3.7: Key-Policy Attribute-Based Cryptography model

The above figure shows a simple access structure which dictates who can retrieve the decryption keys. In this case, Alice would be able to access the decryption key and unlock the ciphertext or part of it if and only if his/her attributes satisfy the corresponding access structure (i.e. she has to be a {Dean OR (a member of Computing Department AND Exam committee) OR (she belongs to two of the three: Lectures, Exam Committee, Admin Office)}).

3.4.2.1.2 Ciphertext-Policy Attribute- Based Cryptography (CP-ABC)

In various distributed systems, any attempt to gain resources is mainly depended on the user's attributes. Therefore, the core limitation of adopting the KP-ABC systems is that data owners have no control over who can access the encrypted messages; because the

access policy which is typically managed by a third party - Private Key Generator (PKG) - is not attached with the ciphertext (i.e. the access policy controls the access to the decryption keys instead of controlling the access to ciphertext).

One year after KP-ABC was introduced, Bethencourt et al. [31] presented a new construction of ABC systems that focused on the ciphertext to solve the dilemma of the KP-ABC. They claimed that the new system, Ciphertext-Policy Attribute-Based Cryptography (CP-ABC) works safely even with an untrusted server.

On the other hand, the CP-ABC shifts the focus to the ciphertext by giving data owners the power of locking their encrypted data with different access policies .i.e., for each message they can decide on who can decrypt that particular message (see Figure 3.8).

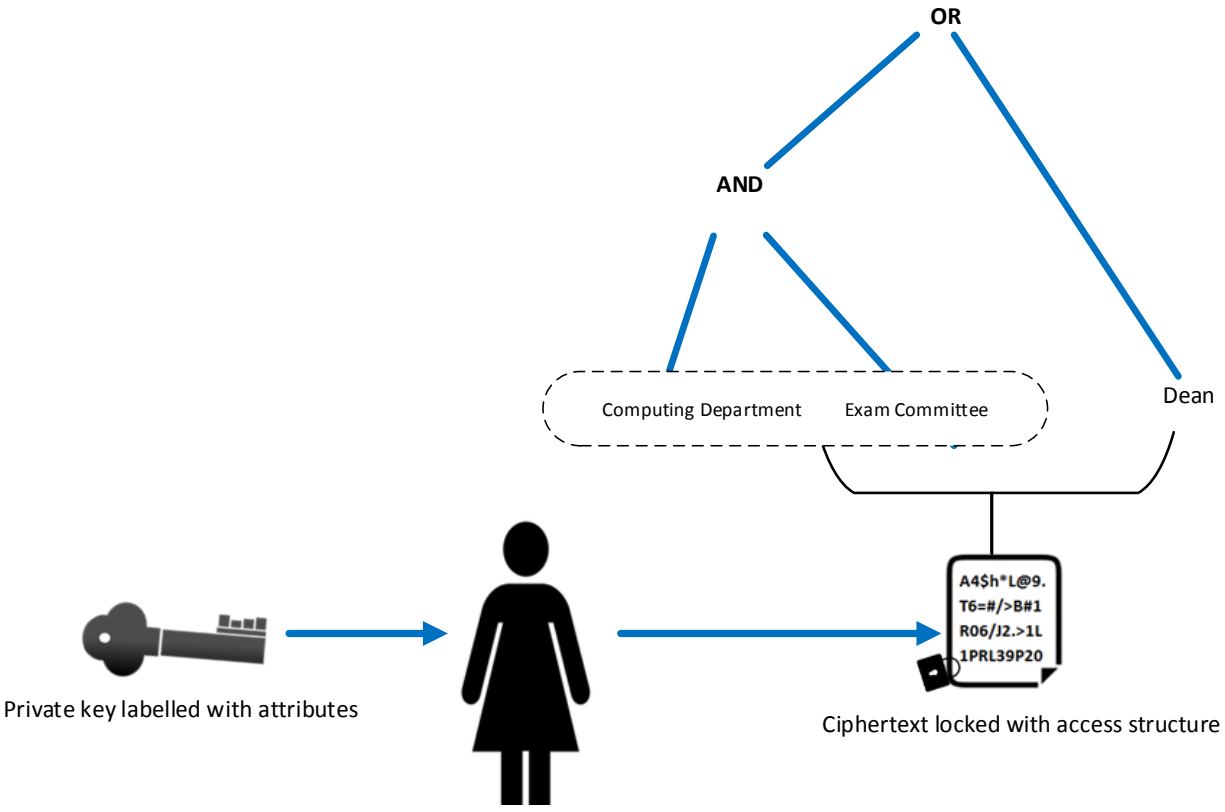


Figure 3.8: Ciphertext Attribute-Based Cryptography Model

The example is given above demonstrates the flexibility provided to by CP-ABC in locking different messages with different access structures (i.e. data owners are able to choose access policies based on the sensitivity and the security of their encrypted messages). For example, the figure shows that the encrypted message in this scenario can only be decrypted by the Dean OR (a member of both Computing and Examination staff).

3.4.2.2 Biometric-based Encryption (Fuzzy Identity-based cryptosystems)

The initial idea of Biometric Based Cryptosystem was presented in [17] where the identity was modelled as a set of descriptive attributes. The key feature of F-IBC is that the private key of an identity x has an ability to decrypt a ciphertext that has been encrypted with another identity y if and only if the distance between x *and* y is less than or equal to a certain threshold value. The F-IBC plays a key role in utilizing biometric data such as fingerprints or faces images as identity. F-IBC is a promising solution that bridges the gap between the exactness of encryption/decryption keys and the fuzziness of biometric data (i.e. the enrolled biometric samples and the freshly captured ones are never the same). This feature enables a private key of biometric identity to decrypt a message that was encrypted with a public key of a slightly different biometric identity.

It can be argued that the weakness associated with the use of traditional IBC is that the identity such as a "name" or "email" needs to be authenticated first before retrieving the corresponding decryption key in [18][55][56]. Therefore, the user might need to provide additional "supplementary documents" to link the name and/or the email with his/her identity. In contrast, the biometric-based F-IBC offers a natural way of authentication by providing biometric data, which is part of the user's identity, to retrieve the decryption private key. It has been proved that F-IBC can withstand collision attacks (i.e. a group of users cannot integrate their keys in a manner that enables them to decrypt messages without individual permission).

In F-IBC, the user's private key is a set of *n* private components or features that are linked within the identity of the user. Shamir's secret sharing [57] is typically employed to distribute the master secret key over the components of the private key by using a polynomial of degree (d-1) where $d <= n$ is the minimum number of private components that the user needs to present to retrieve the decryption (private) key. Figure 3.9 outlines an example of two identities *X* and *Y* where the number of overlapped features is 13 out of 20 features. If the threshold *d* is set to be 10 or more, then the two identities will be deemed to be the same.

Mathematically, F-IBC scheme depends primarily on the concept of the groups (see chapter two). Assume $\mathcal{U}$ refers to the universe of size $|\mathcal{U}|$, identity elements then form a

subset of $\mathcal{U}$. There is also a unique element ($\in \mathbb{Z}_p$) associated with each element in the identity.

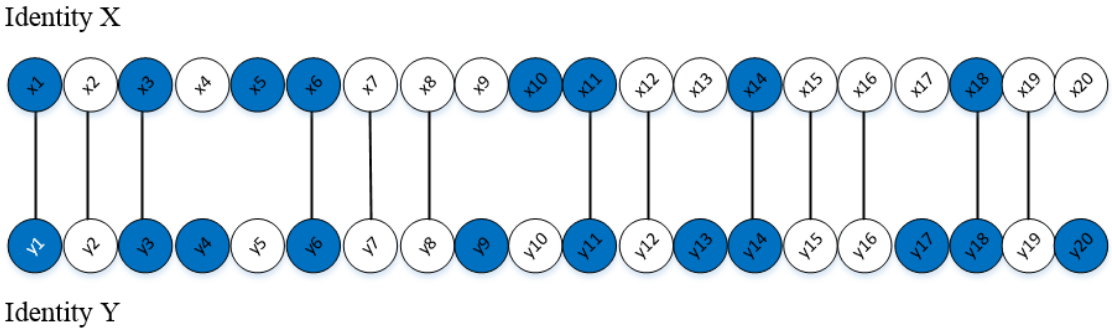


Figure 3.9: Two identities X, Y with 13 out of 20 overlaps

Besides, for each $i \in \mathbb{Z}_p$ and set S of the element($\in \mathbb{Z}_p$), they exploited the LaGrange coefficient to implement the Shamir's sharing secret as follows:

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$$

As the standard IBC, four fundamental algorithms are involved in the construction of F-IBC as described in the following:

- ***Setup**:* This algorithm takes as input the following parameters:
  - Define the elements in the universe $\mathcal{U}$ of size $|\mathcal{U}| \in \mathbb{Z}_p$
  - Picks uniformly at random integers $t_i \in \mathbb{Z}_p$ such that $i \in \{1... |\mathcal{U}|\}$.
  - It finally at random selects an element y uniformly $\in \mathbb{Z}_p$.

The *setup algorithm* after that publish the following parameters: $T_i = g^{t_i}, Y = e(g,g)^y$, where $i \in \{1,.., |\mathcal{U}|\}$as the MPPs and retains the following parameters only for the PKG: t_i, y as the MSPs.

- ***Key generation**:* Its responsibility is to generate the corresponding private key of the identity $\in \mathcal{U}$. To do this, a series of steps must be made as described:
  - Picks randomly a polynomial (q) of degree (*d-1*) so that $q(0) = y$.
  - The corresponding decryption key will then take the following form:

$$Sk_i = g^{q(i)/t_i} \; \forall \; i \in w$$

- ***Encryption***: It takes the public key $\mathfrak{W}$ and a message $M \in \mathbb{G}_1$ as well as random ($s \in \mathbb{Z}_p$). The resulted ciphertext includes the following components:

CT = $(w', MY^s, \{E_i = T_i^s\}_{i \in W'}$. As can be seen, the identity w' is contained within the encrypted message.

- ***Decryption***: it takes the encrypted message, CT, that encrypted under the key of identity w' and by using the corresponding decryption key, Sk_i of the identity w, the corresponding original message will then be reconstructed if the value of $|w \cap w'|$ is greater than d. The following steps summarise the decryption of the ciphertext:

$$MY^s / \prod_{i \in S} (e(Sk_i, T_i^s))^{\Delta_{i,S}(0)}$$
$$= M.e(g,g)^{sy} / \prod_{i \in S} \left(e(g^{\frac{q(i)}{t_i}}, g^{st_i}) \right)^{\Delta_{i,S}(0)}$$
$$= M.e(g,g)^{sy} / \prod_{i \in S} \left(e(q,g)^{sq(i)} \right)^{\Delta_{i,S}(0)} = M.$$

As a final stage, the recipient would be required to authenticate himself/herself to the PKG by presenting a fresh biometric template. The recipient then will be able to decrypt the message if the distance between the two templates is greater than or equal a pre-defined threshold. Burnett et al. [58] endeavoured to use the F-IBC scheme to build a new scheme of digital signature (BIO-IBS). To construct the BIO-IBS, they employed the client's biometric data to generate public and private keys. The scheme depends on Pairing-based signature construction to execute the signing and verification. The biometric data exercised to verify a signature in various domains of non-repudiation documents. For example, legal disputes in regards to whether a contract was signed by a genuine user or not. Typically, a user uses BIO-IBS to sign a particular contract with another person, and after a while, a disagreement occurs. For this scenario, the user is only asked to show his/her fresh biometric data to ensure the authenticity of the signature. [58]argued that due to the variation that may result over time in terms of biometric feature, they use three different matrices to get more accurate biometric measurements. To play this role, they used the Hamming Distance, Set Difference, and Edit Distance. For this reason, a Fuzzy extractor technique is deemed as an additional level of error correction to generate keys form a variable biometric measurement.

3.4.2.3 Security Evaluation of Fuzzy Identity-Based Cryptosystem

The security evaluation of F-IBC systems is on two levels. The first level relies primarily on reducing the Fuzzy-Selective Identity (F-SID) security model to the hardness of the Decisional Modified Bilinear Diffie Hellman (DMBDH) that was adopted in the first

construction of Sahai and Waters in [17]. Whereas, the second level of security analysis derives from the use of biometric cryptosystem technique.

F-SID introduced by Shaia and Waters [17], to some extent, resembles the Selective-Identity (SID) security model associated with the standard IBC suggested by [18]. The only difference is that the decryption keys can be queried for identities which possess *less* than *d-overlapping* with target identity. It gives the adversaries extra concessions which allow them to access the oracles' encryption and decryption systems. This model of security calls a "*game*" which conducts between two active parties— a challenger ($\mathcal{C}$) and an attacker (or an adversary) $\mathcal{A}$.

F-SID security game can be summarised through the following major steps:

- ***Initial.*** $\mathcal{A}$ determines an identity ($\mathcal{ID}$) to be the challenged identity.
- ***Setup.*** $\mathcal{C}$ generates both of the master public parameters (MPPs) to be sent to $\mathcal{A}$, and master secret parameter (MSP) using the setup algorithm.
- ***Private keys' Queries.*** It conducts a batch of private key's queries by $\mathcal{A}$ for several identities (ID_j) with the condition that $|\mathcal{ID} \cap \mathcal{ID}_j| < d$ for all j.
- ***Challenge.*** Two equal length plaintexts m_0 and m_1 are sent from $\mathcal{A}$ to $\mathcal{C}$. One of these messages is randomly chosen by $\mathcal{C}$ by throwing a random coin (β). Generate an encrypted message by encrypting the selected message (m_β) with $\mathcal{ID}$ and send the resulting encrypted message to $\mathcal{A}$.
- ***Repeat The Private Keys' Queries***.
- ***Estimation.*** While the adversary $\mathcal{A}$ decides to finish the previous step, the next step will be to issue a guessing β' of β.
- ***Measuring $\mathcal{A}$'s Advantage***. The last step determines the attacker's advantage to win the above game as follows:

  $\Pr[\beta' = \beta] - {}^{1}/_{2}$.

Concerning Fuzzy Selective-Identity (F-SID) game, any proposed scheme considered secure if there is no polynomial-time adversary able to win this game with non-negligible advantage.

Theorem 3.1- If an adversary in a selective Fuzzy ID model breaks our scheme, a simulator can be built by this adversary to win the DMBDH assumption with non-negligible advantage.

The DMBDH (an acronym for Decisional Modified Bilinear Diffie-Hellman) assumption defined by Sahai and Waters as follows [59]:

DMBDH- Assume a, b, c, z are randomly selected from $\mathbb{Z}_p$ by a challenger $\mathcal{C}$. The Decisional MBDH assumption points out that, it is hard for a polynomial-time adversary to distinguish the tuple $(A = g^a, B = g^b, C = g^c, and\ Z = e(g,g)^{\frac{ab}{c}})$ from the tuple $(A = g^a, B = g^b, C = g^c, and\ Z = e(g,g)^z)$ with non-negligible advantage.

Proof

Assume $\mathcal{A}$ is a polynomial-time adversary which able to break the scheme using the F-SID game with non-negligible advantage (ψ); thus $\mathcal{A}$, based on the theorem 3.1, will be used to construct a simulator ($\mathcal{B}$) capable of winning the DMBDH game with the advantage $\psi/2$. The simulation carries out as follows:

Firstly, symmetric groups $\mathbb{G}_0$ and $\mathbb{G}_1$ are set up by the challenger $\mathcal{C}$ with an efficient bilinear map, $\hat{e}: \mathbb{G}_0 \times \mathbb{G}_0 \longrightarrow \mathbb{G}_1$, and the generator $\langle g \rangle$ of$\mathbb{G}_0$. After that, a binary coin ($\mathcal{B}$) is fairly flipped by $\mathcal{C}$. For random $a, b, c, z \in \mathbb{Z}_p$, the challenger ($\mathcal{C}$) outputs the tuple $(A, B, C, Z) = \left(g^a, g^b, g^c, \hat{e}(g,g)^{ab/c}\right)$ if $\mathcal{B}$ =0; otherwise, it outputs the tuple$(A, B, C, Z) = (g^a, g^b, g^c, \hat{e}(g,g)^z)$. Presently, Currently, it has generated all the elements of the group, and they became ready to play F-SID game. Assume that the universe ($\mathcal{U}$) is defined.

- ***Initial***. The simulator $\mathcal{B}$ executes the adversary $\mathcal{A}$ and gets the challenge identity ($\mathcal{ID}$).
- ***Setup***. It runs by the simulator $\mathcal{B}$ to generate the elements of MPPs and MSP as demonstrated below:

- It sets up the parameter $Y = \hat{e}(g, A) = \hat{e}(g, g^a) = \hat{e}(g, g)^a$.
- It picks at random $\beta_i \in \mathbb{Z}_p \; \forall \; i \in \mathcal{ID}$, then compute $T_i = C^{\beta_i} = g^{c\beta_i}$.
- It picks at random $\omega_i \in \mathbb{Z}_p \; \forall \; i \in \mathcal{U} - \mathcal{ID}$, then computes $T_i = g^{\omega_i}$

Finally, it submits the resulted MPPs to $\mathcal{A}$.

- ***Private keys' Queries.*** $\mathcal{A}$ asks for private keys for a batch of identities ($\mathcal{ID}_j$) such that the overlapping between each one and $\mathcal{ID}$ is less than *d*. Next, for each identity δ in $\mathcal{ID}_j$ in which the above condition is met, Sahai and Waters suggested to define three sets, Γ, Γ', *S* as follows:

  - Γ= δ ∩$\mathcal{ID}$.
  - Γ' refers to any set with the constraint that Γ⊆ Γ'⊆δ and | Γ'|=*d-1*.
  - *S*= Γ' ∪ {0}.

The subsequent step is to set up the decryption key ingredients (D_i) for $i \in \Gamma'$ as follows:

If $i \in \Gamma$ sets up $D_i = g^{s_i}$, where s_i chosen randomly from $\mathbb{Z}_p$

Also, for a polynomial q(x) of degree (d-1), they selected :

$q(0) = a \; and \; q(i) = c\beta_i s_i$.

If $i \in \Gamma' - \Gamma$: $D_i = g^{\frac{\lambda_i}{\omega_i}}$, where λ_i chosen randomly from $\mathbb{Z}_p$, and $q(i) = \lambda_i$.

The simulator computes D_i for $i \notin \Gamma'$ as follows:

$$D_i = \left(\prod_{j\in\Gamma} C^{\frac{\beta_j s_j \Delta_{j,S(i)}}{w_i}}\right)\left(\prod_{j\in\Gamma} g^{\frac{\lambda_j \Delta_{j,S(i)}}{w_i}}\right) Y^{\frac{\Delta_{0,S(i)}}{w_i}}$$

Using Lagrange Interpolation, the simulator can compute $D_i = g^{q(i)/t_i}$, which gives the simulator ability to construct the decryption key for the identity $\mathcal{ID}$.

- ***Challenge.*** $\mathcal{A}$ submits two equal challenge messages $m_0 \; and \; m_1$ to the simulator $\mathcal{B}$ which in turn randomly throws a fair coin (b) and using the encryption algorithm will encrypt m_b to output the following ciphertext:

$$E = \left(\mathcal{ID}, \mathrm{E}' = \mathrm{m_b} Z, E_i = B^{\beta_i}\right)_{i\in\mathcal{ID}}$$

If $\mu = 0$, then $Z = e(g,g)^{\frac{ab}{c}} = e(g,g)^{ar'}, where \; r' = \frac{b}{c}$.

Hence, $\mathrm{E}' = \mathrm{m_b} Z = \mathrm{m_b} e(g,g)^{ar'} = \mathrm{m_b} Y^{r'}$ and $E_i = g^{b\beta_i} = g^{\frac{b}{c} c\beta_i} = g^{r' c\beta_i} = (T_i)^{r'}$

Otherwise, if $\mu = 1$, it results $Z = g^z$, then $\mathrm{E}' = \mathrm{m_b} e(g,g)^z$. Since z is random, thus the resulting E' considers a random in $\mathbb{G}_1$ from adversaries view; therefore, information about b.

- ***Repeat The Private Keys' Queries***.
- ***Estimation.*** An estimation b' of b is submitted by $\mathcal{A}$.

If $b' = b$ then $\mathcal{B}$ will give $\mu' = 0$ as indicating to the DMBDH tuple, otherwise it will give a random 4-tuple.

If $\mu = 1$, the adversary gets no information about b and therefore the probability will be: $\Pr[\, b' \neq \mathrm{b} | \mu = 1] = \frac{1}{2}$.

Since when $b' \neq \mathrm{b}$, $\mathcal{B}$ guesses $\mu' = 1$,

then it has $\Pr[\, \mu' = \mu | \mu = 1] = \frac{1}{2}$.

If $\mu = 0$, then the adversary reveals encryption of $\mathrm{m_b}$ with an advantage ψ (by definition). Therefore, $\Pr[\, b' = \mathrm{b} | \mu = 0] = \frac{1}{2} + \psi$. Similarly, in case $\mu' = \mu$, $\mathcal{B}$ guesses $\mu' = 0$ and thus $\Pr[\, \mu' = \mu | \mu = 0] = \frac{1}{2} + \psi$.

As a result, the overall $\mathcal{B}$'s advantage in the DMBDH security game is:

$$\frac{1}{2}\Pr[\, \mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\, \mu' = \mu | \mu = 1] - \frac{1}{2} = \frac{1}{2}\left(\frac{1}{2} + \psi\right) + \frac{1}{2} * \frac{1}{2} - \frac{1}{2} = \frac{\psi}{2}$$

3.5 Chapter Summary

This chapter described the essential elements of identity-based access control and cryptography directly related to the construction of specific mechanisms to control who can access data stored on a remote site, in addition to protecting its data from any tampering that could happen in the host site. Furthermore, the chapter showed that there is a close relationship between using the attributes in encryption and imposing fine appropriate access control mechanisms. IBCs is a modern form of public key encryption that utilizes the identity, e.g., an email address, telephone numbers, driver licenses, and biometric data, to be used in the generation of the encryption and decryption keys. However, IBC key-escrow and central point of attack are the main concerns that are

accompanying using IBCs. The following chapters describe a number of practical solutions to address the vulnerabilities highlighted in this chapter.

Chapter 4

One-Time Challenge-Response Multifactor Authentication for Fuzzy Identity-based Cryptosystems

As explained in Chapter 3, identity-based cryptosystems (IBCs) are new generations of public key encryptions that depend on two asymmetric keys to perform encryption and decryption. Fuzzy identity-based cryptosystems (F-IBCs) is a promising extension of IBCs that relies on biometric modalities instead of traditional identities utilised in IBCs. In a typical F-IBC, users verify themselves to a Private Key Generator (PKG) using their public biometric modality such as face images before retrieving decryption keys.

This chapter argues that existing F-IBC systems have a serious security vulnerability related to releasing decryption keys without proper user authentication. In fact, security relies on the assumption that biometrics can be only presented by genuine users/owners, which is an unrealistic assumption. Arguably, the security of such systems can be compromised e.g. by obtaining a face image of the target user from their social media profiles and submitting it to the PKG for verification. Furthermore, existing F-IBCs do not address any of the general privacy and security concerns related to biometric data.

The chapter proposes a new solution to address the above vulnerability using a One-Time Challenge-Response Multifactor Biometric Authentication at the user verification stage. The proposal also incorporates the use of cancellable biometrics to provide additional privacy layer.

The chapter is structured as follows. It starts with a brief summary of IBCs and F-IBCs features followed by explaining the security vulnerabilities associated with existing F-IBCs in Section 4.2. Sections 4.3 and 4.4 describes the proposed solution and its main algorithms. Security analysis of the proposed solution and the experimental results are

presented in Section 4.5, whereas Section 4.6 concludes the research carried out in the chapter.

4.1 Introduction

Identity-based cryptosystems (IBCs) are a modern form of PKEs that were introduced to address the difficulties associated with the use of traditional PKE as explained in Chapter 3. This research contends that IBCs provide a great deal of flexibility required in different environments such as Cloud Computing because they bind decryption keys with user identities, and therefore, DOs can be embedded in determining who can access their encrypted data. This feature cannot be provided by traditional PKEs.

Standard IBCs have three principal parties: a central authority server called Private Key Generator (***PKG***), a sender (e.g. ***Alice***) who is the **DO** and a receptor (e.g. ***Bob***). The encryption and decryption processes rely primarily on users' identities such as an email address, telephone number, driver's licence number and passport number as well as some other parameters issued by the PKG to produce asymmetric keys. In IBCs, the PKG is typically responsible for generating two basic sets of parameters: Master Secret Parameter (MSP) and Master Public Parameters (MPPs). While the latter could be known to everyone, the former is strictly known to the PKG only. For example, if Alice wants to send an encrypted message to Bob, she will use his public key derived from his identity (e.g. his email address) and the MPPs. Bob, on the other hand, retrieves his decryption key generated by the PKG based on the MSP in addition to Bob's identity and the MPPs i.e. the MSP managed by the PKG is the vital element to the secrecy of the decryption keys.

Fuzzy identity-based cryptosystems (F-IBCs) are advanced models of the IBCs that were initially introduced by Sahai and Waters [59] (see figure 4-1). F-IBCs typically employ biometrics to overcome the authentication limitation associated with traditional IBCs schemes that require the users to submit some supplementary documents/information to authenticate themselves to a server before the PKG releases the decryption keys [59]. In addition to the practicality of traditional IBCs, the dilemma is that the documents/information used for authentication could be subject to forgery [17].

On the contrary, in F-IBCs, the requirement of presenting further identification documents is ignored due to the reliance on users' public biometric data in the process of

generating encryption/decryption keys. For instance, **Alice** as a DO requires to encrypt her message, using **Bob's** public biometric data such as his face image. On the other hand, to receive his decryption key, **Bob** must present his fresh face image to the PKG as an initial stage, then the private key is issued and delivered based on the matching score as illustrated in Figure 4.1. Further details regarding the standard IBC and F-IBC are provided in chapter 3.

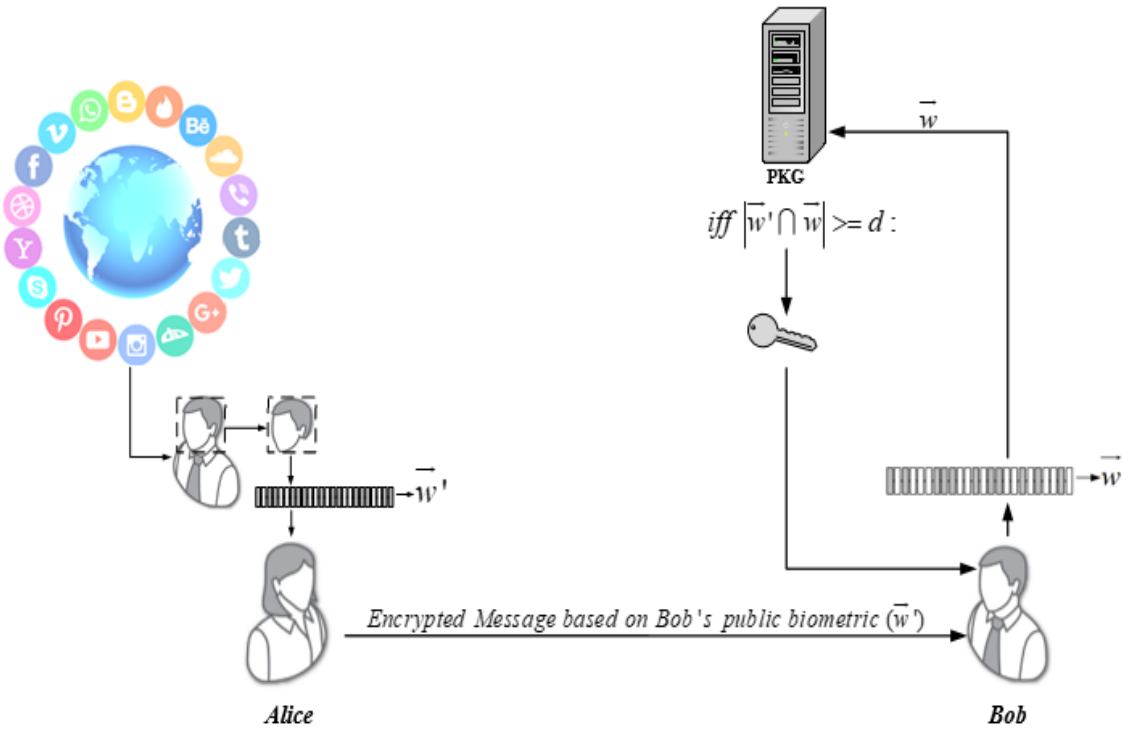


Figure 4.1: Key structure of fuzzy identity-based cryptosystem

It is important to highlight the fact that Sahai and Waters [17] assumed that the PKG cannot be fooled by an imposter, and accordingly only Bob is capable of presenting his fresh public biometric samples and receiving the corresponding private key [17]. However, the above assumption is flawed as the security of the system can be compromised as long as the F-IBCs rely solely on the user's public biometric (e.g., face image). It is not a secret that anyone these days can access face images of almost everyone using various social media platforms. Consequently, an impersonator/ non-genuine user can play the role of a genuine user (by spoofing) and thus can obtain the decryption key.

To improve the security of F-IBCs against various potential attacks, a number of proposals in the literature suggested the use of multimodal biometrics instead of a single biometric modality [17][60][61][62][63]. Further, set overlap and Euclidean distance are

popular similarity measures that have been used to measure the matching score in a single biometric model. Sarier [60] introduced a new construction of the similarity measure based on two different Modalities of biometric to encrypt/decrypt the same message, which was achieved by combining distance-based encryption (DBE) [61] and F-IBCs as in [17][62][63] schemes. The modalities combined include fingerprints, utilising different matching methods (e.g., minutia and non-minutia-based matchers), fingerprint and face recognition as illustrated in Figure 4.2 As a result, the multimodal biometric system produced two layers of encryption using two different distance measures followed by merging the two ciphertexts for the same original message.

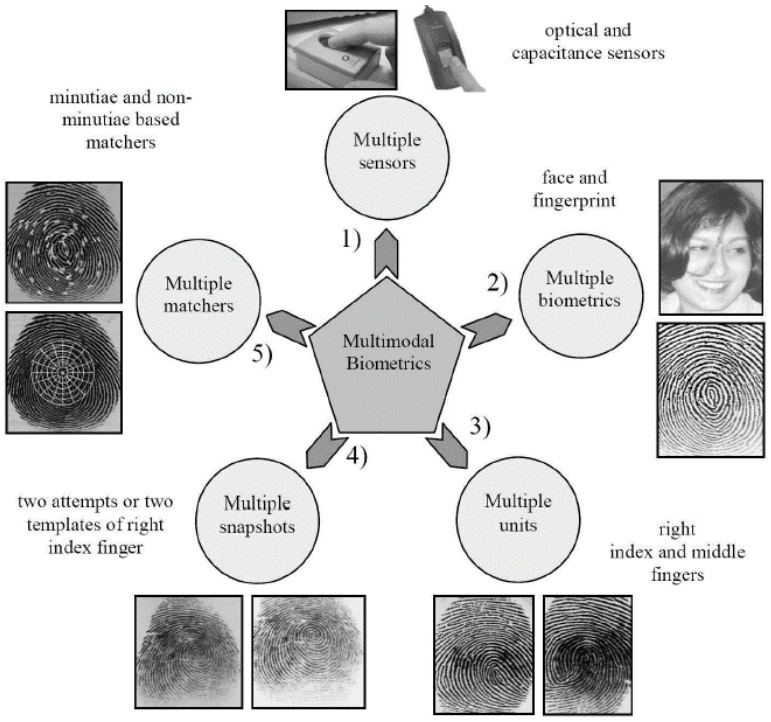


Figure 4.2: Different scenarios of multimodal biometric schemes adopted from [64]

In order to decrypt the ciphertext, the multi-modal system measures the overlap between the two pairs of biometric samples as in F-IBC.

All the above multi-modal based solutions assume that Alice as a DO has access to a copy of Bob's biometric data e.g., she should have access to his face and fingerprint. While this assumption can be valid for public biometrics such as face image, the same is not necessarily true for fingerprints. This could have a big impact on the practicality of existing F-IBCs that are based on multimodal biometrics.

4.2 Security analysis of Existing F-IBC schemes

IBCs, in general, are secure against Selective Identity model [18], which means it allows an attacker to choose a public key ID^* for the challenge. Then the attacker asks PKG the decryption keys of other public keys ID_i with the restriction that $ID^* \notin ID_i$. The resulting decryption keys may use to help the attacker. With these facilities, there must be no polynomial-time attacker can win this model with non-negligible advantage. Regarding F-IBC proposed in [17], it is secure against Fuzzy Selective-Identity model (as described in section4.5). Fuzzy Selective-Identity model is similar to Selective Identity model with the exception that the adversary is only able to ask PKG the private keys for identities ID_i with condition that $|ID_i \cap ID^*| < d$, where d is the agreed threshold value. As it has been observed, both models refer to the method of retrieving and managing decryption keys.

As mentioned earlier in the chapter, F-IBCs are insecure against impersonating attacks. In F-IBC, Sahai and Waters have assumed that impersonators/ untrusted user cannot spoof the PKG using biometric templates of genuine/ trusted users in users' authentication stage [17]. Also, all proposed F-IBCs that followed the work of Sahai and Waters, e.g. [17], [18], [60], [65]–[70]; have adopted the same assumption. Thus, it gives only Bob the capability of receiving his private key and then decrypt the encrypted message sent by Alice (see Figure 4.1). It is undoubtedly an idealistic assumption due to F-IBCs deal with public biometric samples. Figure 4.3 illustrates why the Sahai and Waters assumption was unrealistic. The scenario that can emerge is that since F-IBCs are entirely dependent on public biometric templates (e.g., face images) which are currently easily accessible from various sources (most notably social media platforms); what if an impostor attack/ any other non-genuine user (e.g., **Eve** in Figure 4.3) brings Bob's face image from his profile. This scenario means that Eve can gain the decryption key of Bob then read Alice's encrypted message. Figure 4.3 clarifies the scenario above regarding how Eve obtained the decryption key based on Bob's public biometric $\vec{w}''$ that submitted to PKG.

Existing F-IBCs use the raw biometric templates, which has security and privacy implications. The security implication is due to the potential domino effect that is similar to the use of the same password on multiple servers. If the password is compromised on one server, all the remaining servers will be compromised. The same effect will be in the event of using the raw biometric templates in biometric-based authentication on multiple servers. Once an attacker detects the biometric template, the attacker can access any

server they want. Moreover, the use of the raw biometric templates grows privacy concerns, as it makes the user traceable.

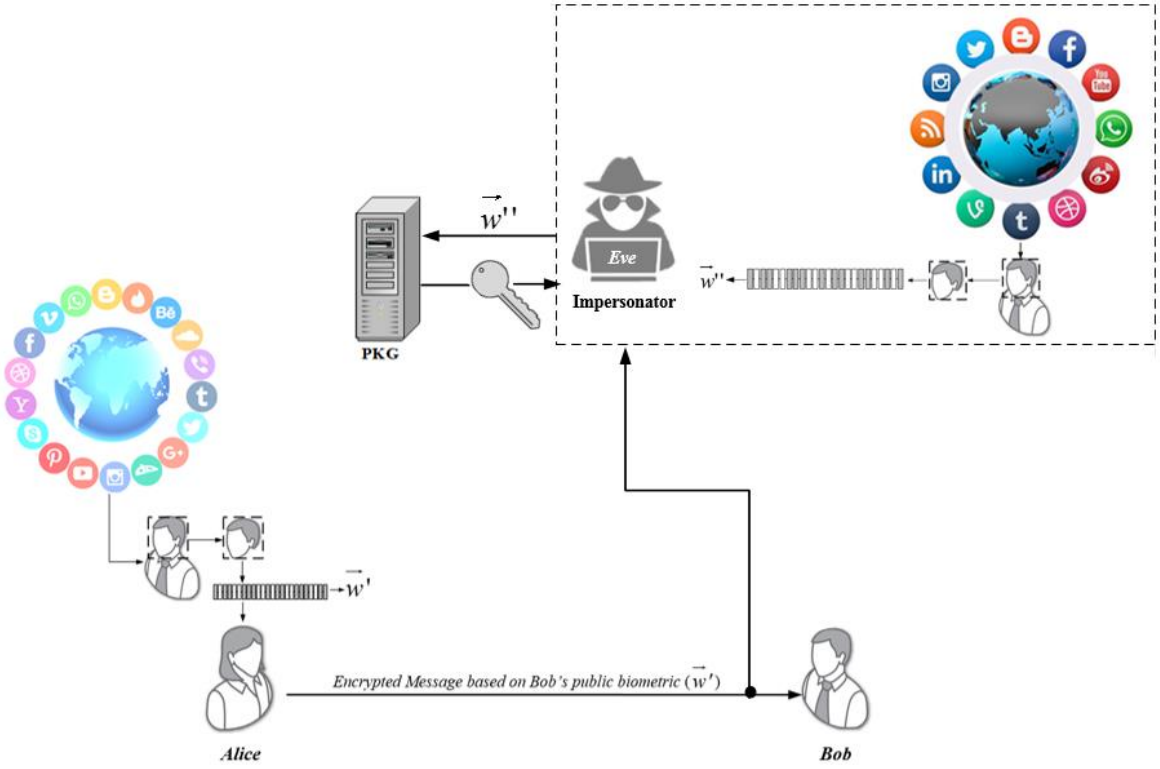


Figure 4.3: Steps that an impersonator (Eve) can follow in the use of Bob's public biometric data in F-IBCs

For these two security vulnerabilities, F-IBCs need to look for a proper mechanism that arranges the decryption keys delivery and prevents the impersonators from exploiting the weaknesses. Besides, there is an urgent need to remove the raw biometric templates from immediate use to avoid the domino effect.

However, the biometric-based authentication systems have also a challenge because the biometric is not secret. It is easy to capture face image, voice, signature as well as fingerprints then potentially abused by imposters without the permission of their owners[71][72]. Consequently, protecting the biometric templates becomes a vital issue to address which has prompted researchers in the field device appropriate techniques to maintain the security and integrity of biometric templates in biometric systems [71].

The proposed solution in this work offers a comprehensive solution to address the above two problems by enforcing a particular biometric-based authentication that organises the delivery of decryption keys to users. Cancellable/ revocable biometrics is one of the promising techniques that have been introduced to address the issue of maintaining biometrics in biometric-based authentication systems even in case of the biometric traits

are detected or stolen [71][72][73]. Therefore, the cancellable biometrics technique is the key element used to safeguard the biometric templates in the proposed solution.

The next section presents the proposed solution in which the transformed biometric is adopted at F-IBCs users' verification stage.

4.3 The Proposed Solution – A One-Time Challenge-Response Multifactor Authentication

This section presents the proposed solution to addresses the security limitation highlighted in the previous section and provides the DO (Alice) with much more control by ensuring only genuine users can retrieve the decryption keys. It also assures Bob that his biometric template is secured and cannot be used to trace him across different servers.

The proposal is a hybrid solution that combines one-time challenge-response multifactor authentication with cancellable Biometrics to improve the security as well as the privacy of existing F-IBCs. The One-Time Challenge-Response Multifactor Authentication (OTCR-MFA) increases the difficulty for imposters to compromise the PKG in order to retrieve the decryption keys of any genuine user in F-IBCs. The proposed solution has two main stages: the enrolment and authentication stages.

Enrolment stage: at this stage, users enrol their biometrics in the PKG database. Figure 4.4 shows the steps involved in the enrolment stage. It involves subjecting their biometric samples (e.g., face images) to a feature extractor to produce the corresponding features vector. After that, an orthonormal random projection based on PIN or password is applied to preserve the original biometric data by producing a secure cancellable biometric template, as proven in [74], for the user u ($\mathrm{CBio_u}$) to be stored in the PKG's database. In order to preserve a user's privacy and improve the security, a cancellable (or revocable) biometrics technique is utilised to give the user the ability to enrol different versions of their biometric data (not the original ones) in different servers. Consequently, it improves the security by preventing the domino effect of having one template stolen and replayed to all servers, in addition to protecting the privacy of the user by preventing their tracking across different services. More on cancellable biometrics is illustrated in an implementation pane in section 4.4. The figure below explains how the user is registered in the enrolment stage within the proposed OTCR-MFA.

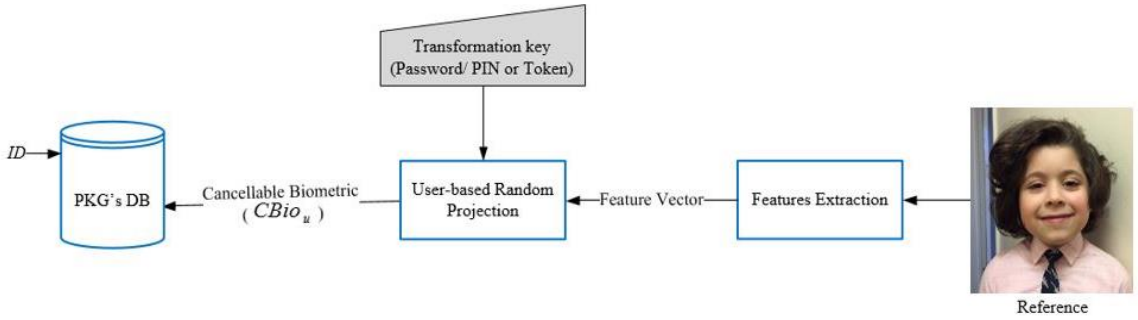


Figure 4.4: The main steps of an enrolment stage in OTCR-MFA

Authentication stage: This stage is triggered when the user u attempts to retrieve their corresponding decryption key from the PKG. The security of this stage depends on exchanging a One-Time Random Secret (OTRS) based on Diffie–Hellman key exchange protocol for every authentication attempt, as explained in Section 4.4.3.

In our proposed solution, the authentication stage composes of four main phases: (1) authentication initialization phase; (2) exchange of One-Time Random Secret (OTRS) phase; (3) generation of one-time permutation phase; (4) decision and keys releasing phase.

In phase 1, Bob initiates the authentication process by providing a fresh biometric sample, and then it adopts the same steps that were used at the enrolment stage to generate the cancellable biometrics $CBio'_b$, i.e. using user-based orthonormal random projection. Phase 2 establishes the mutual OTRS between PKG and Bob. The phase mainly relies on run a challenge-response game and is the backbone of the authentication stage. For that reason, a Diffie-Hellman Key Exchange is used to ensure that the outputs of this phase are protected from any manipulation that may happen by a man-in-the-middle. Phase 3 generates a one-time permutation that is implemented by shuffling $CBio'_b$ to add further protection. Both Bob's ID and the shuffled $CBio'_b$ — the two elements that Bob have to send to PKG in order to run the subsequent phase. The last phase is the decision phase where a decision is made whether or not to grant the corresponding decryption key. If the outcome of the decision phase is successful, it means that the user that the server is dealing with is genuine, otherwise, the user is an impersonator/ non-genuine user. Figure 4.5 shows the steps in the authentication phase within the proposed OTCR-MFA.

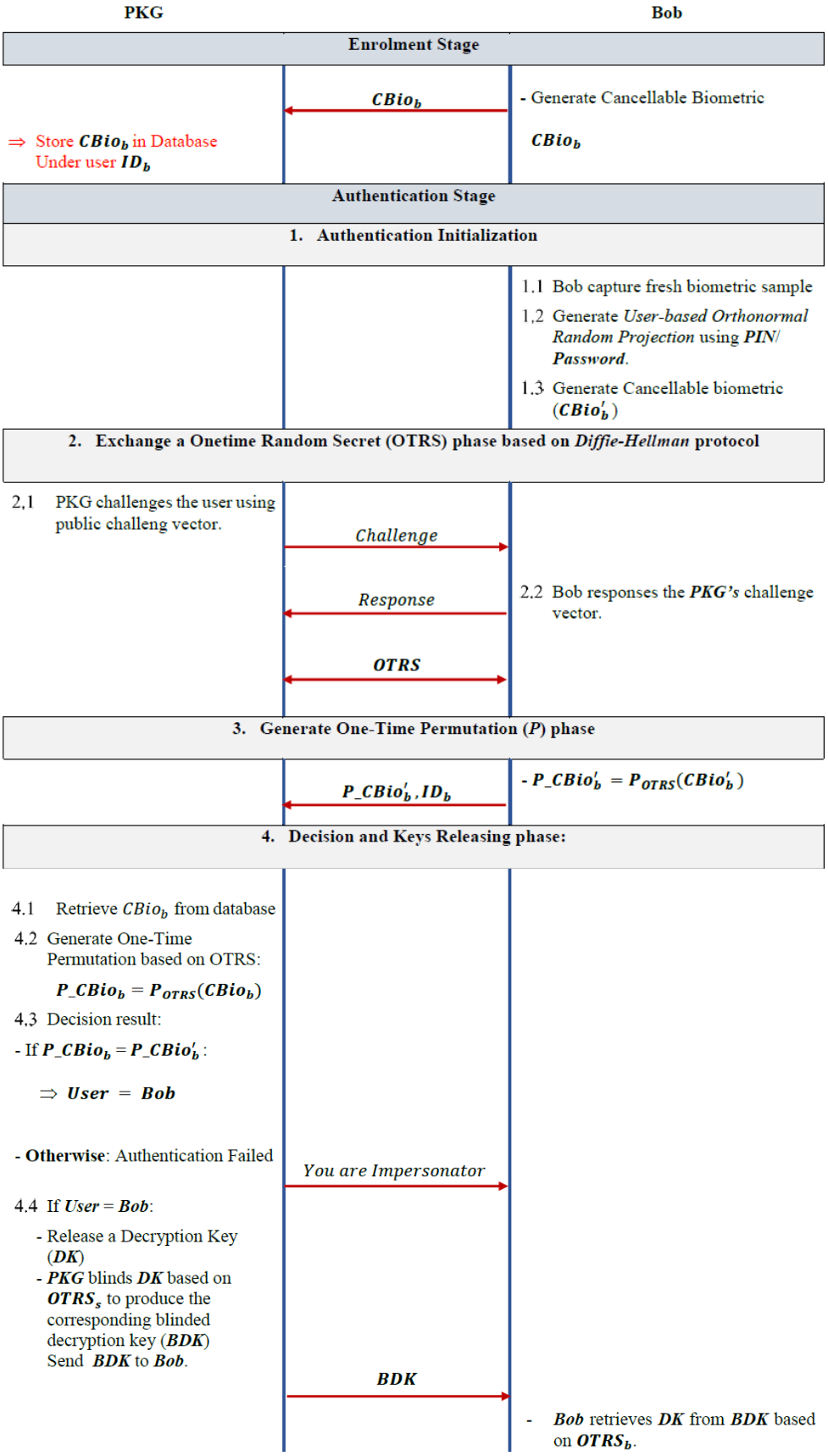


Figure 4.5: Main steps of an authentication stage within the proposed OTCR-MFA.

4.4 The Proposed Solution - Algorithms and Implementation Details

This section describes the fundamental algorithms that contribute to forming OTCR-MFA. The proposed solution is built on the standard F-IBC introduced by [17].

Let the symmetric cyclic group $\mathbb{G}_0$ of prime order p, $\langle g \rangle$ represents the generator of $\mathbb{G}_0$, and a security parameter l which identifies the groups' size, are core elements of a bilinear map function $\hat{e}: \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$. Also, the Lagrange coefficient $\Delta_{i,S}$ is defined for each $i \in \mathbb{Z}_p^*$, and a set $S \in \mathbb{Z}_p^*$ such that:

$$\Delta_{i,S}(f) = \frac{f-j}{i-j}.$$

Assume Alice is a data owner, and she wants to send an encrypted message to Bob using F-IBC scheme. The following sub-sections present the details of the stages (including the main algorithms) of the proposed solution.

4.4.1 Setup and Enrolment:

This algorithm is composed of two stages: Setup and Enrolment stages.

1. **Setup Stage (n, d):** It runs by the PKG, and it takes a length of user's identity, *n*, and an agreed threshold value, *d*, as inputs. The main task of this algorithm is to create a set of fundamental parameters: Master public parameters (MPPs), and Master secret parameters (MSPs). The main components of MPPs include the following:

   $$\{\{\mathrm{T}_i = g^{z_i}\}, Y = \hat{e}(g,g)^k, H, H_1, H_2\},$$

   where z_i is a set of random uniformly elements selected from $\mathbb{Z}_p$ for $i = (1, \dots, |u|)$ and $k \xleftarrow{R} \mathbb{Z}_p^*$, $H: \{0,1\}^* \rightarrow \{0,1\}^n$, $H_1: \{\mathbb{G}_0\} \rightarrow \mathbb{Z}_p^*, and\ H_2: \{\mathbb{G}_T\} \rightarrow \mathbb{Z}_p^*$. Both $\{z_i\}$ and *k* represent the MSPs' components.

2. **Enrolment Stage:** This is the user registration phase in OTCR-MFA. It involves a series of steps outlined in Figure 4.5. Bob presents his biometric sample, e.g. face image, and certain features are extracted to produce a features vector ($\vec{\mathrm{w}}_\mathrm{b}$) of length *n*. Next, an Orthonormal Random Projection (ORP_b) based on user PIN/Password is applied over the features vector ($\mathrm{CBio}_\mathrm{b} = (\mathrm{ORP}_\mathrm{b} \times \vec{\mathrm{w}}_\mathrm{b})$, the output (cancellable biometric) of which is sent and stored in the database of PKG under Bob's unique

identity (ID_b). Note that, ORP_b can be stored in Bob's token/ any other digital storage media. The enrolment stage outputs $CBio_b$ and ID_b.

4.4.2 ENCRYPTION ALGORITHM (*MPPs, M,* $\vec{w}_b'$): This algorithm is executed by *Alice* to produce the ciphertext that is sent to Bob. ***Bob's*** public identity $\vec{w}_b$ is also included in the ciphertext. The resultant ciphertext consists of the following components:

$$CT = (\vec{w}_b', E' = MY^t, \{E_i = T_i^t\}_{i \in \vec{w}_b'}, \text{where } t \text{ is a random element in } \mathbb{Z}_p.$$

4.4.3 AUTHENTICATION STAGE ($MPPs, ID_b, CBio_b$): This stage is directly related to the proposed OTCR-MFA, and it has four main phases, as shown in Figure 4.5.

1. *Authentication Initialization Phase*: The phase has the following processes:

1.1. Bob captures his fresh biometric sample to produce the corresponding features vector ($\vec{w}_{b'}$) using the same features extractor used in the enrolment stage.

1.2. Cancellable biometric ($CBio_b'$) is generated by using ORP stored on Bob's token such that $CBio_b' = (ORP \times \vec{w}_{b'})$.

2. *Exchange a One-Time Secret (OTRS) Phase*: This phase depends on the Diffie-Hellman Key Exchange protocol to produce the mutual One-Time Secret (OTRS). The process runs between the PKG and Bob in two important steps: Challenge and Response steps.

2.1. *Challenge step*: This step is implemented by the PKG server to produce the challenge message (*CM*). The step includes the following processes:

- Select at random $\alpha_i \xleftarrow{R} \mathbb{Z}_P^* \ \forall\ i$ in *n*.
- Calculate the challenge of the PKG: $Cs = g^{\alpha_i} \in \mathbb{G}_0$.
- Lastly, a challenge message (*CM*) is produced (as shown below) to be transferred to Bob.

$$CM = \{Cs, sk_{pkg}(H(Cs)), pk_{pkg-cert.}\},$$

where sk_{pkg} is the private key of PKG, and $pk_{pkg-cert}$ refers to the digital certificate of the PKG's public key.

2.2. *Response Step:* Bob is responsible for implementing conducting this step. Bob needs firstly to guarantee that *CM* has not tampered with, then he will perform the following operations:

- Select at random $\beta_i \xleftarrow{R} \mathbb{Z}_P^* \ \forall\ i$ in *n*.
- Calculate the response of Bob: $Rb = g^{\beta_i} \in \mathbb{G}_0$.

- Bob generates a One-Time Secret ($OTRS_b$) as follows:

$$X_b = Cs^{\beta_i} = g^{\alpha_i \beta_i} \in \mathbb{G}_T$$

$$OTRS_b = H_2(X_b)$$

- To complete the OTRS creation process, Bob sends a response message (RM) to the PKG, which consists of the following components:

$$RM = pk_{pkg}\{Rb, H(Rb)\},$$

where pk_{pkg} indicates to the public key of the PKG server.

On the PKG side, when *RM* is received and ensure its integrity, PKG generates a One-Time Random Secret ($OTRS_s$) as follows:

$$X_s = Rb^{\alpha_i} = g^{\alpha_i \beta_i} \in \mathbb{G}_T$$

$$OTRS_s = H_2(X_s)$$

Now both Bob and the PKG own the OTRS.

3. *Generate a One-Time Permutation (P) Phase*: For further security, Bob shuffles the $CBio'_b$ relying on $OTRS_b$ to produce $P\_CBio'_b$ such that:

$$P\_CBio'_b = OTRS_b(CBio'_b)$$

4. *Decision and Keys Releasing Phase*: This is the most crucial phase of user authentication. The phase is executed by PKG to determine whether Bob can receive his decryption key or not. It begins by sending Bob $P\_CBio'_b$ and ID_b to PKG. The following bullet points explain the main steps of this phase:

4.1. PKG retrieves $CBio_b$ from the database.

4.2. Shuffle the $CBio_b$ based on $OTRS_s$ such that $P\_CBio_b = OTRS_s(CBio_b)$.

4.3. As long as $P\_CBio_b = P\_CBio'_b$, it means the user corresponding with PKG is Bob; otherwise, the user is an impersonator/ spoofed user, not Bob.

4.4.4 Decryption Keys Extraction Algorithm ($MSPs, MPPs, n, \vec{w}_b, d$)

: It is in charge of generating the decryption key. It takes the parameters generated by PKG, the user's features vector and its length as well as the value of the agreed threshold, as inputs. It yields the following decryption key: $Dk_i = \{ g^{q(i)/z_i} \}_{i \in \vec{w}_b}$, where

q is a random polynomial equation of degree (d-1), with a compulsory requirement that $q(0) = k$. Note that, d refers to the prearranged threshold value (d) -1) chosen by PKG.

In the event that the user is Bob, and for further protection, the PKG blinds the decryption key (Dk_i) resulted from decryption keys extraction algorithm using the following formula:

$$BDK = Dk_i.Q, \text{ where } Q = e(g,g)^{OTRS_s}.$$

PKG submits BDK to Bob.

4.4.5 DECRYPTION ALGORITHM ($MPPs$, CT, $\vec{w}_b$, BDK): This algorithm is executed by Bob. Assuming that $P_s = P_b$, this algorithm begins with presenting Bob his biometric features vector $\vec{w}_b$ resulted from his fresh face image. To release the original message, both $\vec{w}'_b$ and $\vec{w}_b$ must be close enough (i.e., $\vec{w}'_b \cap \vec{w}_b \geq d$). Besides, Bob needs to calculate $Q = e(g,g)^{-OTRSS_b}$ to reconstruct the decryption key (Dk_i) from BDK. To decrypt the encrypted message, Bob needs to use the following expression:

$$E'/\prod_{i\in S}(e(\mathcal{R}, E_i))^{\Delta_{i,S}(0)}, \text{ where } \mathcal{R} = BDK.Q^{-1}$$

4.4.5.1 THE CORRECTNESS OF THE PROPOSAL. To ensure that the original message can be released based on the decryption formula if and only if $|\vec{w}_b \cap \vec{w}'_b| \geq d$:

$$E'/\prod\nolimits_{i\in S}(e(\mathcal{R}, E_i))^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}(e((BKD.Q^{-1}), E_i))^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}(e((Q.Dk_i.Q^{-1}), E_i))^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}(e(Dk_i, E_i))^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}\hat{e}(g^{q(i)/z_i}, g^{tz_i})^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}\hat{e}(g,g)^{tq(i)})^{\Delta_{i,S}(0)}$$
$$= M.\hat{e}(g,g)^{tk}/\prod\nolimits_{i\in S}\hat{e}(g,g)^{tq(i)})^{\Delta_{i,S}(0)}$$

$$= M.\hat{e}(g,g)^{tk}/\hat{e}(g,g)^{tk}$$

$$=M$$

The subsequent paragraphs are devoted to describing the techniques used to implement the proposed solution and then discuss the experimental results.

4.4.6 Multi-Factor Cancellable Face Recognition

The concept of biometric revocability/cancellability was introduced as a means to preserve the security and privacy of biometric templates in biometrics-based authentication systems [75]. The biometric revocability has typically two main categories [76]: (1) feature transformations, and (2) biometric cryptosystems. The first category is produced by choosing a particular function-based key to transform the original biometric template into another one but in a secure domain. The matching procedure is then carried out on the transformed template rather than the original. In the verification phase, the same function that was used on the biometric template is also implemented with the fresh biometric data. In the biometric cryptosystems category, the biometric data and cryptography are merged to produce a biometric-based key or hash to be used as a means of biometric revocability [74]. In our cancellable biometrics, the feature transformations (see section 4.3) is adopted to protect the original biometric data of Bob. The Orthonormal random projection (ORP_b) is represented as 2D of 128 × 128 features. Furthermore, in the proposed OTCR-MFA the experiments were carried out over Cambridge Olivetti Research Lab (ORL) face database[77]. The ORL database consists of 400 face images of 40 persons. For each person, there are ten images captured in different conditions. A sample of set images in the ORL database are shown in Figure 4.6.

Additionally, as a proof-of-concept, a Discrete Wavelet Transform (DWT) method is used as a features extraction technique to produce feature vectors FVs. It is important to note that there are no restrictions on using any other advanced technology other than DWT. In order to generate fixed FVs, a low-pass subband LL3 is applied as a third level of resolution approximation of the face image data to produce 168 features for each image. Mean and standard deviation functions were applied to produce unique feature vectors. Furthermore, each person's face images set is split into two sets: the first set consists of three images chosen as the training set, while the second set consists of the remaining seven images chosen as the testing set.

Figure 4.6: Sample of 10 face images in the Olivetti Research Lab (ORL) face database

On the other hand, to calculate the performance and accuracy of OTCR-MFA, a simple Nearest Neighbour (1-NN) Classifier has used to find the distance to the nearest training case. Euclidean distance is the method used to measure the score of convergence in the matching process which is an important issue in pattern recognition [78].

Also, the implementation of the permutation upon the cancellable biometrics supports the users' privacy. However, the permutation applied in the proposed solution is an update of the proposed shuffling introduced by [79]. The figure below illustrates how the features vector is shuffled based on OTRS.

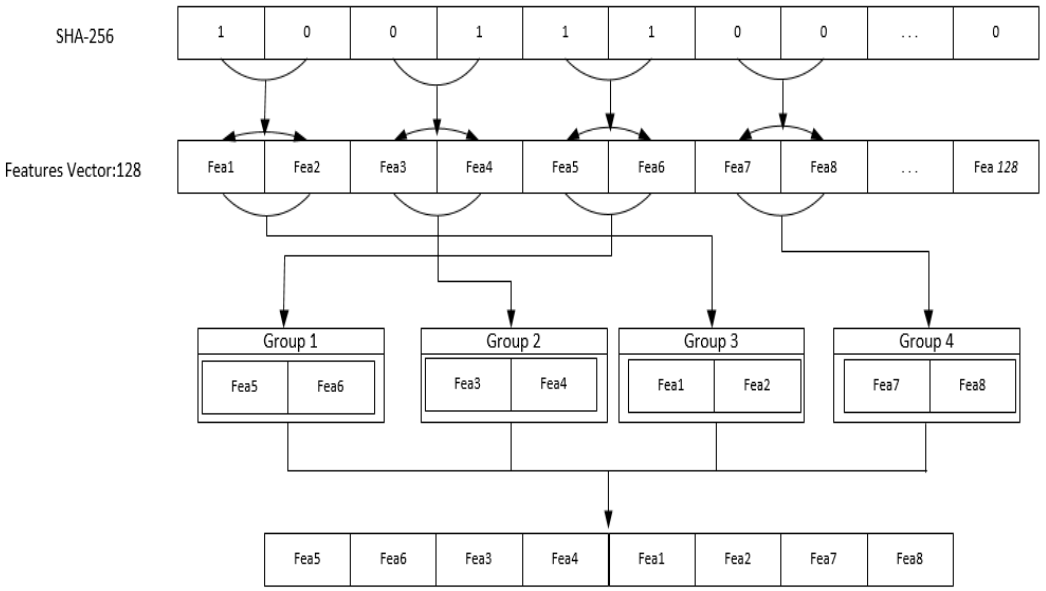


Figure 4.7: Permutation- based on OTRS "adopted in [79]"

4.4.7 Putting Everything Together

The proposed solution allows Bob/ genuine user to get his decryption key if he succeeds in the challenge-response game that OTCR-MFA is based on; hence, preventing imposters from gaining the decryption keys and consequently decrypting the ciphertexts. Therefore, the solution enhances the use of F-IBCs by DOs by protecting the existing F-IBCs from impersonators in addition to preserving users' security and privacy.

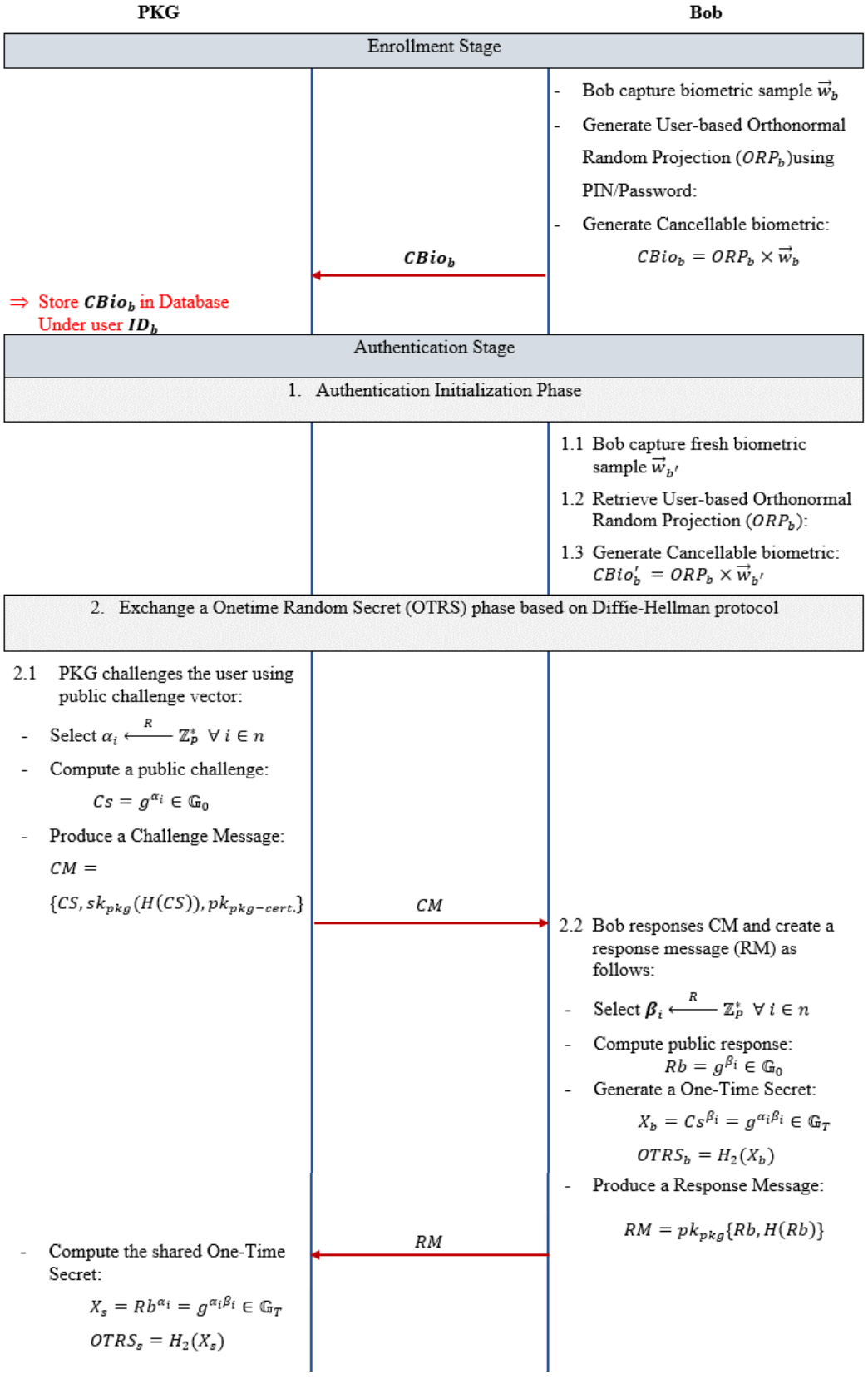


Figure 4.8: The enrolment as well as phases (1&2) of One-Time Challenge-Response Multifactor Biometric Authentication (OTCR-MFA)

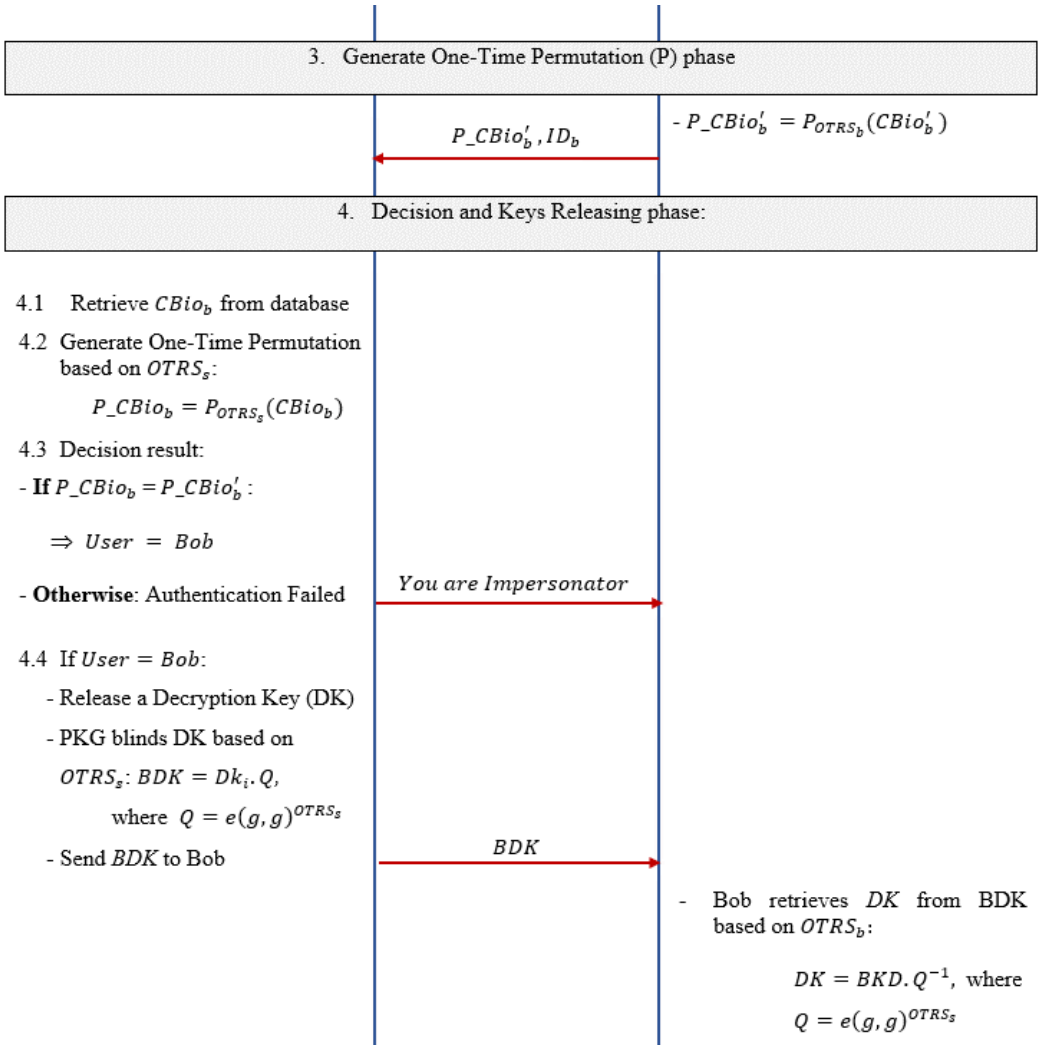


Figure 4.9: Phases (3 & 4) of One-Time Challenge-Response Multifactor Biometric Authentication (OTCR-MFA)

4.5 Security Analysis of the Proposed Solution

This section analyses addressing the vulnerability (security) of the user's authentication associated with F-IBCs.

The security analysis is carried out from two angles:

- Evaluating the proposal against the Fuzzy Selective-ID (F-SID) attack model [55].
- Analysing the performance of the biometric element under different scenarios -- to address questions such as can an imposter fool the system if they get access to the transformation key and/or the permutation key?

4.5.1 Fuzzy Selective ID Attack

F-SID attack model is derived from a more generic model called the "selective ID attack model" proposed in [80] to assist in evaluating the security of standard IBC [81]. The selective ID model provides adversaries with the ability to select an arbitrary identity *ID** to be challenged upon in addition to giving them access to a number of other IDs with their decryption keys as explained in Figure 4.10.

The F-SID can be seen as a challenge game between two parties, a challenger (usually PKG) and an adversary. The security level of any F-SID scheme depends on measuring the advantage of the adversary to win the following challenge game.

(1) An identity ID^* is chosen by the adversary to be challenged upon.

(2) The adversary receives the MPPs generated by the challenger.

(3) The adversary runs private key queries over other identities, ID_i, which has an overlapping less than the pre-agreed threshold value, d, ($|ID_i \cap ID^*| < d$).

(4) The adversary will send two challenge messages M_0 and M_1 of equal length ($|\mathrm{M}_0|=|\mathrm{M}_1|$) to the challenger.

(5) The challenger flips a binary coin, $b \xleftarrow{R} \{0,1\}$, and executes the encryption algorithm on M_b as follows:

   a. $CT \leftarrow Enc(MPPs, ID^*, M_b)$

(6) Repeat step 3 until the adversary decides to interrupt it.

(7) Lastly, the adversary yields a guess b' of b, and be the winner if $b' = b$.

It is essential to highlight that the advantage of the adversary in the game can be measured as $|\Pr[\mathrm{b}' = \mathrm{b}] - 1/2|$.

For that, F-IBC scheme considers secure if there is no polynomial-time adversary can win the above game with non-negligible advantage[17] i.e. it is secure if $\Pr[\mathrm{b}' = \mathrm{b}] \leq \frac{1}{2} + neg(adv)$. Refer to Chapter 3 for more details.

It has been shown that the F-SID depends on reducing (mapping) the F-IBC's security to a Decisional Modified Bilinear Diffie-Hellman hardness assumption over the discrete

logarithms in cyclic groups used in F-IBCs, which was used as proof that there is no polynomial adversary can win Selective-ID with non-negligible advantage.

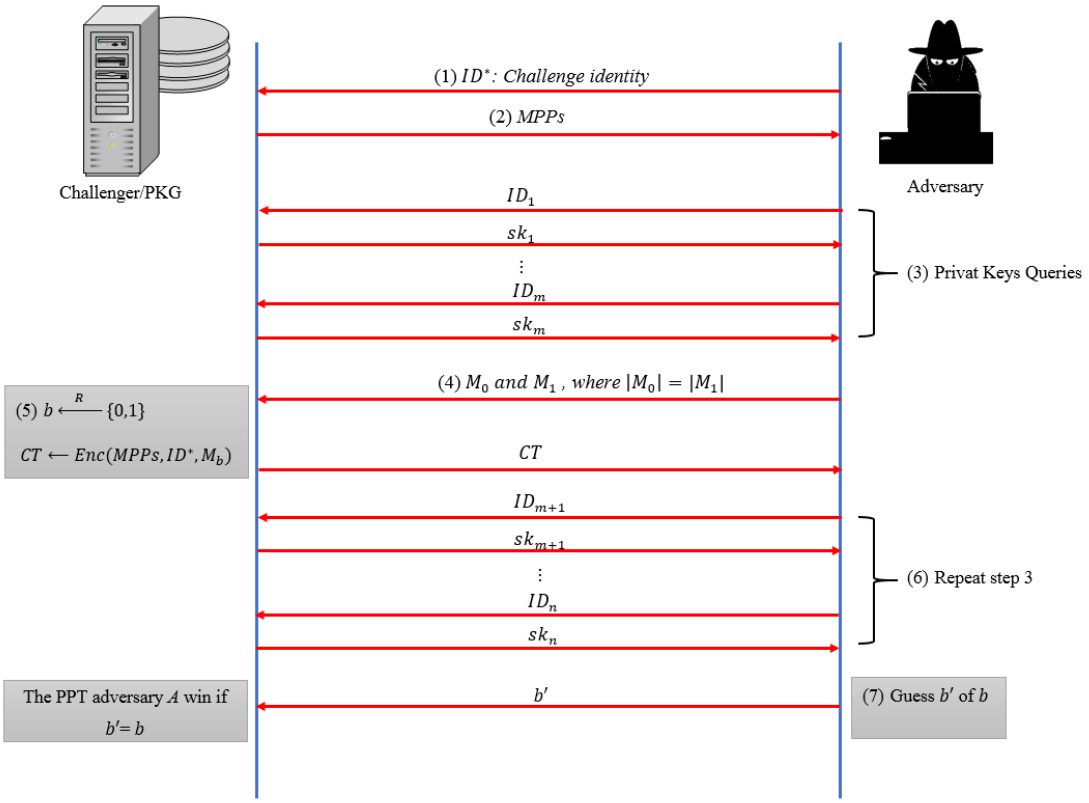


Figure 4.10: The main steps of the Fuzzy Selective Identity attack model

4.5.2 Evaluating the Multi-Factor Biometric Authentication

As it has been seen in the previous sections, F-IBCs have a significant problem-related to user authentication in the keys delivery stage. As a result, we found it essential to find the appropriate mechanism to address this problem. This chapter designed to overcome the downside of the user's verification by preventing impersonators/ non-genuine user from taking advantage of this problem; thus accessing to decrypting the encrypted message.

Initially, OTCR-MFA derived from the proposed work introduced by [82] in which the enrolment stage depends on storing the users' original biometrics in the server database. This idea has another potential security problem concerning user privacy. It allows a dishonest server to track the user. Besides, any breach by adversaries will lead to a failure of the biometric-based authentication system. In OTCR-MFA, version of the biometric

sample that is stored in the server is cancellable biometrics (*CBio*) not the original ones. Thus, it prevents the two challenges mentioned above. *CBio* can be removed instantly and use another version once an adversary detects it. Also, the nature of *CBio* allows the user to handle a number of servers using different versions of *CBio*. It is important to note that the strength of OTCR-MFA stems from the reliance on the One-Time random shared secret (OTRS). Because of the One-Time property, it means OTRS secures against a replay attack which deems one of the lower layer version of the man-in-the-middle attack. Diffie-Hellman Key exchange (DHKE) protocol is adopted to perform the game of challenge-response between the PKG server and the user. OTRS is therefore exposed to the discrete logarithm problem (DLP). The primary purpose behind using the game is to transport OTRS between the two parties above. To protect OTRS from any potential tampering by the man-in-the-middle (as explained in GERT-EU security whitepaper [83]), a cryptographic hash function is utilised.

OTRS also considers the core element for giving the biometrics templates more protection. OTRS is utilized to shuffle the features of *CBio* in the permutation operation. The permutated *CBio* sends with the user's ID to PKG in decision phase to determine whether the user is genuine. It is important to highlight that there is no benefit of intercepting the permutated *CBio* by the man-in-the-middle attack. The reason for this is when the attacker wants to authenticate; the OTRS used will be different from those adopted by the genuine user. On the other hand, for more protection for the decryption key, it is blinded by a bilinear map-based on OTRS. It consequently prevents the impersonators from benefiting from detecting the blinded decryption key even if *CBio* and *P* are detected.

On the other hand, a simple Nearest Neighbour (1-NN) Classifier is also utilized to measure the performance and accuracy of the proposed solution. A new sample is classified by finding the distance to the nearest training case. The 1-NN classifier depends on Euclidean distance to measure scores of convergence in the matching process. Ultimately, the accuracy of OTCR-MFA concerning False Accept Rate (FAR)—which measures the probability of accepting an unauthorized individual—and False Reject Rate (FRR)—which measures the probability of rejecting an authorized individual[84]—can be observed in Figure 4.11with six potential scenarios. It can be seen that the idealistic performance when Equal Error Rate (ERR) is zero can be accomplished in scenarios d and e. Nevertheless, the first two scenarios (i.e., a and b) are similar due to compromising the ORP's PIN/Password and Permutation makes OTCR-MFA will depend on the

threshold of face recognition only. For that, we select to maintain only the biometrics' operating threshold to give the key factor of the proposed OTCR-MFA. The last scenario represents the case of compromising the face biometric by an impersonator while keeping the ORP and Permutation. It means the impersonator will use different PIN/ Password for cancellable biometric and different OTRS for Permutation.

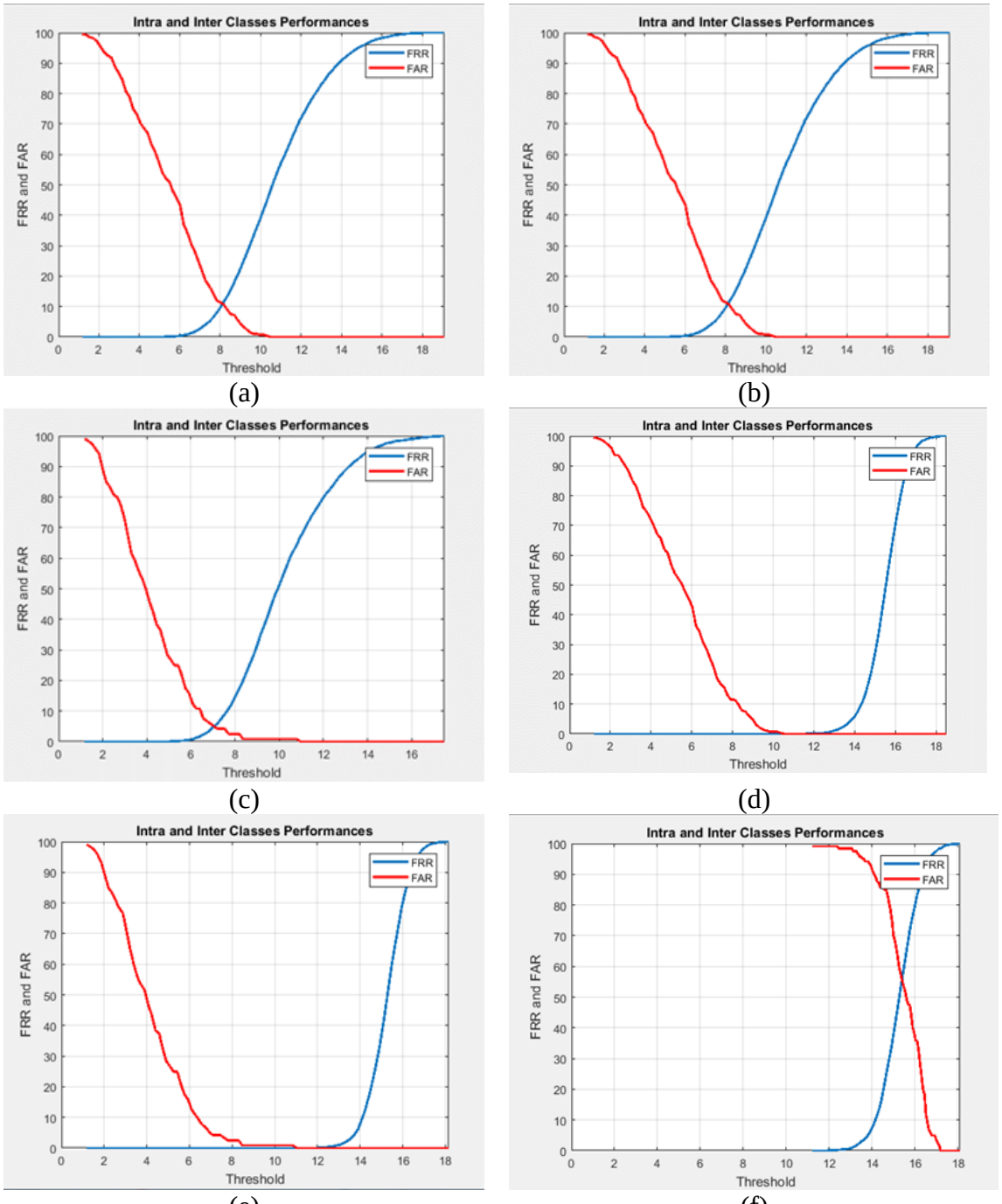


Figure 4.11: OTCR-MFA accuracy in terms of FRR and FAR in six scenarios: (a) Secure face biometric only, (b) Secure face biometric plus compromised ORP & Permutation, (c) Face biometric & Permutation are secured and compromised ORP, (d) Face biometric & ORP are secured and compromised Permutation, (e) All OTCR-MFA factors are secured, (f) ORP & Permutation are secured while face biometric is compromised

4.6 Chapter Summary

Although F-IBCs, as a new model of public key encryption, was invented in order to overcome the complications of using traditional public key encryption, this Chapter highlighted the security vulnerability of the model related to user verification and proceeded to present a solution to address the vulnerability.

The chapter argued that all the existing F-IBCs do not have an efficient and secure mechanism that controls the process of handing the decryption keys to the users. Therefore, we proposed a one-time multi-factor for mutual authentication between PKG and F-IBCs users at the stage of handing the decryption keys. The proposed solution appropriately blends biometrics with other authentication factors in order to provide additional security. OTCR-MFA was suggested to avoid an impersonator attack / non-genuine user from fooling the PKG to obtain the decryption key of authorised users using available biometrics. The OTCR-MFA depends on a mutual challenge-response game that adopts the Diffie-Hellman key exchange protocol to prevent replay and man-in-the-middle attacks.

The study also provided information about the implementation and simulation outcomes to illustrate the feasibility and viability of the solution.

As a users' privacy-preserving layer, the proposed solution utilises cancellable biometric templates. The experiment was carried out on the ORL face database and, as proof-of-concept, a Discrete Wavelet Transform (DWT) was adopted to address the issue of features extraction. To sum up, this Chapter presented a more secure version of F-IBCs with a robust user authentication and non-repudiation characteristics, which prevent the decryption keys from being released to untrusted users.

Chapter 5

Improving Key Generation and Revocation in Fuzzy Identity-based Cryptosystems

In Chapter 4, we proposed a multi-factor biometric mechanism to withstand the security vulnerability surrounding users' verification. However, existing F-IBCs have two further vulnerabilities which are:

1. Decryption keys are fully managed and controlled by PKG i.e. Data Owners (DOs) have no say in the generation, storage and distribution of their decryption keys. Furthermore, storing all decryption keys' components creates a single point of attack, i.e. once the PKG is compromised, all decryption keys are compromised.
2. Existing F-IBC schemes do not provide DOs with explicit key validity/ expiry mechanisms. However, the key revocability gives an admin the possibility of revoking any compromised key at any time as well as providing access to encrypted data for a limited time only.

This chapter builds on the solution proposed in Chapter 4 and proposes two solutions to address the above vulnerabilities.

1. Layer 1: Both PKG and DOs share the process of generating decryption Keys, thereby reducing PKG dominance. There are two components to the decryption key; one is issued by PKG, while DOs release the other portion.

2. Layer 2: Enforcing keys validity (or keys revocation) by DOs to safeguard the encrypted data. It adopts Shamir Secret Sharing to play this role to give DOs reasonable control over their encrypted data.

This chapter shows that the above layers are to support the DOs by including in the process of generating decryption keys and by giving them reasonable control over their

encrypted data stored in the remote site (e.g., cloud environment) via specifying how long their encrypted data can be made available.

The chapter starts by presenting some security challenges related to IBCs and F-IBCs in section 4.1, and then the related works will be discussed in section 5.2. It is then followed by a description of the proposed system and the details of the underlying algorithms in sections 5.3 and 4.4, respectively. Section 4.5 presents the security analysis of the proposals while section 5.6 concludes the research carried out in the chapter.

5.1 Introduction

As explained in Chapter 3 and 4, identity-based cryptosystems (IBCs) and Fuzzy-IBCs (F-IBCs) are an advanced class of public key encryption systems (PKEs) introduced to overcome the key distribution challenge associated with traditional PKEs.

However, existing F-IBCs have vulnerabilities related to the way decryption keys are managed, i.e. generation, storage, and distribution. The decryption keys management is completely controlled by PKG, which is a main concern to the DOs.

Keys revocation is essential to protect the keys from different serious risks, including theft, loss, or even the user no longer being the rightful system user [85]. In this context, it is necessary that decryption keys be subject to revocation and/or replacement. In the previous chapter, it was noted that although F-IBCs was proven to secure against Fuzzy-Selective Identity [59], we showed that there was a security vulnerability in terms of user authentication. In Chapter 4, OTCR-MFA was proposed to address the problem of user authentication.

However, proposing a new solution for user authentication does not mean that F-IBCs are fully secure. The reason is that the existing F-IBCs still faces the problem of the PKG dominance over the management of decryption keys. It was described in chapters 3 and 4 that the PKG is in charge of issuing and controlling two main parameters: MPPs and MSPs. The MSPs are the main component in generating decryption keys. In typical IBCs, only PKG controls the generation of these keys.

Besides, the survival of the decryption keys for indefinite use may also expose them to tampering or collusion. A useful key revocation has been well considered in traditional PKEs setting, but the burdensome management of certificates described in chapter 3 is precisely the dilemma that IBCs seek to improve. The keys have to be subjected to revocability process for many reasons, e.g. the decryption key has been stolen, lost, or even the user is no longer a rightful system user [85]. In these instances, it is crucial to subject the decryption keys to revocation and/or replacement.

The existing chapter is developed to eliminate the above flaws by participating DOs alongside with PKG in the decryption key management process. In the proposed solution, the decryption consists of two parts: one part is generated by the DOs while keeping the other at the disposal of the PKG. Also, the Shamir Secret Sharing solution presented in [57] is exploited to enforce the proposed key validity/revocation by distributing a random

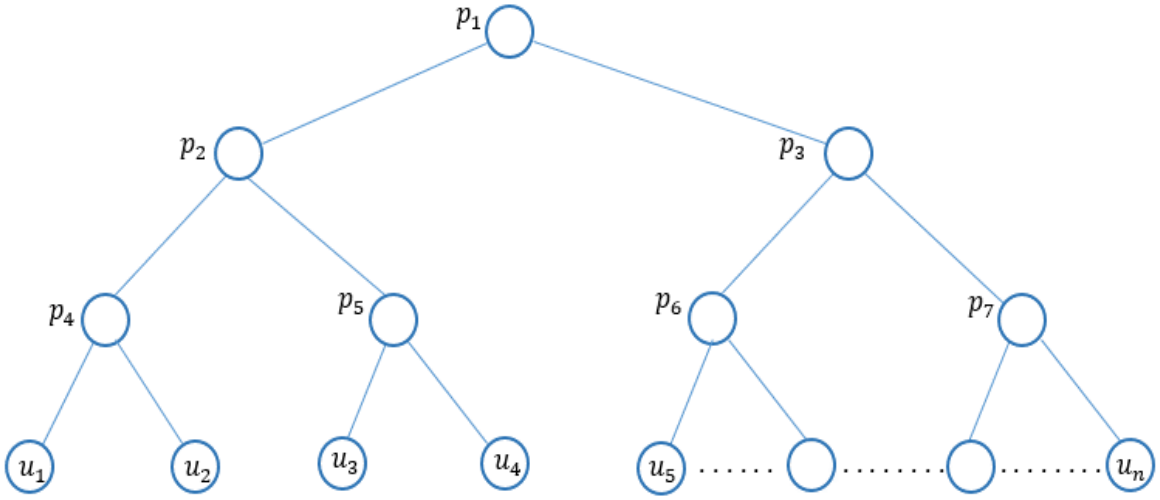
secret value has been chosen by the DOs over the period in which they want the decryption keys to be valid. Making the generating of the decryption keys cooperative between the DOs and the PKG, as well as enforcing key validation features, gives the DOs reasonable control over their encrypted data, which is supposed to be located in a remote site.

5.2 Existing Work on IBC Key Management

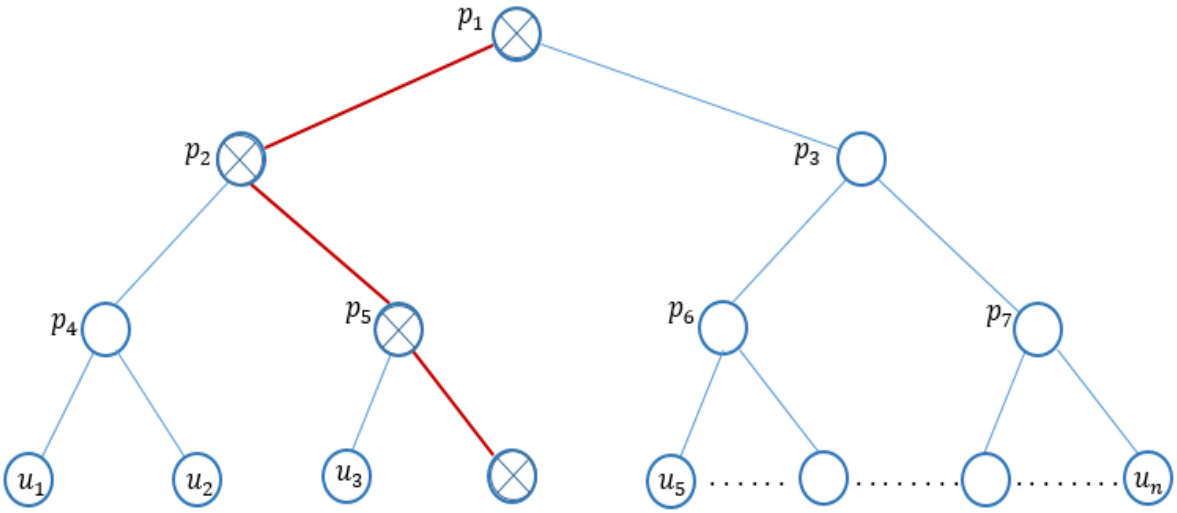
A few studies in the literature focused on revoking keys in IBCs, e.g., [18], [86],[85], [87], [88], [89], [90], [91], [92]. The first key revocation was suggested in standard IBC proposed by Boneh and Franklin by concatenating the current time with the recipient's identity [18]. Alice needs to send her encrypted message that should be available for the current year (only), the public key of Bob will be similar to "Bob@gmail.com||2020". Hence, Bob can use his private key within 2019. But this method would lead to overhead on the PKG side [85], [87], [88]. The reason for this is that all users have to update their private keys periodically, i.e. the user needs— in each time—to interact with the PKG for key updating in addition to verifying their identities. Li et al. pointed that the key revocation suggested by Boneh and Franklin needs the PKG server to be online; thus, it enforces the need to secure and maintain the connection channels, which in turn may result in bottlenecks due to the growing number of users [88]. Therefore, the work in [2] is considered not scalable due to the linearity relationship between the number of users and the occurrence of overhead on the PKG. For that, all the subsequent IBC-related proposals were interested in constructing key revocation that the relationship between the PKG's overhead and the number of users is logarithmic [92].

A binary tree is one of the methods adopted for this purpose in many studies related to the IBC's revocation keys to minimizing the number of key updates in which a trusted authority need to calculate. In [92], a binary tree was used in the F-IBC proposed by [17] to achieve the key revocability. The decryption key issued by PKG is separated into two parts, the private key and key update. The key update is the part responsible for the establishment of the key revocation that will be available to all users. User identity represented as a unique leaf node, and each node will be associated with a random polynomial (p). Therefore, the keys corresponding to the identity are computed based on all the polynomials located on the path, as shown in a pictorial explanation in Figure 5.1. The complexity of key update in terms of PKG and the number of users has reduced from linear to logarithmic.

There have however been a number of defects which can occur using the binary tree; For example, a pair of private keys are needed for every node situated on the route from the leaf to the root. It makes it more complex to issue the appropriate single private key while increasing the number of users. This will also reflect on the amount of the binary tree nodes; thus, the PKG server will face another bottleneck [88].



No user is revoked



User 4 is revoked

Figure 5.1: A pictorial explanation for key revocation using binary tree structure adopted from [92].

Li *et al*. suggested a new scheme to deter collusion in the key revocation process by engaging a cloud service provider (CSP) to determine the revocation for compromised users [88]. PKG and users were locally responsible for implementing certain simple operations. However, the involvement of CSP in this process as a sensitive task has negative aspects because of the return of control to another party (i.e., CSP) rather than

their owners. A public key broadcast encryption (PKBE) system using bilinear maps has been employed by [89] to provide a proper method of IBC's revocation with short private keys and keys update. Two significant schemes have been proposed by [93] and [94] which were combined to produce a new scheme of IBC's revocation in three levelled bilinear maps. Besides, they also proposed another IBC's revocation technique which has short keys and master public parameters by employing the multilinear maps in [93] and [95] schemes. Donghoon et al. suggested another IBC's revocation from codes with rank metric and using the binary tree [90]. To generate decryption keys from public identities in IBC scheme from codes with rank matric, a trapdoor function is suggested to be used which depends on a particular digital signature was proposed in [96]. The binary tree is again used for controlling the PKG's overload for main update computing.

5.3 The Proposed System

This section presents the proposal to address the security limitation highlighted in the previous section. The proposed framework has two main layers. The first layer provides DOs with much more control over the generation and distribution of the decryption keys. The second layer supports the DOs by allowing them to determine the time where the decryption key is valid. Thus, the messages of F-IBC, in our scheme, will encrypt using two main factors— the recipient's identity and the time period.

Furthermore, OTCRMFA has introduced in chapter 4 will exploit in the users' authentication phase—hence, the same scenario was used in Chapter 4 will apply in this chapter. That is, there are three parties: a sender or DO (e.g., Alice), a receptor (e.g., Bob) and a third trusted server (PKG) (see Figure 4.1). Alice wants to send her encrypted message to Bob using the concept of F-IBC proposed by [17].

In general, the proposed framework consists of four main stages, as shown in Figure 5.2.

Stage One: Exchange a One-time Random Secret (OTRS) and enforcing keys validity-based on Shamir Secret Sharing using Diffie-Hellman protocol.

This stage depends on a challenge-response game played between PKG and Alice. They are, in fact, like those used by phase 2 in section 4.4.3. This game aims to establish a mutual OTRS between PKG and Alice basing on the Diffie-Hellman key exchange protocol. The key-validity parameters will be included with the response message submitted by Alice to Bob.

Stage Two: User Authentication-based on OTCR-MFA.

In addition to preventing impersonator attack by OTCR-MFA proposed in Chapter 4, this stage indicates the current time/date that Bob wants to decrypt the encrypted message. It is, therefore, analogous to the time where Bob requests the decryption key.

Stage Three: Decryption Key Generation.

This stage is concerned with issuing decryption keys. To give DOs more power over their encrypted data, which does not exist with current F-IBCs, we propose to generate the decryption key based on a collaboration between PKG and DOs. The decryption key, therefore, is made up of two parts, one issued by PKG, and the other by DOs. It is essential to highlight that both of the decryption key parts based on OTRS issued by stage one.

Stage Four: Make a Decision.

This is the final stage that determines whether Bob can decrypt Alice's encrypted message. If Bob succeeds in testing OTCR-MFA, he will receive the decryption key and move on to the next stage of the verification—which depends on the current time/date; if the time of Bob decryption key's request is within the Alice chosen time, he can read Alice's message otherwise, the process will fail.

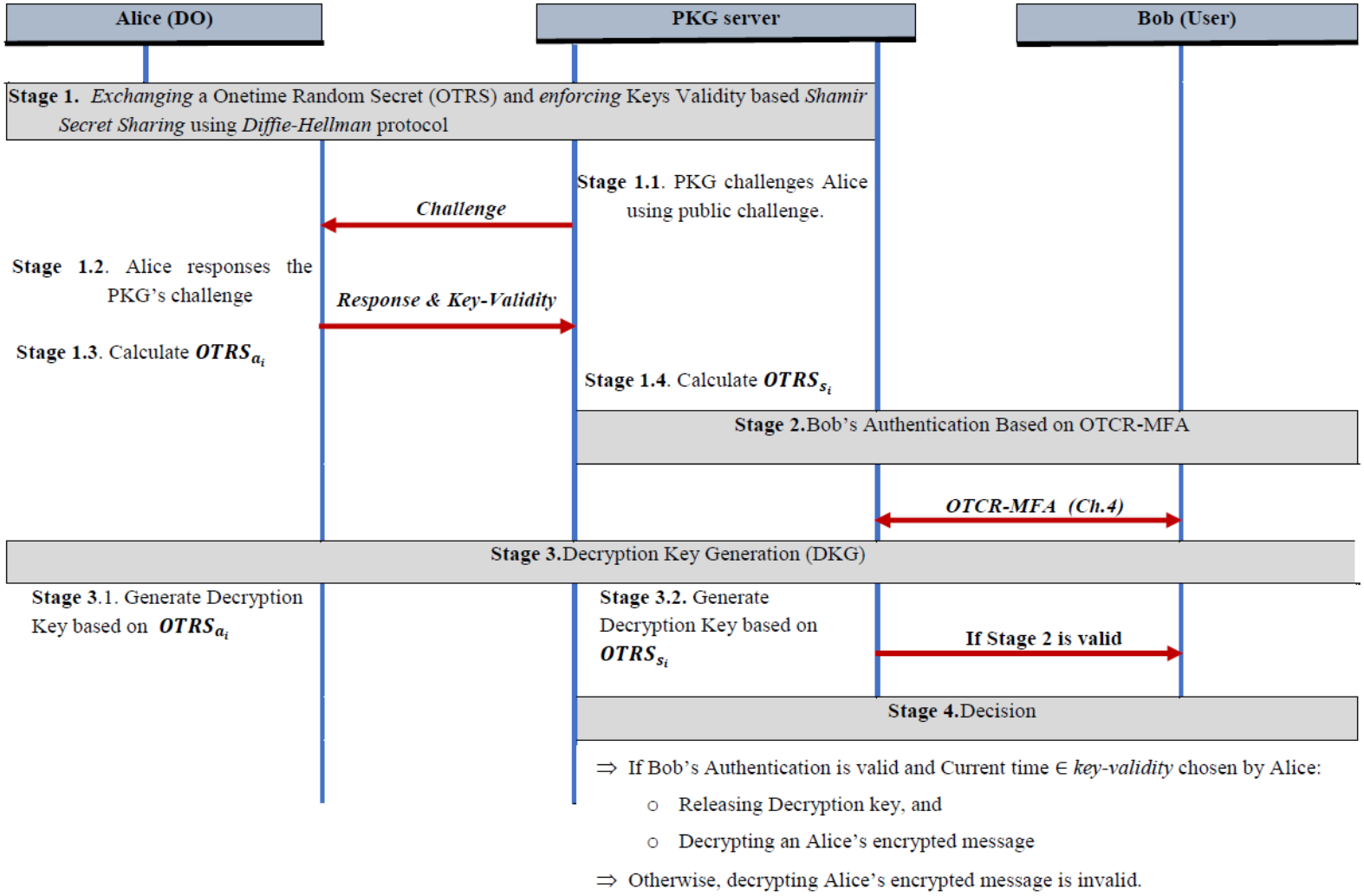


Figure 5.2: An overview of the proposed framework solution

5.4 Algorithms and Implementation Details

This section describes the algorithms of the proposed framework solution. The reset algorithms are similar to those in the proposed OTCR-MAF. However, the proposed solution is applied to the Sahai and Waters scheme[17]. A bilinear map of prime order, *p*, will be considered to build this work. For this, we have $\mathbb{G}_0$ as a bilinear group and $< g >$ refers to the generator of $\mathbb{G}_0$. Therefore, $\hat{e}: \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ is a bilinear map. Also, define the following Lagrange Interpolation $\Delta_{t,sub}$ for each $t \in \mathbb{Z}_P^*$ and a set, *sub*, of an element in $\mathbb{Z}_P^*$:

$$\Delta_{t,sub}(x) = \prod_{j \in sub, j \neq t} \frac{x - j}{t - j}$$

Suppose Alice wants to send an encrypted message to Bob. Also, assume that Bob did all the requirements necessary to implement OTCR-MFA, e.g., enrolment phase. The following subsection describes the algorithms of the proposed framework.

5.4.1 Setup Algorithm (*n*, *d*): This algorithm runs by the PKG to generate the MSP and MPPs. It takes a length of user's identity, *n*, and an agreed threshold value, *d*, as inputs. The setup algorithm follows the same steps involved in section 4.4.1. However, there are two main stages that form the setup algorithm—enrolment and generate F-IBC's parameters. The enrolment phase was introduced to execute the proposed OTCR-MFA and has also been introduced to complement the proposed solution. The outcomes of this algorithm formulate as follows:

- MPPs = $(g, H, H_1, H_2, Y = (g, g)^y)$, where $H: \{0,1\}^* \rightarrow \{0,1\}^n, H_1: \{\mathbb{G}_0\} \rightarrow \mathbb{Z}_p^*, and\ H_2: \{\mathbb{G}_T\} \rightarrow \mathbb{Z}_p^*.$
- MSP = $y \xleftarrow{R} \mathbb{Z}_p^*$.

5.4.2 Proposed Framework: The proposed framework can be divided into four main stages in addition to the F-IBC algorithms. Figure 5.2 depicts these stages, which are described as follows:

Stage 1. Exchanging OTRS and Enforcing Keys-Validity (MPPs, *n*). This stage depends on implementing the challenge-response game in which takes place between

PKG and Alice. This game depends on applying the Diffie-Hellman Key Exchange protocol to share a one-time secret value between Alice and PKG. Moreover, it involves two principal phases that can be identified as follows:

Stage1.1. Challenge (MPPs, *n*). PKG runs this stage to generate the challenge message (CM) as an initial step towards establishing the mutual One-Time Random Secret (OTRS) between PKG and Alice. To do this, the following statements are executed by PKG:

- PKG picks a random set of integers $\alpha_i \in \mathbb{Z}_P^*$, and computes a public challenge vector CV_i to be sent to Alice where

  $CV_i = \{g^{\alpha_i}\} \in \mathbb{G}_0$ for all $i = (1,2,..,n)$, where *n* is the size of the biometric template.

- To protect CV_i from any potential tampering that may occur by *a man-in-the-middle attack*, PKG will use a public cryptographic hash function (H) to produce the following signed challenge message that consists of two main parts:

  $$CM = (CV_i, sk_{pkg}\{H(CV_i)\}, PKG_{pk\_certificate}),$$

  where sk_{pkg} the PKG's private is key and $PKG_{pk\_certificate}$ refers to the PKG's digital public key certificate.

Stage1.2. Response and Assigning an Initial Key-Validity (MPPs, CM, *n*). It runs by Alice/DOs to produce the corresponding response message, which includes the period chosen by Alice for her encrypted message. After receiving *CM*, Alice will check it to make sure it is not tampered with. The responding message (*RM*) is produced based on the following steps:

- Alice picks a random set of integers $\beta_i \in \mathbb{Z}_P^*$, and computes a response vector $RV_i = g^{\beta_i} \in \mathbb{G}_0$ for all $i = (1,2,..,n)$ where n is the size of the biometric template.

- Alice sends RV_i to PKG

Suppose Alice wants to make her encrypted message valid for *k-days* (it could be hours, weeks, and months). To assign the proposed Key-Validity using Shamir Secret Sharing technique, Alice will perform the following actions:

- Compute $T_{m_r} \in \mathbb{Z}_p^*$, where $r \in (1, \dots, k)$.

- Computes the complementary part (CP), which is the core element of applying key revocation that can be imposed by Alice by selecting a random integer $CP \in \mathbb{Z}_p^*$.

  Then, the *key-validity* (T_{m_z}) will consist of the concatenation of T_{m_r} and CP such that: $T_{m_z} = (\{T_{m_r}\}, CP)$.

- To use Shamir Secret Sharing, Alice needs to do the following:

  - Select a random integer $h \in \mathbb{Z}_p^*$.

  - Select a random linear polynomial equation (μ) of the first degree and find $\mu(t)$ for each $t \in T_{m_z}$, with the condition that $\mu(0) = h$—Bob must succeed at least (2, *k*).

- Alice sends the response ciphertext as the response message (RM) encrypted using the public key of PKG (pk_{pkg}). RM consists of two principal parts as shown:

$$RM = (pk_{pkg}\ (H\{RV_i || CP\}))$$

Stage1.3. Generate $OTRS_{a_i} = H_2\{CV_i{}^{\beta_i}\} = H_2\{g^{\alpha_i\beta_i}\} \in \mathbb{Z}_P^*$.

Stage1.4. After PKG receiving the RM and ensuring its integrity, PKG formulates $OTRS_{s_i}$ such that:

$$OTRS_{s_i} = H_2\{RV_i{}^{\alpha_i}\} = H_2\{g^{\alpha_i\beta_i}\} \in \mathbb{Z}_P^*$$

Now, Both Alice and PKG has $OTRS_i = OTRS_{s_i} = OTRS_{a_i}$ as the core element of the proposed framework solution. However, Figure 5.3 shows the steps of exchanging $OTRS_i$ as well as assigning the period time.

We assume that Alice's message (*M*) is encrypted using the encryption algorithm proposed by Sahai and Waters[59].

Stage 2. User Authentication-based on OTCR-MFA ($OTRS_{s_i}$, MPPs, *n*). It follows the steps used in chapter 4. Therefore, in this stage, we will focus on the part related to complete the proposed Key-Validity. Once Bob requests the decryption keys, the current time will be recorded as the time request of the decryption key, which will be checked later to see if it is within the period selected by Alice. For further information about OTCR-MFA, see chapter 4.

Stage 3. Decryption Keys Generation (DKG). This stage explains the second part of the proposed framework regarding the decryption key. It aims to include Alice in the process of generating decryption keys and thus reduce the dominance of PKG in the uniqueness of the implementation of this process and make it a collaborative process. Thus, it gives Alice a reasonable control over her encrypted data stored in a remote location. The resulting decryption key consists of two parts, one issued by PKG and the other by Alice.

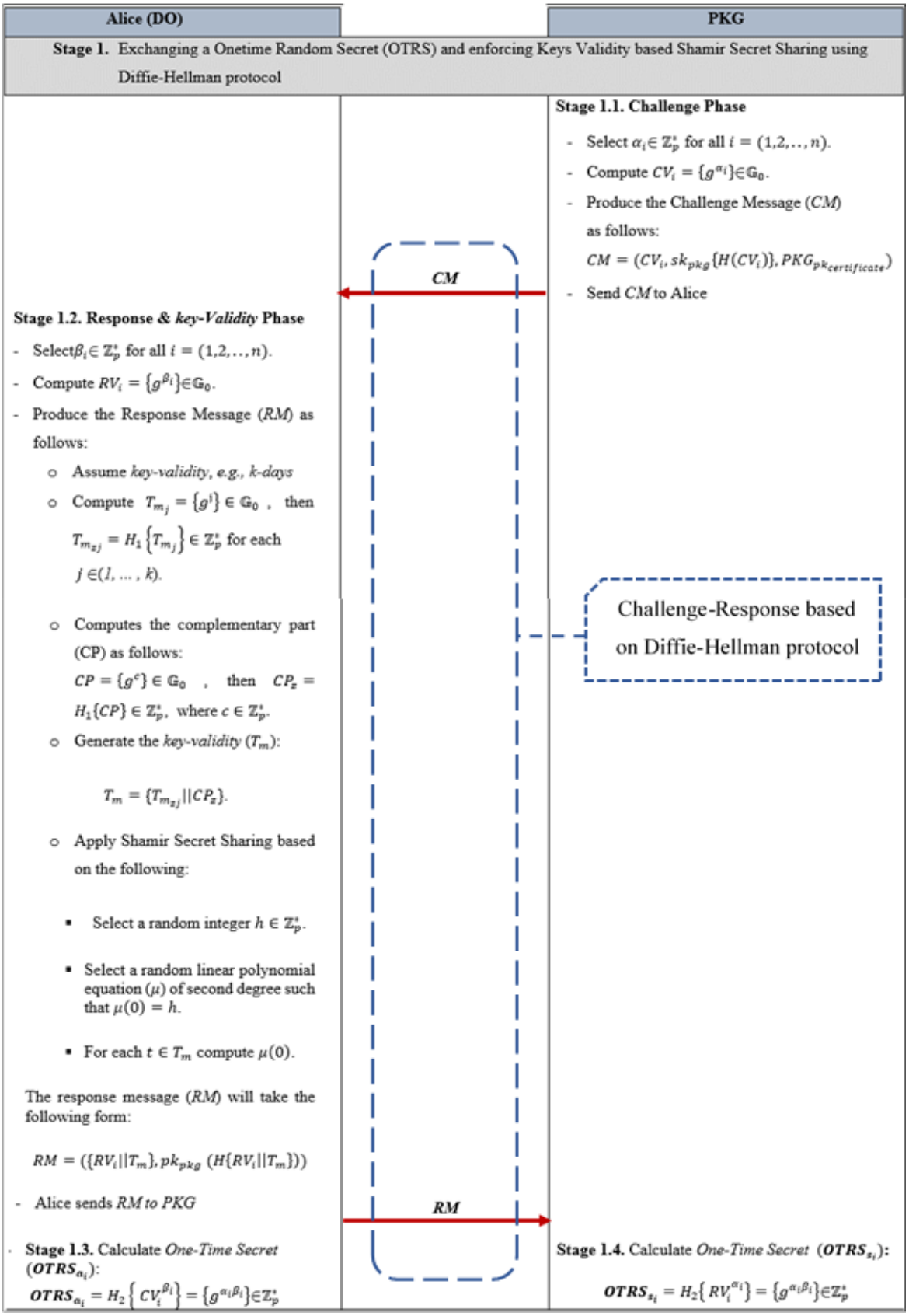


Figure 5.3: The challenge-response game between Alice and PKG to establish the shared $OTRS_i$ and assign the period related Key-Validity

Stage 3.1. DO-based DKG (MPPs, *n*). This stage is carried out by Alice/ DOs to generate the first part decryption key. To do this, Alice needs to conduct the consequent step:

- Select a random integer $s \in \mathbb{Z}_p^*$,

- Compute the first part of the decryption key as follows:

$$D_i' = \{g^{sOTRS_{a_i}}\}_{i \in \vec{w}'},$$

where $\vec{w}'$ represents Bob's public biometric template.

- To assign the Key-Validity, Alice must also compute the following formula to be included in her ciphertext.

$$Q = e(g, g)^h$$

Stage 3.2. PKG-based DKG (MPPs, RM, OTRS_{s_i}, *n*). This stage is implemented by the PKG on Sahai and Waters scheme to generate the second part decryption key. The PKG's decryption key part will be

$$D_i = \{g^{q(i)/OTRS_{s_i}}\}_{i \in \vec{w}} \qquad (5.1)$$

, where q is a polynomial equation of (*d-1*) degree chosen by the PKG.

Suppose that the current time is chosen by Bob to get his decryption key (D_i**)** is $T_{m_b} \in \mathbb{Z}_p^*$. PKG then retrieves the complementary part CP from RM submitted by Alice then computes $T_{m_z'} = \{T_{m_b}, CP\}$. If Bob succeeds in the verification test based on OTCR-MFA by Chapter 4, the following components will then be calculated to be submitted to Bob:

$$DT_b = \{(D_i)_{i \in \vec{w}}, T_{m_z'}\} \qquad (5.2)$$

Additional description in terms of stage 3 can be seen in Figure 5.4**.**

Stage 4. Decision Stage. This stage is responsible for identifying whether Bob can decrypt the Alice's encrypted message or not. The result of this stage will be decided based on the decryption process as will be seen in the next section.

5.4.3 DECRYPTION ALGORITHM ($CT, DT_b, \vec{w}$): This algorithm is executed by *Bob*, it takes the encrypted message (*CT*) produced by *Alice,* the fresh public biometric of Bob ($\vec{w}$) (i.e., fresh face image data), and DT_b generated by PKG. Suppose that the ciphertext (*CT*) created by the encryption algorithm has the following ingredients:

$$CT = (\vec{w}', Q, E', \{D_i'\}_{i\in\vec{w}'}), \text{ where } E' = MY^s.$$

The original message can be released if and only if both $\overrightarrow{w'}$ and $\vec{w}$ (also called "error-tolerance") are close enough, i.e., the agreed threshold value is met, as well as $\left|T'_{m_z} \cap T_{m_z}\right| = 2$. The following expression is used as follows:

$$M = Q.(E'/\prod\nolimits_{i\in S}(e(D_i, D_i'))^{\Delta_{i,S}(0)}).Q'^{-1} \tag{5.3}$$

, where

$$Q' = \prod_{t\in T_{m_z}} \hat{e}(g,g)^{\mu(t)})^{\Delta_{t,T_{m_z}}(0)} \tag{5.4}$$

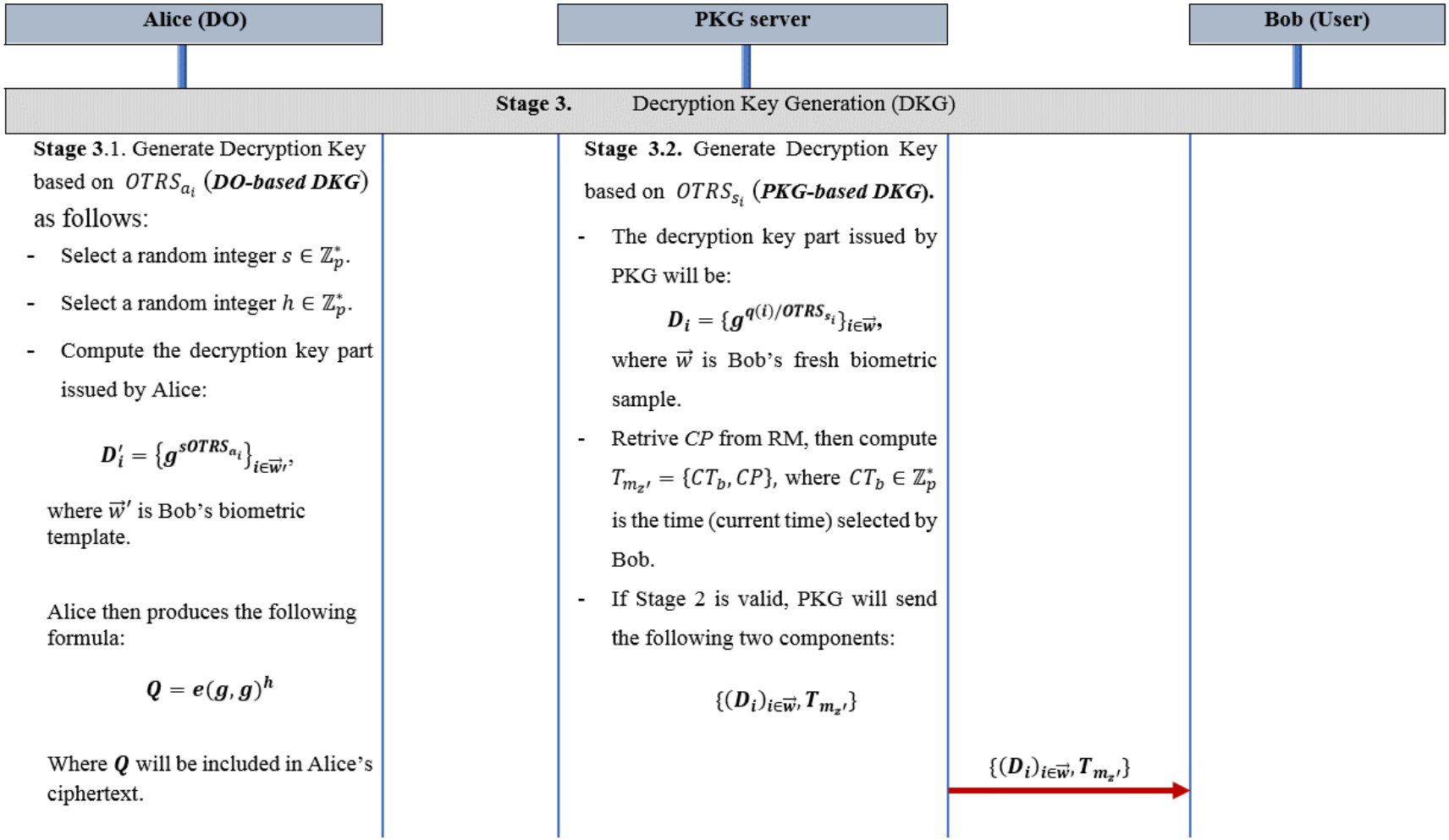


Figure 5.4: The steps to be followed for the production of Stage 3 in the proposed framework

5.5 Security Analysis of the Proposed Solutions

This section demonstrates the security of the proposed framework system that has been applied to the F-IBC proposed by Sahai and Waters in [59].

At the outset, it is vital to guarantee that the equation (5.3) is correct; in other words, it produces Alice's original message. To do that, we have the following correctness form.

Correctness: In order to ensure the original message (M) can be reconstructed based on equation (5.1), it should follow the steps below:

$$Q.(E'/\prod_{i\in S} e(D_i, D_i')^{\Delta_{i,S}(0)}).Q'^{-1} \tag{5.3}$$

$$= e(g,g)^h.(M.\hat{e}(g,g)^{sy}/\prod_{i\in S}\hat{e}\,(g^{\frac{q(i)}{OTRS_{s_i}}}, g^{sOTRS_{a_i}})^{\Delta_{i,S}(0)}).$$

$$(\prod_{t\in T_{m_z}}\hat{e}(g,g)^{\mu(t)})^{\Delta_{t,T_{m_z}}(0)})^{-1}$$

$$= e(g,g)^h.(M.\hat{e}(g,g)^{sy}/\prod_{i\in S}\hat{e}\,(g^{q(i)}, g^{s})^{\Delta_{i,S}(0)}).$$

$$(\prod_{t\in T_{m_z}}\hat{e}(g,g)^{\mu(t)})^{\Delta_{t,T_{m_z}}(0)})^{-1}$$

If $|\vec{w}\cap\vec{w}'|\geq d$, then

$$= e(g,g)^h.(M.\hat{e}(g,g)^{sy}/\hat{e}(g,g)^{sy}).(\prod_{t\in T_{m_z}}\hat{e}(g,g)^{\mu(t)})^{\Delta_{t,T_{m_z}}(0)})^{-1}$$

$$= e(g,g)^h.M.(\prod_{t\in T_{m_z}}\hat{e}(g,g)^{\mu(t)})^{\Delta_{t,T_{m_z}}(0)})^{-1}$$

if $\left|T'_{m_z}\cap T_{m_z}\right| = 2$, then

$$= e(g,g)^h.M.e(g,g)^{-h}$$

$$= M$$

In Chapter 4, user authentication technique was the primary concern to be resolved by the proposed OTCR-MFA. It has been shown that despite the solution proposed in Chapter

4, F-IBC still faces other problems. The security of the proposed framework can typically be categorized into two layers:

- Layer One: Decryption key issuance method
- Layer Two: Concerning the Keys-Validity issue.

5.5.1 The Security of the Decryption Key

Initially, the construction of the decryption key under the proposed solution is primarily based on the One-time Random Secret (OTRS) parameter exchanged between PKG and DOs. To exchange the OTRS, a challenge-response protocol derived from a Diffie-Hellman Key Exchange or Computational Diffie-Hellman (CDH) assumption was employed. Therefore, the security of OTRS does completely rely on the hardness of the discrete logarithm problem of CDH. Below, we prove that the probability of attacker access the OTRS is negligible.

Definition 5.5.1- Discrete Logarithm (DL) problem. Let $\mathbb{G}_o$ be a multiplicative group of prime order p, $\langle g \rangle$ be a generator of $\mathbb{G}_o$ and α be uniformly chosen at random from $\mathbb{Z}_p^*$. Let $X = g^\alpha$, then for all Probabilistic Polynomial-Time (PPT) adversaries A, it is hard to compute α from given X with non-negligible advantage. Hence, we get

$$\Pr[A(\mathbb{G}_0, p, \langle g \rangle, X) = \alpha] \leq negl(adv).$$

Definition 5.5.2- Computational Diffie-Hellman (CDH) assumption. Let $\mathbb{G}_o$ be a multiplicative group of prime order p, $\langle g \rangle$ be a generator of $\mathbb{G}_o$ and select α, β uniformly at random from $\mathbb{Z}_p^*$. Let $X = g^\alpha$ and $Y = g^\beta$, then for all Probabilistic Polynomial-Time (PPT) adversaries A, it is hard to compute $g^{\alpha\beta}$ from given X, Y with non-negligible advantage.

Definition 5.5.3- Decisional Diffie-Hellman (DDH) assumption. Let $\mathbb{G}_o$ be a multiplicative group of prime order p, $\langle g \rangle$ be a generator of $\mathbb{G}_o$ and select α, β uniformly at random from $\mathbb{Z}_p^*$. Let $= g^\alpha \quad Y = g^\beta$, and $Z = g^{\alpha\beta}$ then for all Probabilistic Polynomial-Time (PPT) adversaries A, it is hard to determine whether $Z = g^{\alpha\beta}$ or $Z = g^c$, where c is uniformly chosen at random from $\mathbb{Z}_p^*$.

Because ($OTRS=g^{\alpha\beta}$) is the core element of the two-part decryption key, it is hard to compute the mutual ($OTRS=g^{\alpha\beta}$) by *an attacker* with non-negligible advantage. To prove that, the reduction concept is applied over CDH.

In order to prove that the hardness of CDH problem with respect to the instances $(\mathbb{G}_0, p, \langle g\rangle, g^{\alpha}, g^{\beta})$ imply it faces the hardness of DL problem with respect to the instances$(\mathbb{G}_0, p, \langle g\rangle, g^{\alpha})$, we reduce the CDH to DL (it is written CDH $\leq_{\mathrm{p}}$ DL). To do this, assume that $\mathcal{A}$ is a probabilistic polynomial-time algorithm that outputs $\alpha' \in \mathbb{Z}_p$ on the input $(\mathbb{G}_0, p, \langle g\rangle, g^{\alpha})$. Thus, $\mathcal{A}$ wins the game if $g^{\alpha'} = g^{\alpha}$ due to it implies $\alpha' = \alpha$. In some of the literature, $\mathcal{A}$ is called an oracle/ efficient algorithm.

We assume that $\mathcal{A}$ won the DL game, it will be used to construct the simulator $\mathcal{B}$ which solves the game in CDH. Given instances $(\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta})$ with respect to CDH, $\mathcal{B}$ queries $\mathcal{A}$ on $(\mathbb{G}, p, \langle g\rangle, g^{\alpha})$ and get $\alpha' \in \mathbb{Z}_p$. Next, B calculates $(g^{\beta})^{\alpha'}$, and it will be that $\mathcal{B}$ will win if and only if $\mathcal{A}$ wins such that:

$$(g^{\beta})^{\alpha'} = CDH_g(\, g^{\alpha}, g^{\beta}) \Leftrightarrow \alpha' = \alpha$$

Now, the hardness of CDH in relation to $(\mathbb{G}_0, p, \langle g\rangle, g^{\alpha})$ implies that the success probability for every probabilistic polynomial-time algorithm (especially $\mathcal{B}$) is limited by some negligible advantage ($negl(adv)$).

$$\Pr\left[DL_{\mathcal{A},(\mathbb{G},p,\langle g\rangle,\, g^{\alpha})}(n) = 1\right] = \Pr[\mathcal{B}\left((\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta}\right) = g^{\alpha\beta}\Big] \leq negl(adv).$$

Further security analysis proves that CDH is also harder than DDH using the following facts: assume $\mathcal{A}$ is an arbitrary probabilistic polynomial-time algorithm for CDH which takes $(\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta})$ as inputs to give $\mathcal{Q} \in \mathbb{G}$. $\mathcal{A}$ will win the game if $\mathcal{Q} = CDH_g($ $g^{\alpha}, g^{\beta}) = g^{\alpha\beta}$. Thus, $\mathcal{A}$ will be used to construct the simulator $\mathcal{B}$ to be carried out on DDH using the following: $\mathcal{B}$ will give access to $\mathcal{A}$ and use the DDH instances $(\mathbb{G}, p, \langle g\rangle), g^{\alpha}, \mathcal{Q}')$ such that $\mathcal{Q}' = g^{\alpha\beta}$ or $\mathcal{Q}' = g^{c}$, where c is chosen uniformly at random from $\mathbb{Z}_p^*$. $\mathcal{B}$ queries $\mathcal{A}$ on $(\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta})$ and get $\mathcal{Q}$. The outcomes of $\mathcal{B}$ will be 1 if $\mathcal{Q} = \mathcal{Q}'$ and 0 otherwise. This produced

$$\Pr[\mathcal{B}\left((\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta}, g^{\alpha\beta}\right) = 1\Big] = \Pr[\mathcal{A}\left((\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta}\right) = g^{\alpha\beta}\Big]$$

In contrast,

$$\Pr[\mathcal{B}\left((\mathbb{G}, p, \langle g\rangle, g^{\alpha}, g^{\beta}, g^{c}\right) = 1\Big] = \frac{1}{p}$$

Assuming that DDH is hard relative to $(\mathbb{G}, p, \langle g \rangle), g^{\alpha}, \mathcal{Q}')$ in which will give

$$\Pr[\mathcal{B}\left((\mathbb{G}, p, \langle g \rangle, g^{\alpha}, g^{\beta}, g^{c}\right) = 1\Big] - \Pr[\mathcal{B}\left((\mathbb{G}, p, \langle g \rangle, g^{\alpha}, g^{\beta}, g^{\alpha\beta}\right) = 1\Big] \leq negl(n).$$

By simplicity we get,

$$\frac{1}{p} - \Pr[\mathcal{A}\left((\mathbb{G}, p, \langle g \rangle, g^{\alpha}, g^{\beta}\right) = g^{\alpha\beta}\Big] \leq negl(n)$$

$$\Pr[A\left((\mathbb{G}, p, \langle g \rangle, g^{\alpha}, g^{\beta}\right) = g^{\alpha\beta}\Big] \leq negl(n) + \frac{1}{p}$$

This proves that CDH is hard as long as $\mathcal{A}$ has a negligible advantage to win the game. The hardness of CDH reflects on the OTRS is used to construct the proposed decryption key. Also, using a cryptographic hash function implies preventing eavesdroppers (or man-in-the-middle) from altering the OTRS by providing their $g^{\alpha'}$and $g^{\beta'}$ as indicated in the GERT-EU security whitepaper [83]. Besides, it secures against replay attack in which it is one of the lower-tier versions of the man-in-the-middle attack.

On the other hand, the contribution in generating the decryption key by DOs resists dishonest PKG or the negative consequences of a single point of attacks.

Considering the decryption key issued by the DOs (or Alice) is strong ensures their role in imposing their control over their encrypted data, and this is one of the tasks that this Chapter aimed to reach.

5.5.2 KEY-VALIDITY

In order to present the security of the proposed *key-validity*, assume the following scenario: Alice wants to send an encrypted message to Bob using Sahai and Waters scheme. Alice wants her message to be readable from 10/11/2019 to 15/11/2019, i.e., the message cannot be read before or after the period specified. To apply the Shamir Sharing Secret, a secret value should be chosen by Alice to be distributed over the period she selected. In other words, the secret value $h \in \mathbb{Z}_p^*$ distributes over the days = {10/11/2019, 11/11/2019, 12/11/2019, 13/11/2019, 14/11/2019, 15/11/2019} where each element in days will be chosen from $\mathbb{Z}_p^*$. The distribution process takes place by adopting a polynomial equation μ of first degree with a mandatory requirement $\mu(0) = h$. For each element (x) in days, Alice computes $\mu(x)$ to produce the point (x, $\mu(x)$). Furthermore, Alice picks a one-time random integer ($\in \mathbb{Z}_p^*$) which is the essential factor of assigning the key revocability feature to the proposed key-validity. We call this factor a

complementary part (CP). Alice after that calculates $\mu(CP)$ to produce another point $(CP, \mu(CP))$; hence, h becomes now distributed over 7 elements and generated 7 points.

After Alice sends her encrypted message, Bob begins by authenticating himself utilizing the OTCR-MFA proposed in chapter 4. If Bob succeeds in passing the OTCR-MFA test, the period in which he asked for his decryption key will be recorded as the current day (CD_b) chosen to access and read the encrypted message sent by Alice. By recovering the CP from RM that was sent by Alice, it will be concatenated with the CD_b chosen by Bob; the process will occur by PKG to generate $T_{m_{z'}} = \{CP, CD_b\} \in \mathbb{Z}_p^*$ in order to send to Bob.

The Lagrange Interpolation polynomial is then used in order to reconstruct the secret value h. The Lagrange Interpolation polynomial $\mu(x)$ of degree $\leq (d-1)$, where d is the number of parties which are used in the process of rebuilding the secret value. Thus, in our proposed work, it has been used as a polynomial of the first degree (2-1=1) that passes through the points $(CP, \mu(CP)), (CD_b, \mu(CD_b))$. The formula of the Lagrange Interpolation can be written as follows:

$$\mu(x) = \sum_{j=1}^{d} \mu_j(x) \tag{5.5}$$

, where

$$\mu_j(x) = y_j \prod_{\substack{i=1 \\ i \neq j}}^{d} \frac{x - x_i}{x_j - x_i} \tag{5.6}$$

Releasing the secret value h is *achieved* if and only if *CP* exists as well as $CD_b \in T_{m_z}$. However, due to having only 365 (based on our example) days a year, Bob or any other attacker can easily guess CD_b. Nevertheless, they also need to guess CP, which is a very large one-time random integer, in order to complete the reconstruction of h. Therefore, the proposed key-validity support the DOs' control by preventing any other user from getting access and read the encrypted data unless *CP*. Besides, this scheme prevents any collusion attack that could occur between a dishonest PKG and other user/attacker; since without the second part of the decryption key $\{D_i'\}_{i \in \vec{w}'}$, which is under the control of DOs, they cannot read the encrypted message. As a consequence, for key revocation, only what Alice needs is updates *CP*.

5.6 Chapter Summary

This Chapter builds on the proposal described in Chapter 4 by addressing other vulnerabilities of existing F-IBCs. The key limitation of current F-IBCs lies in the complete dominance of PKG on the process of issuing decryption keys without any contribution of DOs to the process. As an additional security layer, the proposal employed the hardness of CDH problem to prevent any tampering the might affect the core ingredients of the solution.

The chapter presented analyses of the proposed framework in terms of the security from two angles: 1) exploiting the existing hardness with respect to Diffie-Hellman key exchange and 2) using Shamir Secret Sharing for the inclusion of DOs in the process of generating decryption keys and specifying a certain validity period of these keys, respectively. The analysis showed the strength of decryption key's components against adversaries to win CDH with non-negligible advantage. Besides, unless a user has the complementary part (CP) issued by DO, they cannot decrypt the encrypted message; in other words, the proposed key-validity has indeed contributed to supporting DOs' control.

CHAPTER 6

BIOMETRIC CRYPTOSYSTEMS FOR SECURITY IMPROVED FUZZY IDENTITY-BASED CRYPTOSYSTEMS

As stated earlier fuzzy identity-based cryptosystems (F-IBCs) and the standard identity-based cryptosystems (IBCs) in general are mainly based on the use of user identity when issuing asymmetric keys. In existing F-IBCs, two essential elements are involved in creating the decryption keys of users: a master secret parameter (MSP), which is only accessible by PKG servers, and users' unique identities. While the same users' identities are being used in addition to as other public variables, called master public parameters (MPPs), to generate the encryption keys.

The F-IBC solutions proposed in Chapter 4 and Chapter 5 addressed three significant security vulnerabilities related user verification, decryption keys management, and keys revocability. This Chapter proposed the final security layer to address the following F-IBCs' security vulnerabilities:

- PKG is the only party that generate, store, and manage the MSPs. This gives PKG complete control over the encrypted data instead of DOs.
- The MSPs are stored in one place i.e. on the PKG database, which in turn could lead to one of the two security vulnerabilities:
    - Domino's effect resulting from the central point of attack i.e. once the PKG is compromised, all communications are compromised.
    - PKG-based Key Escrow. In current IBC systems, PKG can access or give third party access to the user's key without the user's permission.

The chapter proposes a new solution that enables DOs to take control of their keys and shared encrypted data instead of the PKG server. The solution binds the MSP of a user to their secure biometric so that users are the only party with full control over the management of MSPs and consequently the encryption/decryption keys.

The chapter begins by introducing F-IBCs that include IBCs and how they are compared to traditional public-key encryption systems, particularly in key management aspects. Details on key management are presented in Section 4.2, followed by the related work in Section 5.3. An overview of traditional biometric Cryptosystems will be the content of section 6.4. Section 6.5 has been devoted to present the proposed solution, while its security analysis is presented in section 6.7. Section 6.8 summarises the chapter.

6.1 INTRODUCTION

Conventional public key encryption systems (PKEs) (or asymmetric keys) rely on two different keys— public and private keys (pk, sk)— in its infrastructure. Asymmetric keys are a vital element of securing electronic goods through unsecured (or public) channels, i.e., open networks. Senders, therefore, must already own their key pairs to use the PKEs. To carry out a decryption process, recipients also need to own their keys. The public keys of the users come exclusively from random strings, often not affiliated with any of their identities. Therefore any user u_i needs a public key pk_i, they must communicate with a well-known, trust-based certification body (CA), which is the only entity that connects $u_i$with pk_i, to issue the corresponding digital certificate.

It can be argued that the presence of different CAs among users adds further requirements between authorities and consequently increases the burden on users. Also, the increase in the number of users will be reflected in the number of digital certificates, which in turn contributes to high storage cost and the requirements to verify and revoke the certificates, plus a time factor.

Another issue that also imposes a significant challenge in adopting PKEs is the key exchange (or key establishment). Given a PKE system of *n*-users, it requires 2*n* pairs of keys. For vast numbers of the users, it is necessary to set up particular mechanisms in order to manage their keys. Key Distribution Centres are systems used to mitigate the challenges that may result from the exchange of keys [97]. The existence of these systems does not mean that the optimal solution is supplied. Diffie-Hellman (DH) is one of the most widely used protocols for key exchange [98] wherein its general form; it is secure against eavesdropping but not secure against man-in-the-middle attacks [83]. Existing solutions to overcome the man-in-the-middle attack incorporate authentication of two trusted parties, which cannot be adopted in the cloud due to the absence of an agreeable trust model in the cloud. For further clarification, Figure 6.1 shows a general framework of traditional PKEs.

IBCs has been introduced to overcome the above-mentioned complexities associated with PKEs [18], [21]. IBCs eliminate the PKEs requirements related to public key distribution by using users' public identities instead so that the keys are directly related to users' identities as illustrated in Figure 6.1.

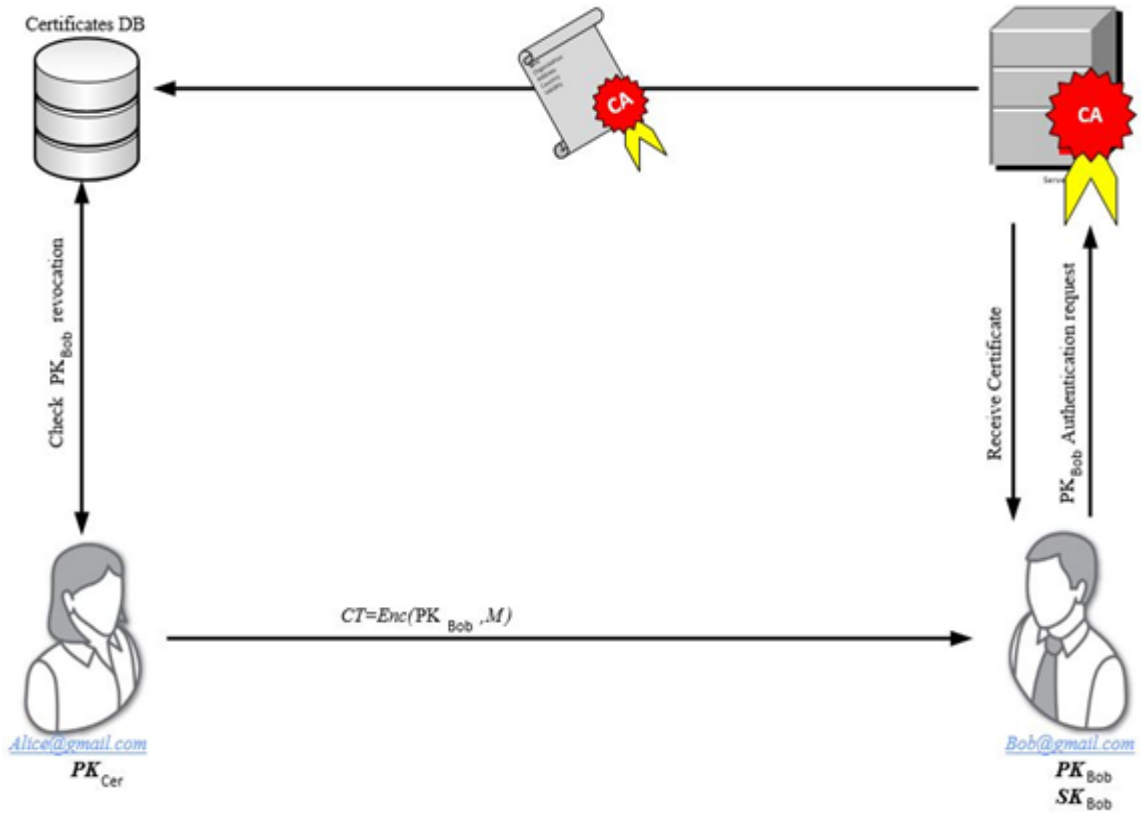


Figure 6.1: A general framework of public key encryption system (adapted from [99])

Later on, [17] introduced a new paradigm of IBCs to overcome IBCs authentication vulnerability. They have adopted biometric user data as an alternative to traditional identities. Chapter 3 and 4, however, contain abundant information relating to F-IBCs and IBCs.

It is essential to realize that the distinct characteristic of all existing IBCs and F-IBCs is that the encryption/ decryption keys are mainly derived from two main parameters: MSP and MPPs. Moreover, the MSP, used for generating the decryption keys, is entirely under the PKG's control. Hence, it gives PKG full control over keys' management and distribution. The IBC and F-IBC schemes would allow the occurrence of PKG-based key escrow (PKG-BKE) problem. In other words, all the encrypted data can be available to decryption by PKG even without the consent of DOs.

On the other hand, storage all MSPs of users on a central PKG database create a central point of attack. Once an adversary succeeds in attacking the PKG server, it implies all encrypted data is susceptible to abuse. These challenges discourage DOs from adopting F-IBCs and IBCs to protect their data.

This chapter, thus, aims to provide a new solution for mitigating security vulnerabilities by bringing the control back to users. The re-control process involves moving the MSP,

after they are generated by the PKG server, to the users. The MSP is then discarded from the server location to prevent it from being reused. It is argued that the proposed solution gives the DOs further power over their data so they can decide who can access their encrypted data by controlling the process of creating the decryption keys.

6.2 Key Management of PKEs and IBCs

Encryption systems usually depend on mechanisms for the management of their keys. An encryption key management is a set of processes that aim to protect and preserve the encrypted data, as well as the keys used throughout the key lifecycles [100][101]. A set of operations covered by keys management include *creation*, *distribution*, *storage*, *utilization*, and *revocation* of keys[102].

Initially, the process of issuing the public and private keys of traditional PKEs is carried out concurrently, while the process is different in the case of IBCs. The release of the IBC-related public keys is primarily dependent on the accessibility of MPPs [18], which implies that the procedure for the issuance of the users' decryption keys is altogether independent of the process for generating their public keys.

Furthermore, the process of issuing digital certificates associated with traditional public keys usually precedes the stage of generating the public keys. Trusting Authorities (TAs) should constantly check for the validation of the certificates. It is also essential to point out that the only thing that connects traditional public keys to their private keys is the certificate, which is why the certificates are essential but they could result in additional burdens for users of PKEs.

In IBCs, the public key digital certificates have been replaced by distinctive user identities, and thus the need for users to connect to CAs is eliminated. IBC systems, on the other hand, makes the process of issuing public keys independent from the private keys, which gives the ability to verify the validity of the private keys at different stages. The public keys are normally issued by the same party as the private keys (i.e. the PKG), whereas in PKE, a user needs to communicate with a third-party (i.e. CA) so that the public key can be verified.

Another point of difference is that the generation of public and private keys of PKEs is based on an arbitrary random secret while users' public identities are the key inputs of

IBCs' users. This could be the main motivation of why most users prefer to use IBCs rather than PKEs in their applications [101].

PKE's decryption key is provided by trusted authority or users [101], while the PKG server controls the whole process for IBCs [18]. However, storing the MSPs on the PKG site highlights another challenge which presents —e.g., IBC key-escrow [17], [18], [54], [56], and central point of attack. Furthermore, managing the MSPs by PKG means to take complete control from DOs to PKG and causes PKG-BKE. The MSPs have been shown to be the primary factor used by PKG in IBCs to produce user decryption keys. Therefore, PKG-BKE implies the PKG can decrypt all the encrypted messages. Table 6-1 shows the IBCs infrastructure with regards to the MPPs and MSPs for different proposed schemes. It also describes how these parameters constitute the public (encryption) and decryption keys.

Keeping sensitive data or products in one place makes them vulnerable to a central point of attack; thus, the system, which was designed to protect those data/files would be ineffective. Such case fully applies to PKG infrastructure, because it is the only one who is empowered with managing and storing MSP—hence storing MSP in PKG site may expose to the central point of attack.

Boneh and Franklin were of the first to resolve the key escrow-based PKG. They have suggested a mechanism to avoid the key-escrow issue of IBCs by employing multiple trusted authorities (TAs) rather than one and choosing a threshold value for re-aggregating the master secret parameter (MSP) then issuing the private key[18].

Table 6-1: Different IBCs schemes and their encryption and decryption keys based on MPPs and MSPs.

Scheme		MSP	MPP	Public key	Private key
[18]		$s \in \mathbb{Z}_p$	$(p, n, P, P_{pub}, G, \hat{e}, H)$	sP	sQ_{ID}
[58]		$s \in \mathbb{Z}_p$	$(P, P_{pub}, G_1, G_2, \hat{e}, H_1, H_2)$	sP	sQ_{ID}
[56], [70]		g_2^{α}	(g, g_1, g_2, U, k)	$H_k(ID)$	$g_2^{\alpha} * \ldots.$
[17]	N.U	t_i, y	$T_i = g^{t_i}, Y = e(g,g)^y$	$w' \in \mathbb{Z}_p$	$D_i = g^{q(i)/t_i}$
	L.U	y	$(g_1, g_2, t_i, \ldots, t_{n+1})$	$w' \in \mathbb{Z}_p$	$D_i = g_2^{q(i)} T(i)^{r_i}$
[30]		w, t_1, t_2, t_3, t_4	$(e(g,g)^{wt_1t_2} g, g_0, g_1, t = g^{t_i})$	$w' \in \mathbb{Z}_p$	d_0, d_1, d_2, d_3, d_4
[65]		$s', s \in \mathbb{Z}_p$	$(P, G_1, G_2, \hat{e}, g, g_1, g_2, H_1, H_2, H_3)$	$h_{ID} = H_1(ID)$	$h_{ID}^{s'}, h_{ID}^{s}$
[103]		β, g^{α}	$(G_0, , g, h = g^{\beta}, f = g^{1/\beta}, \hat{e}(g,g,)^{\alpha})$	$w' \in \mathbb{Z}_p$	$D = g^{(\alpha+r)/\beta}, D_i = g^r . H(i)^{r_i}, D_i = g^{r_i}$
[60]		$x_1, \ldots, x_n, \beta$	$(G, F, g, P_{pub}^i = g^{x_i}, \hat{e}(g,g)^{\beta}, H_1)$	$w' \in \mathbb{Z}_p$	$D_i = g^{q(i)/(x_i+h_i)}$
[104]		$s \in \mathbb{Z}_p$	$(q, G_1, G_2, \hat{e}, n, \alpha, \beta, H_1, H_2, H_3, H_4)$	$w' \in \mathbb{Z}_p$	$sk_{ID} = x + x_i + H_2(ID)^s$

6.3 Related Work to the Key Escrow Problem

According to Dorothy E. et al. [19], the key escrow (or key recovery) system is an encryption system that allows authorised individuals, e.g., persons, officials of an enterprise and the government, to decrypt an encrypted message— under certain circumstances and according to the retrieval criteria by recovering the decryption keys from one or more trusted entities. Note that, the recovery key is typically not necessarily similar to those used in data encryption/decryption, it provides a way to determine the encryption/decryption key [19] [20].

In IBCs, the key escrow is an inherent property due to the decryption keys being primarily dependent on public identities and MSP, and hence, the PKG knows all the users' decryption key [105], [106],[107].

In IBCs, the MSP is a significant component for keeping encrypted data secure. For that reason, it gives the PKG full control over the encrypted data rather than the DOs.

Moreover, despite the advantages that the IBCs can offer [60], [68], [69], [103], [104], [108], the mechanism in which the MSP is managed by the PKG can lead the system to PKG-key escrow problem. It means that all their private keys corresponding to their public keys (here public identities) have become available in principle to the PKG. It consequently gives the PKG server a full potential to decrypt all the encrypted messages. For that, many studies have been carried out to address or alleviate the dominance of the PKG server in various encryption systems [109]–[112].

Boneh & Franklin [18] suggested that countering the downside of key-escrow by distributing the MSP across several authority servers. To complete this task, they also employed Shamir's secret sharing technique [57]. Therefore, it is necessary to share *d* (two or more)-authorities to reconstruct the MSP, where *d* represents the value of a pre-agreed threshold.

L. Chen et al. [113] made use of the scheme in [18] to overcome the problem of key-escrow. They employed a multi-authority system such that each authority has its separated MSP. They suggested two types of keys—standard encryption/ decryption keys (R, s) and identifier keys (Q_{ID}, S_{ID}), where $R, Q_{ID}, and\ S_{ID} \in \mathbb{G}_0$, and $s \in \mathbb{Z}_p$. There is some trusted authority (T_A) associated with the standard encryption/ decryption keys given by (R_{T_A}, s). It leads the following relation will establish between the keys in T_A and the identifier: $S_{ID} = sQ_{ID}$ where $Q_{ID} = H_1(ID)$. They assumed if there are n trusted authorities T_A, each one has own standard encryption/decryption keys, such that: $R_{T_{A_i}} = s_i P$ for $i = (1, \dots, n)$, where $\langle P \rangle$ is a generator of $\mathbb{G}_0$. All the $R_{T_{A_i}}$ are expected to be trusted by all the entities within the system. Also, for a certain fixed identifier ID, there are n corresponding decryption keys given by these authorities as follows: $S_{ID_i} = s_i Q_{ID}$ for $Q_{ID} = H_1(ID)$.

The decryption key then can be calculated based on the following formula:

$$S_{ID}, b = \sum_{i=1}^{n} b_i S_{ID_i}, \tag{6.1}$$

Where b is a bit string, i.e., $b = (b_1, \dots, b_n)$, and the corresponding encryption/ decryption keys related to T_A can be calculated as follows:

$$R_{T_A}, b = \sum_{i=1}^{n} b_i R_{T_{A_i}} \; and \; s_b = \sum_{i=1}^{n} b_i s_i \tag{6.2}$$

For n T_A, it will produce 2^n different encryption/decryption key pairs, which were called "key addition". By choosing any subsets of trusted authorities, senders can divide this key addition over several trusted authorities on the fly at the time of encryption. Besides, this construction makes it resistant to a central point of attack. However, both proposed schemas [18][113] add an extra burden to the users since it requires managing the MSPs across different authorities, time-consuming, as well as each user need to present his/her identity to each sole authority.

Z. Cheng et al. developed the standard IBC scheme of [18] by [105] when they suggested a secure channel between entities upon different domains for mutual authentication required as well as keys agreement protocol. In order to remove the IBC inherited key escrow, they introduced another encryption/decryption key pair (N_{ID}, t), where the decryption key $t \in \mathbb{Z}_p^*$ is selected by the user who is the owner of an identity ID. The encryption key N_{ID} can be computed based on the decryption key t as $N_{ID} = (pk_1, pk_2) = (tP, tP_{pub})$, where $\langle p \rangle$ and P_{pub} are similar to those in Boneh and Franklin IBC scheme. Instead of using only two the key pair (Q_{ID}, d_{ID}) in [18], it made the encryption/decryption process subject to the presence of another key pair (N_{ID}, t). Since only the user who owns the identity ID knows t, this will control the impact of key escrow related to IBC. However, in the event that the number of subscribers is rather growing, this poses a key management problem.

Other such as C. Gentry et al. [66] suggested using a hierarchical structure in their proposed scheme to play the role of preventing the server key-escrow. It, however, also does not meet the optimal solution because the MSP has still threatened the users' privacy[105].

A new trend to address the problem under discussion is by introducing a fine-grained revocation technique [114]. It takes advantage of the same scope to support a key agreement protocol. The process of generating the decryption keys goes through two stages where the users initially receive a portion of their private keys from the PKG server. The second portion comes from a secret built-in value that is picked up by the same users. This latter portion of the private key transmits to another online authority. It thus will be released after the user authenticates themselves to the online authority. Since the users possess the right secret built-in value (a big integer), then considers as a genuine user; consequently, the secret key will then be configured. In other words, there are three parties involved in the building of the private key, not just the PKG.

On the other hand, the work carried out by B. Lee et al. [115] relied on employing multiple key privacy authorities to maintain the privacy factor. Users need to retrieve their data, to apply a particular protocol in which only the rightful user who has secret blinding parameters can gain the corresponding private key. However, traditional PKG server is still in charge of storing users' private keys. Besides, this type of solution is complicated and could be subjected to excessive computation and communication. It is the view of B. Lee et al. that adopted a traceable identity-based cryptosystem as a technique advised by [116] to prohibit any misuse that the PKG can do. Besides, they argued that their proposed scheme secure in Indistinguishability against Adaptive Chosen Ciphertext Attacks. As a standard IBC, they focused on their scheme on applying single PKG rather than multiple. The key idea of TIBC depends on establishing a protocol that transfers a private key sk_{ID} of an identity *ID* securely from the PKG to a user U_{ID} based on his/ her identity. Ordinarily, the secure key generation protocol, as it is called, prevents the PKG from knowing which private key is delivered to a user. Hence, the user U_{ID} is going to have two different private keys sk_{ID} and sk'_{ID} if the PKG decides to issue the private key sk'_{ID} for the identity *ID* for malicious use. Moreover, they adopted a "*catch and prosecution"* mechanism against the PKG, who issues and distributes various private keys to the same identity.

On the other hands, some security concepts have been used by other researchers in an attempt to develop a robust scheme against the key escrow which could happen because of the use of IBCs. In [111], they proposed an anonymous ciphertext indistinguishability technique which was introduced in [117] by adopting multi-authorities in charge of issuing an identity certificate. After that, these certificates will be utilised to deny the server from acquiring any information that would reveal users' identities at the stage of obtaining their private keys. However, considering the users' identities (the public keys) will become mostly dependent on the digital certificate; therefore, it eliminates the most imporant characteristics that distinguishes the IBCs from conventional public key encryption systems PKEs.

6.4 Traditional Biometric Cryptosystems

The proposed solution described in Section 6.5 relies on user biometric information to solve the PKG-BKE and central point of attacks issues. For this reason, this subsection provides an overview of biometric cryptosystems.

Biometric Systems (BSs) were generally invented to support demands for identification and authentication. Feature extraction techniques have been employed to produce distinctive feature vectors which represent the most critical phase in BSs besides it is liable for inferring individual personality traits for security aims. Biometric recognition systems permanently appropriated as automatic tools to identify individuals relied on physiological and behavioural characteristics [118]. The identification/ authentication procedure conducts by comparing a new biometric template against the stored biometric template(s) [118].

Commonly, there are two main phases in biometric systems, namely **enrolment** and **identification/authentication** phases. The enrolment phase involves scanning users' image using an appropriate sensor, extracting distinctive features, producing feature vectors as biometric templates, and storing them in a database. In the second phase of biometric recognition, the same processes used at the enrolment phase is repeated, but on fresh biometric data. The decision of accepting or rejecting depends on a comparison between the biometric template (s) stored in the database and feature vector resulting from the authentication phase.

Biometric traits used in Biometric Cryptosystems (BCs) recently through the combination of cryptography and biometric data for security purposes.

The BCs are of significance in two major applications: either to bind cryptographic keys securely to a users' biometric data or to reconstruct bound keys from a users' biometric data [119]. The following key points outline the way of employing BCs for binding or reconstructing the cryptographic keys.

1. **Key binding systems** (KBSs). KBSs describe the manner in which a helper (or subsidiary) information is produced. The helper data is created by binding a cryptographic key to a user's biometric template. Instead, a key's recovery (or reconstruction) carries out at an authentication or matching phase using the helper data and biometric query template. KBSs can be classified into the following two commonly systems:
   - **Fuzzy vault systems** (FVSs). FVSs has been invented by A. Juels and M. Sudan [120] to lock a chosen cryptographic key utilizing a user's biometric feature vector λ. Further, the FVSs has two main levels: encoding and decoding levels. The encoding level is carried out at the enrolment stage while the latter at the authentication stage. One more time, the user's biometric feature vector λ' is

created in the authentication stage by presenting his/her fresh biometric data. The authentication is successful as long as both λ' and λ are close enough, which indicates the same person.

Besides, error code correction (ECC) mechanisms are widely considered to be the core element of biometric intra-variation mitigation, and thus, it deems to be an essence of the FCs [121].

- **Fuzzy commitment systems** (*FCs*). This sort of KBSs was introduced by Ari and Wattenberg to safeguard biometric templates using the ECC mechanisms [122]. It proceeds by extracting a user's feature vector in an enrolment phase then binarizing to generate a binary feature vector $B_{FV} \in \{0,1\}^n$ of length *n*. Moreover, a chosen secret key is also binarised to form a binary key $k \in \{0,1\}^n$ whose length is equal to the length of the binary vector. The next process of FCs involves encoding *k* using the ECC to produce a codeword $C_w = (c_{w_1}, c_{w_2}, ..., c_{w_n})$. After that, a fuzzy commitment η is generated by XORing B_{FV} with C_w ($\eta = B_{FV} \oplus C_w$); hashing the resulted fuzzy commitment to generate an auxiliary data ($H(\eta)$). Ultimately, a particular database is selected that may store in addition to the auxiliary data, the hash of codeword $(H(\eta), H(C_w))$. In the authentication phase, the same process of binarizing is applied over a fresh user's biometric query to output $\bar{B}_{FV} \in \{0,1\}^n$. Then, the resulted $\bar{B}_{FV}$ is XORed with the auxiliary data $H(\eta)$ stored in the database. The authentication is successful if and only if they are close enough; otherwise, it is rejected.

2. **Key generation systems** (KGSs). These systems are used to reconstruct the cryptographic keys. To this end, since the auxiliary data mainly relies on the user's biometric template, both of the auxiliary data and the biometric template have to exist to release the keys.

6.5 The Proposed System

This section explains the proposed MSP binding based on user biometrics to address a range of vulnerabilities related to the use of F-IBCs. The basic idea of the proposed solution lies in improving the security of the MSP in F-IBCs by switching the control

from the server (i.e. PKG) to the user (recipient). This means the decryption keys will, therefore, be generated under the user management rather than the PKG. The proposed scheme, as any biometric-based verification system, has two stages— an enrolment and verification stages, as highlighted in Figure 6.2.

The enrolment stage begins by capturing the recipients' biometric samples, then extract distinct features vector (FV) using an appropriate extraction mechanism, which ultimately produces the corresponding biometric templates. In order to safeguard the resulting templates, the subsequent step is developed to produce cancellable templates by converting them to another secure domain. This step takes place by generating a random orthonormal matrix based on a user PIN/Password. The adopted user-based transformation relies on the personalised random projection introduced by [74]. In order to conserve the security and privacy requirements, only the cancellable version (FV_c) based on a user PIN/Password will be submitted to the PKG server. Applying the binary operation on the FV_c is the subsequent step to generate a binary vector (BFV_c). On the other hand, due to the Intra-Class variations among the biometric samples, the proposed scheme employs an error correction code (ECC) to address these variations in the enrolment stage.

It is worth mentioning that the operation of creating the F-IBC's parameters (i.e. MSP and MPPs) in our scheme remains as it is. Once the MSP is bound to a user's biometric, it will then be ignored from the domain of PKG. After generating the MSP by PKG, it will be subjected to an error correction code (ECC) encoding. The amount of corrections relies on the size of the MSP and biometric templates, as well as the biometric data's error tolerance of user biometric samples. Next, a binary operation is carried out between the BFV_c and the encoded MSP to produce a biometric lock (BL) (or an auxiliary data). An exclusive OR (XOR) operation was chosen to play the role of the binary operation.

The resulting BL is then stored on the user's token, e.g. USB, smart devices, or any other storage media. Hence, the operation of generating the decryption key has become dependent on this token as well as the user's identity (multi-factor).

As shown in Figure 6.2, the procedure of releasing the MSP is officially conducted in the verification stage which adopts the same steps used in the enrolment stage, but in a reverse manner. Therefore, in order to handle the verification stage, each user (recipient) must submit two things— a fresh biometric data and the token. The following steps describe the protocol of the user verification stage:

- The user needs to provide a fresh biometric sample then system extracts the corresponding features vector (FV') using the same features extraction method in the enrolment stage.
- Next, the user retrieves their PIN/ Password to generate the random orthonormal matric which is used to produce the cancelable version FV_c' of user FV'.
- The same binary transformation approach is then implemented to generate a binary vector ($BFV_{c'}$) to be XORed with the BL stored on the user's token.
- The subsequent procedure includes exposing $BFV_{c'}$ to the ECC decoding process in order to release the MSP.

The original MSP reconstruction is achieved if and only if the referenced biometric template is close enough to the fresh biometric sample, i.e. they are within the pre-specified threshold value. In the event that the requirement is met, it implies that both of them (the biometric samples) belong to the same person; otherwise, the samples are of different persons. The next sub-sections describe the practical implementations of the proposed MSP binding.

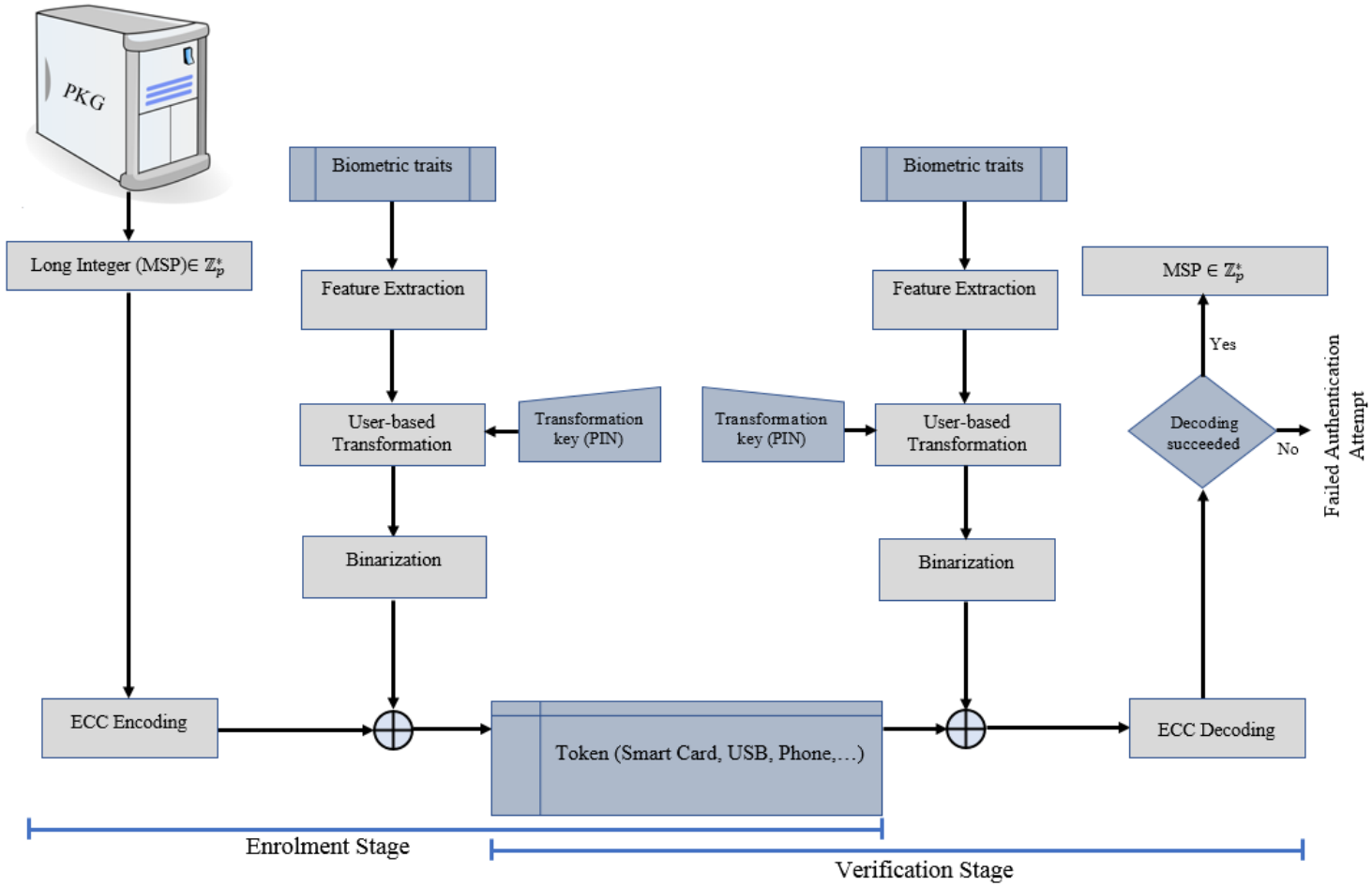


Figure 6.2: A General Proposed A Scheme For Locking Msp Using A User Biometric Data

6.5.1 Biometric Facial recognition-based MSP binding

In our implementation, we select face biometrics due to image cameras installed in most contemporary smart devices and they can be used as a suitable biometric data reader (scanner). As shown in Figure 6.3, three main factors govern the process of binding the MSP to user biometrics—the length of the MSP, the ECC technique used, and the size of FVs derived from user biometric data. As stated before, for security and privacy requirements, only the cancellable version of the biometric vector is submitted to PKG at the enrolment stage.

To clarify further, we present the following scenario. The user of the IBC system can use the camera on their smart device (Phone, Tablet, Laptop, etc.) to capture their face images. The user-based transformation is then derived from PIN/Password to be used for generating the orthonormal random projection. As a proof-of-concept, we use a Discrete Wavelet Transform (DWT) using the Haar filter for features extraction to produce distinct features vectors (FVs) for face recognition. In order to make the length of FV appropriate to deal with the length of MSP, our experiments rely on using low-pass subband (LL2) in which it describes the second level resolution approximation of each image in the ORL face database (described and used in Chapter 4) to produce 644 features. After generating the features vector, the process of building the *Orthonormal Random Projection* (ORP) introduced by [82] is carried out on this features vector to tackle biometric diversity and revocability. ORP's significance lies in maintaining the same distances before and after execution between the biometric features. Reed-Solomon (RS) is empowered to provide an appropriate *error-correction code* (ECC) strategy to tolerate intra-class variations connected with a user's biometric data. As is customary with biometric systems, the biometric sample used in the enrolment stage (MSP binding) differs from those used in the verification stage (MSP reconstruction). For this reason, RS (or any other ECC) is used. The RS includes the following parameters: the number of bits per symbol is (m = 9) and consequently the codeword length ($n = 2^m - 1$) will be (n = 511). Based on n, the first 511 of each FV will be selected. The other significant parameter of RS is the message length (k=163) referring to MSP, where $MSP = \{0,1\}^k$. Hence, we choose RS (511,163) to carry out our experiments.

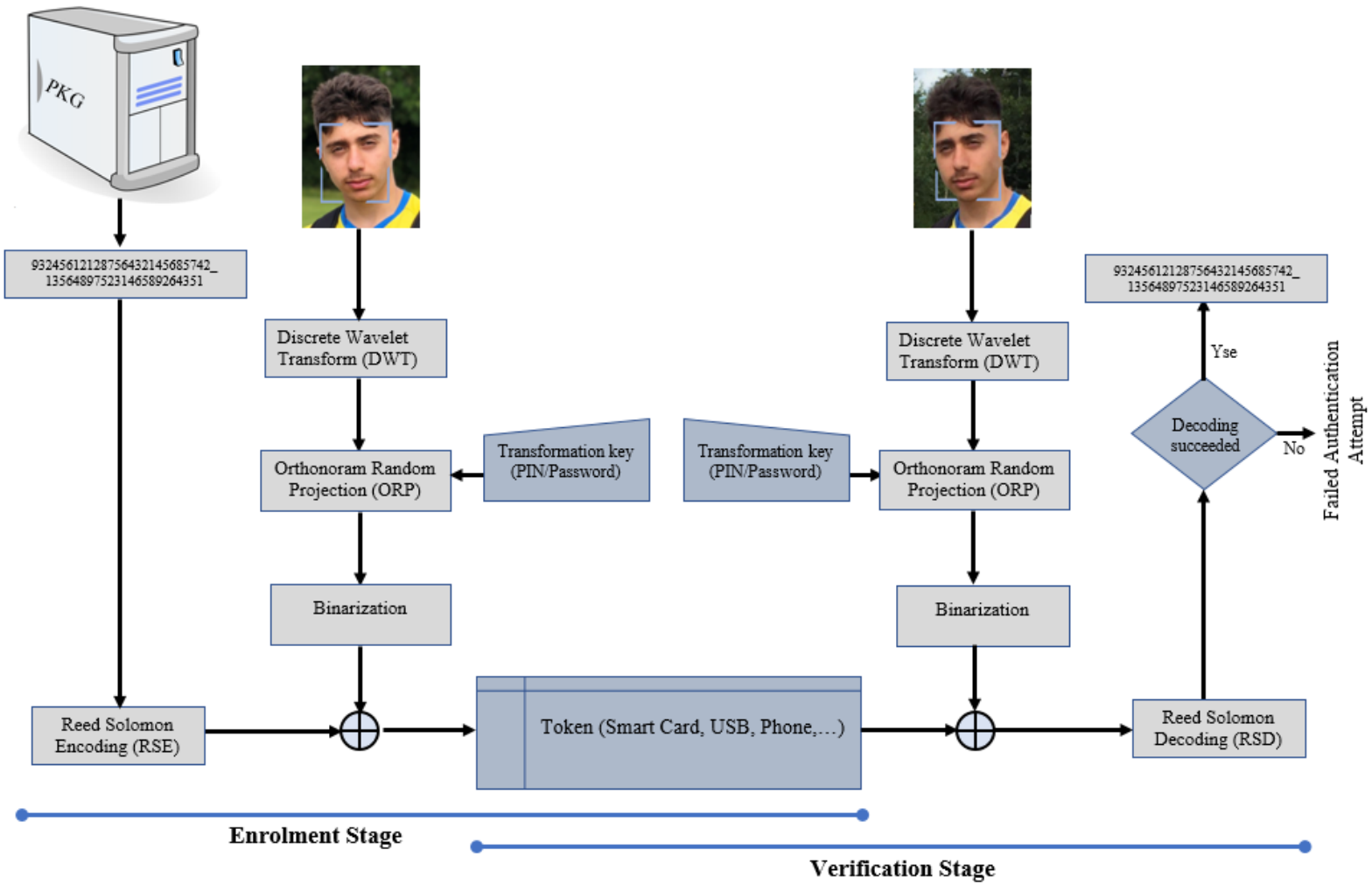


Figure 6.3. The proposed scheme of MSP binding based on Face recognition using LL2

A biometric cryptosystem can bind the MSP with maximum size *k* using the ECC and correcting up to *t* bits. In case of choosing the Reed Solomon (RS) as an *error-correction code*—as seen in the above figure, each codeword should have *n*- bits length— while the MSP length is *k*- bits. For this construction, it is committed to correcting $t = (n - k)/2$, with the constraint that $n - k \in 2\mathbb{Z}^+$. Therefore, the process of choosing the ECC encoding parameters should be carefully depending on analysing error patterns concerning Inter and Intra-classes variations of biometric samples. In our application, RS(511, 163) is selected to correct up to 18% of face FVs. A threshold is determined (30% in our work) for the definition of biometric verification thresholds, i.e. identifying a convenient tolerance of equal error rate (EER) in relation to false accept rate (FAR) and false reject rate (FRR) depends on a training operation. Such thresholds depend on the application's context—certain applications require strict tolerance while others do not. DWT (and LL2) was used as the feature extraction techniques, however, any other technique can be used to play this role depending on the parameters of RS that determines the size of FVs that is used to bind the MSP.

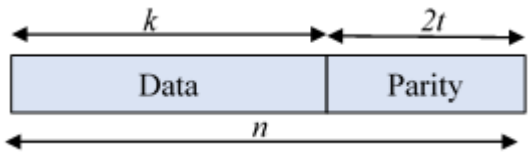


Figure 6.4: A structure of codeword in RS (n, k)

The binarization system of [82] is used to support the binary conversion. The binarization process is conducted as follows: in the training set, the mean vector γ is calculated for every feature vector of all clients comparing with user's feature vector μ of size *n*, the binary feature vector, β, is obtained based on the following formula:

$$\beta(i) = \begin{cases} 1 & if\ \mu(i) < \gamma(i) \\ 0 & otherwise \end{cases}$$

The process of generating the binary vector will take place on FV_c. In the meantime, the master secret parameter (MSP) is generated and will be ready to bind to the user's biometric.

For each person's images in the ORL database, we define the first three images as reference (training) images while the other seven images remain for the testing process. In order to conduct the matching process, our proposed solution depends on the Euclidean distance between biometric FVs, i.e., the one that has the minimum distance with the other seven FVs of each person will be adopted as a matching consequence.

6.5.2 Fingerprint recognition-Based MSP binding

In this experiment, we used the Second Fingerprint Verification Competition (FVC2002-DB2) database [123]. An optical sensor is employed to produce this dataset which embraces 100 distinct fingers (8 impressions per finger) of size 560×296 and resolution 569 dpi. It has benefited from the work published by [124]in the operation of features extraction, classification and matching. Also, we use the same techniques used in face recognition to protect and binarize the fingerprint traits vectors. In order to overcome the disparity in the sizes of templates that can be generated in the fingerprint recognition systems based on minutiae, it has been replaced by the FingerCode-based fingerprint recognition method proposed by [124] to produce a fixed-length of templates. For conducting the matching process between every two corresponding FingerCodes, a Euclidean distance is used. The algorithm of generating the FingerCode can be represented as follows: tessellation the Region of Interest (RoI) surrounding the reference point, then relied on the x-axis, set of Gabor filters are utilised in the following different eight orientations—0°, 22.5°, 45°, 65.5°, 90°, 112.5°, 135°, 157.5° (refer to [124] for further clarification). The ROI of our work was portioned into 64 sectors such that each one can be represented by only one value resulted from applying a standard deviation on the eight filters. The fixed-length FingerCode traits vector consequently will be 512 traits (64×8 discs - eight orientations) as described in Figure 6.5. Also, the figure depicts the proposed scheme of MSP binding based on Fingerprint recognition.

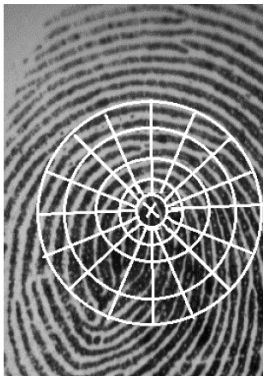


Figure 6.5: Sectors of Fingerprint ($16 \times 4=64$) based on the reference point (x) and the ROI (retrieved from [124])

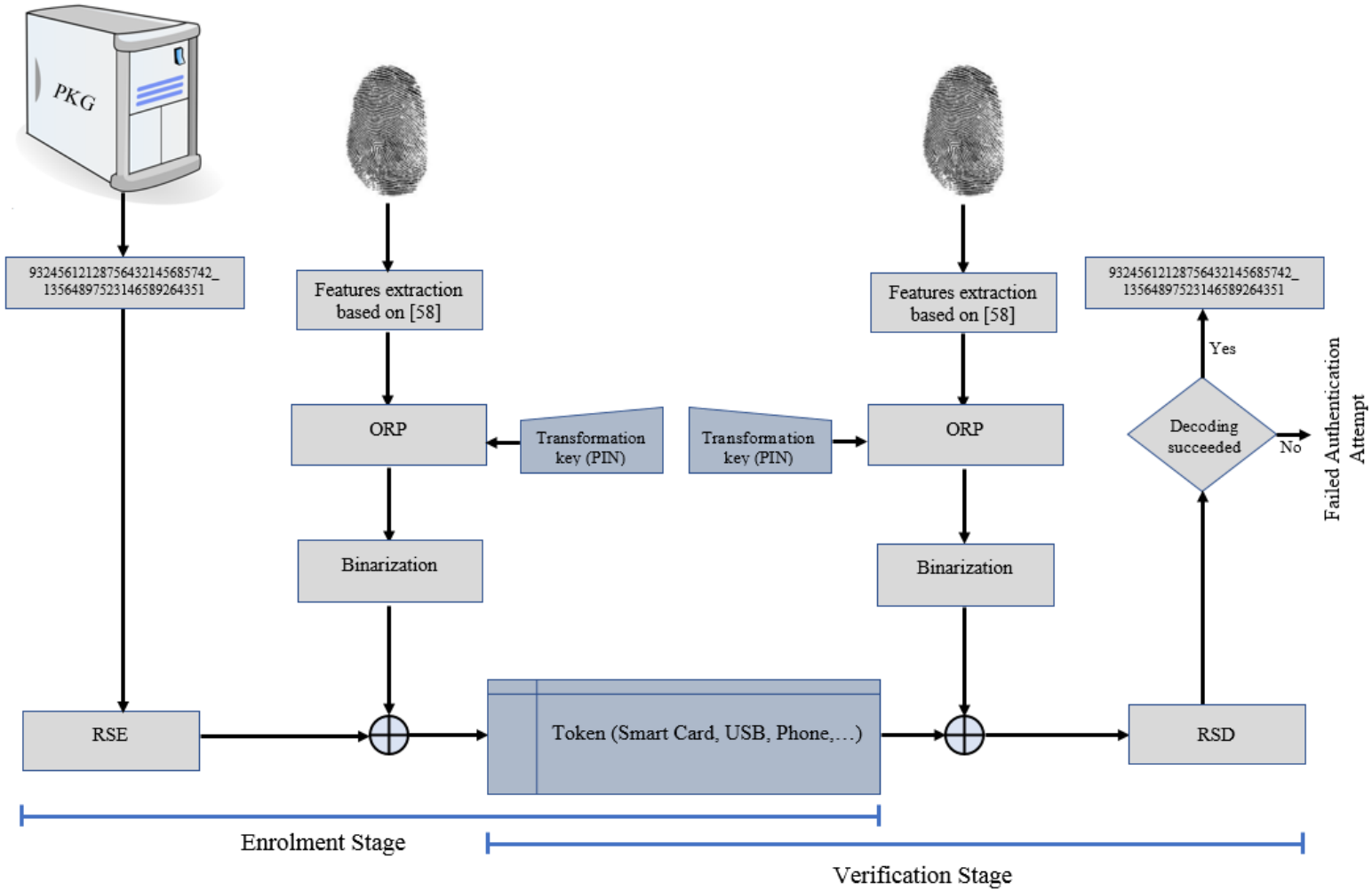


Figure 6.6: Proposed scheme of MSP binding based on Fingerprint recognition

6.6 Biometric Evaluation

Our proposal was evaluated on the following two databases:

a) The Cambridge Olivetti Research Lab (ORL) face database (described in Chapter 4).
b) The Second Fingerprint Verification Competition (FVC2002-DB2) database[123].

6.6.1 Face evaluation

As proof-of-concept, we relied on a Discrete Wavelet Transform (DWT) to extract FVs from ORL dataset. The low-pass subband LL2 is utilized, that refers to the second level resolution approximation of the image, to produce vectors of length 644 features. The first three images for each user in face dataset were selected to represent the training set and the remaining seven images for the testing set. Even though, this is not a standard division of the data set for training and testing, however, this division has used within the proposed technique as a proof-of-concept, not necessarily for training and testing on a large data set. The performance of the face recognition-based MSP binding is reported with FAR and FRR as shown in Figure 6.7. This figure shows that the operational threshold (i.e., EER≅ 0) is between [23-26]. Based on this operational threshold, an unauthorized user cannot reconstruct the MSP.

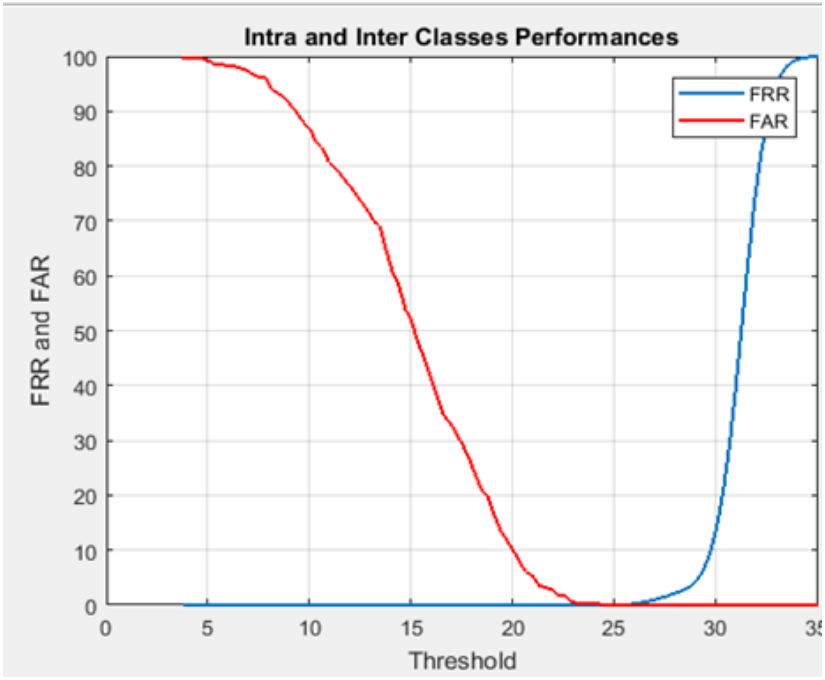


Figure 6.7: Face Authentication Accuracy based on FAR and FRR for Biometric Facial recognition-based MSP binding

6.6.2 Fingerprint Evaluation

In our fingerprint evaluation, we select only the first three impressions per finger; the first refers to a reference template, while the other two represent the testing. It, thus, forms 100 (1×100) images as a training set and 200 (2×100) images for a testing set. Also, the accuracy of fingerprint recognition-based MSP binding in regards with FAR and FRR can be shown in Figure 6.8. This figure also shows that its operational threshold, when EER ≅ 0, indicates that an unauthorized user cannot reconstruct the MSP.

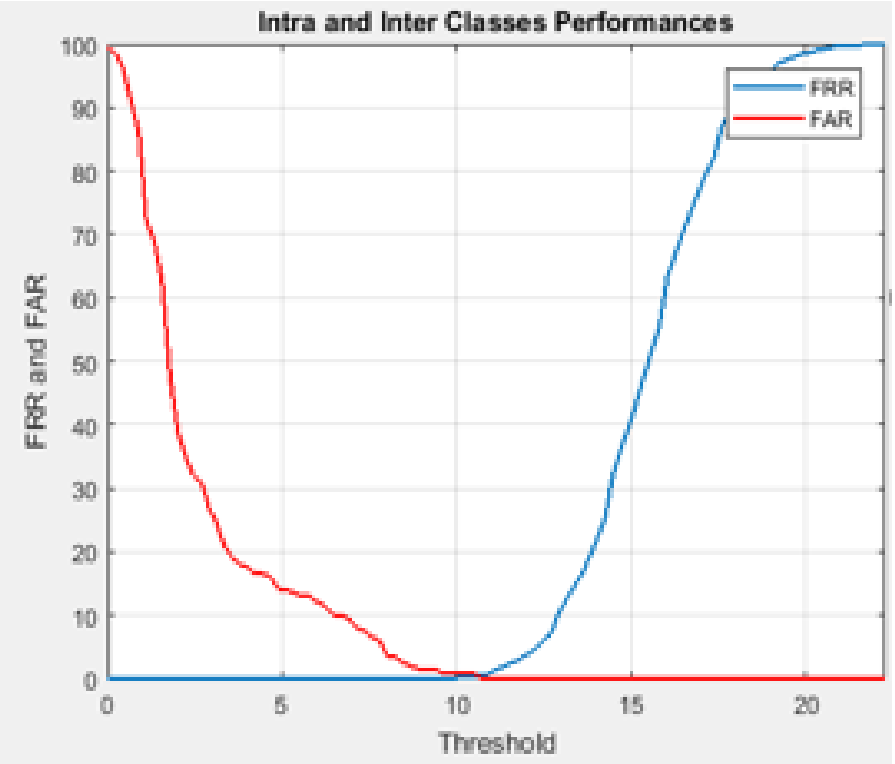


Figure 6.8: Fingerprint Authentication Accuracy based on FAR and FRR for Biometric Fingerprint recognition-based MSP binding

6.7 Security Analysis

This section intends to show the security analysis of two different modalities of biometrics—face and fingerprint—using three different scenarios:

- Scenario A: using biometric only.
- Scenario B: using biometric and a compromised key.
- Scenario C: using biometric and a secure key.

With regard to face recognition-based MSP binding, Error! Reference source not found. illustrates the above three scenarios as follows: A) The proposed scheme using only face biometric for MSP binding, B) The proposed scheme using face biometric with compromising the user-based transformation key in terms of ORP, and C) the proposed scheme using face biometric and securing the user-based transformation. As clarified in *Figure 6.9*, the accuracy (calculated in terms of FAR and FRR) for scenarios A and B are identical, which confirms that the transformation preserves the Euclidean distances between different templates. Scenario C, on the other hand, gives the perfect performance (zero Error Equal Rate (EER)). However, the process of gaining perfect performance requires adjusting the operating threshold; i.e. the number of ECC tolerable bits. For further security, we picked the operating threshold to be at EER point.

The same scenarios will also follow with regard to fingerprint recognition-based MSP binding. The accuracy of these scenarios can be observed in Figure 6.10, as follows: A) using the only fingerprint for MSP binding, B) using fingerprint with compromising ORP (i.e., compromising user-based transformation PIN/Password), and C) using fingerprint and securing ORP. In the case of the key is secured, each FingerCode will be protected via ORP using user-based key/PIN. Figure 6.8 shows that the scenario C has also the optimum performance in terms of Equal Error Rate (EER), while the scenarios A and B result in the same performance; in other words, the performance of only the fingerprint is precisely similar to the fingerprint with revealed ORP.

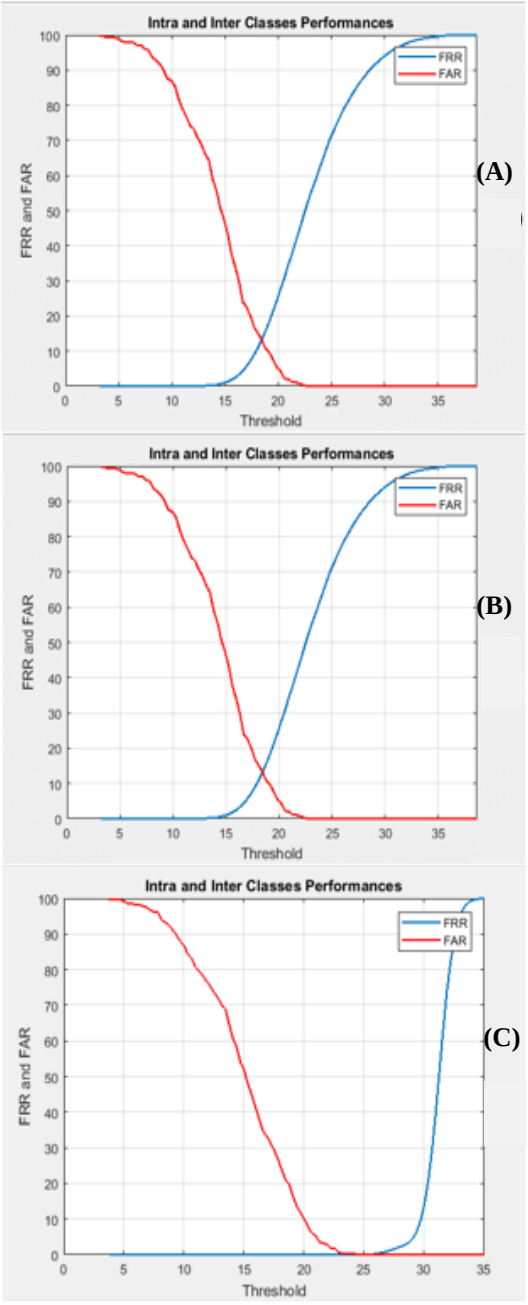


Figure 6.9:Face recognition systems using the three different scenarios

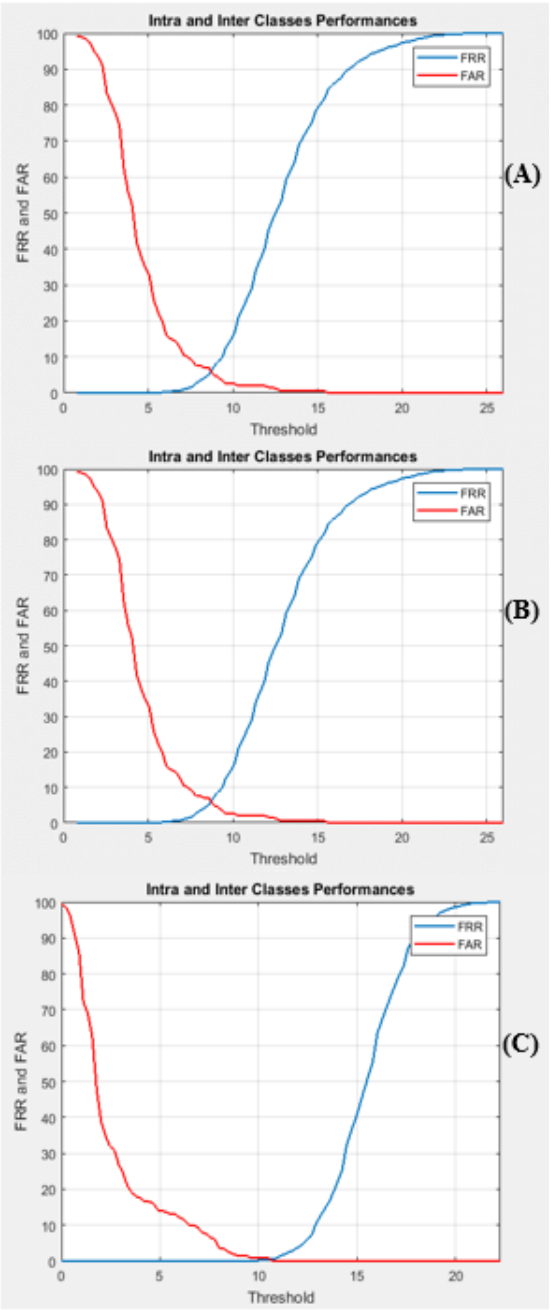


Figure 6.10: Fingerprint using the three different scenarios

By generating Biometric Cryptographic Key, biometric cryptographic systems are combined with biometrics to provide stronger safety mechanisms, thus defending against identity theft. MSP in our proposed biometric-based key binding is randomly generated under control of the PKG in an enrolment stage which would be completely independent of the biometric template (s).

Demonstrating a threat model scenario must be implemented on two different modalities of biometrics—face and fingerprint modalities. The difficulty to guess the random user-

based projection, as well as the way in which the user token is protected are two key factors in protecting the proposed

The security of the cancellable biometrics lies in its property as a one-way function i.e. the process of rebuilding the original bio-template from the transformed output computationally infeasible. Thus, the security of cancellable biometrics is primarily dependent on the user PIN/Password.

A user token highlights another challenge that should also be taken into account. In the face recognition-based MSP binding, if the token is stolen and the attacker can use the real user's public biometric, he again must guess the user's password/ PIN to generate the same user-based projection applied on biometric template. This obstacle will be more complicated and challenging in the case of fingerprint-based MSP binding it is private and not easily accessible.

The adoption of cancellable biometrics satisfies the diversity feature, which gives a positive answer to the following three questions: 1) is there a way to track the user without their consent by cross-matching their biometric templates over different databases? More precisely, what is the probability of knowing that the two cancellable templates belong to the same user? 2) if the biometric data becomes vulnerable to an attacker, what is the probability of access to the system while the key is not detected? 3) in the case of one of the transformed templates being stolen from the database, what is the probability of breaking another system that uses another cancellable template utilising a reply attack?

6.8 Chapter Summary

The process of generating the decryption keys in F-IBCs and IBCs can be considered as an essential element these schemes. The way that PKG manages MSP (storing and distribution) raised concerns regarding the DOs. The presence of MSP under the PKG control away from DOs may lead to the possibility of tampering. This tampering could be by dishonest PKG (PKG-based Key escrow) or by external attackers (the central point of attacks). PKG-based Key escrow allows the PKG to decrypt all the encrypted messages in addition to having a central point of attacks at the PKG side.

To address the above challenge, we proposed to bind the MSP with the users' biometric (fingerprint, and face). After the PKG generates the MSP to be linked with the user biometric, it will then be completely discarded from the PKG database.

We demonstrated that the proposal solves the above two challenges and supports the control of DOs over their encrypted data. Experimental results based on face and fingerprint recognition systems confirms the viability of the proposal.

Chapter 7

Key Exchange Using Biometric Identity Based Cryptosystem for Sharing Encryption Data in Cloud Environment

Storing and exchanging cryptographic keys between different parties over open environment such as cloud computing are the main problems associated with using symmetric/ asymmetric keys. A practical solution for session key-exchange for lots of web services is provided by Public Key Infrastructure (PKI) or public key encryption systems (PKEs). The need for a trusted third party (e.g. certificate authority) is not only the key challenge of PKEs solutions but also the missing link between the data owner and the encryption keys. The latter is arguably more critical where accessing data demands to be connected with the identity of the owner. Due to trust couriers or secure channels are existing key-exchange protocols, it can be subjected to a man-in-the-middle and various other attacks.

This chapter aims to introduce a new application of fuzzy identity-based cryptosystems (F-IBCs) by adopting it as a protocol in the exchange of symmetric keys. It permits parties to exchange cryptographic keys securely even if an adversary is monitoring the communication channel between them. The proposal incorporates IBC with biometrics which provides a secure channel to exchange symmetric keys utilising the parties' identities in an unsecured environment. Our proposed key exchange will also provide efficient access control that supports the control of DOs by allowing only the genuine user (biometric owner) to decrypt the encrypted message stored on cloud computing. In other words, it will give DOs the power of fine-grained sharing of encrypted data by controlling who can access their data.

The rest of the chapter is organised as follows. The introduction is presented in Section 7.1, followed by the proposed solution described in section 7.2. Section 7.3 provides a

full description of the proposed system as well as security analysis. Finally, the chapter concludes in section 7.4.

7.1 INTRODUCTION

The growing number of small and medium-size organisations have started to realise the benefits of transferring their data or applications to be hosted in a cloud environment [125]. Nevertheless, the increasing number of applications and the volume of sensitive information that individuals, companies, and organisation are storing on the cloud has led to serious security concerns. This is particularly important because once data is transferred to a cloud environment, the control is wholly conveyed to be in the hand of a third "trusted" party, i.e. cloud service providers CSPs. Therefore, the security of the data and the privacy of the users are the principal issues in the reluctance of some individuals and companies to use the cloud environment [125][126].

Further, the data owners cannot control who accesses their data due to the full transfer of control to service providers. Many researchers have focused on the possibility of protecting such data even if it is outside the physical control of the data owner. One intuitive solution to maintain data security is by encrypting the data before migrating it to the cloud. However, the key exchange or key establishment issue is a big challenge using traditional cryptography to exchange cryptographic keys between parties. The keys exchange are simple mechanisms used in the exchange of public keys (and some extra information) in order to guarantee secure communication between two parties.

There are a number of mature solutions in traditional cryptography to exchange keys based on the so-called Key Distribution Centres [127]. Diffie-Hellman (DH) is one of the most convenient protocols for key exchange [98] wherein its general form; it is secure against eavesdropping but not secure against man-in-the-middle attacks [83]. Existing solutions to defeat the man-in-the-middle attack incorporate authentication of two trusted parties, which cannot be adopted in the cloud due to the absence of an agreeable trust model in the cloud.

Identity-based cryptosystem (IBC) presented by [21] and described in chapter 3 is a fundamental step forward to solve the obstacles associated with key distribution in public key infrastructure. IBC eliminates the need for public key digital certificates and, therefore, the need for pre-distributed keys before any encryption/decryption in traditional cryptography will be illuminated, which gives a great deal of flexibility required in the environment such as the cloud. More importantly, IBC link decryption keys with user identities. Therefore, it enables data owner to be an integral part of selecting who can access their encrypted data in the cloud environment. Fuzzy IBC (F-

IBC) [59], on the other hand, is a further development of IBC in which users are issued with the decryption key (private keys) associated with their identities *id*. The user will be able to decrypt a ciphertext that was encrypted with the public keys of their identities *id'* if and only if the overlapping between id and id' is bigger than an agreed threshold. Further information about F-IBC can be found in Chapter 3.

Moreover, providing an effective data access mechanism is another concern and a necessary issue to protect cloud user data from unauthorised access. However, data encryption is not sufficient as PKG can grant access to any party without the consent of the data owners, even an illegal party. Further information about IBCs and F-IBCs were addressed in Chapter 3.

This chapter describes a new protocol that introduces a key exchange by integrating users' biometric data and the IBC scheme. A key exchange using biometric identity-based cryptosystem (KE-BIBC) establishes to permit parties to exchange cryptographic keys securely even if an adversary is observing the communication channel between the parties. Furthermore, KE-BIBC supports DOs by enforcing a new access control mechanism on their data by identifying which party is granted access to their data stored on the cloud computing, and consequently recognising who can share their data even when they are away from them.

The proposed protocol mainly use the users' biometrics as an alternative identity to the traditional IBC (which uses unique identifiers such as email, phone numbers, social security number) to provide a secure channel of exchanging symmetric keys based on the users' identities in an unsecured environment. In the KE-BIBC, the data owners encrypt their messages using traditional symmetric key then transfers the ciphertext to the cloud storage. Further, the data owner selects the public biometric data of the user (recipient) and encrypt the symmetric key. The chapter argues that the proposed protocol eliminates the needs for a key distribution centre as in traditional cryptography. Additionally, it guarantees that only the selected users can decrypt the ciphertext by providing a fresh biometric sample i.e. it ensures the data owner has the power of fine-grained sharing of ciphertext by controlling who they authorise to read their data.

7.2 The Proposed Solution: Key Exchange Based on Biometric IBC

In general, the proposed KE-BIBC system provides a new protocol for keys exchanging that enable two parties, Alice and Bob, to securely exchange cryptographic keys even when an adversary is monitoring the communication channel between them. Assume Alice has encrypted data (or message) stored in a cloud environment and she would like to give Bob access to the encrypted data. Typical PKEs solutions do not only require crucial pre-distributed management and a trusted third party (e.g. certificate authority) but also, they do not offer a clear link between the data owner and the encryption keys. Therefore, the proposed KE-BIBC protocol offers a practical solution that gives data owner (Alice) the power of fine-grained sharing of here encrypted data by control who can access their data.

The main stages of the proposed solution can be summarised as follows.

- Alice encrypts her data using traditional encryption e.g. using Advance Encryption Standard (AES)

$$\mathcal{E}_M \longleftarrow Enc(sk, M) \tag{7.1}$$

  - She stores the encrypted data in a cloud environment.
  - Now, if Alice wants to allow Bob to decrypt the message, she encrypts the AES encryption key *sk* using a public key of Bob's unique identity *w'* (i.e., Bob's biometric such as a photo of his face) to produce $\mathcal{E}_{sk}$.

$$\mathcal{E}_{sk} \longleftarrow Enc(pk_{id}, sk) \tag{7.2}$$

  - Alice sends the output $\mathcal{E}_{sk}$ to Bob.
  - To retrieve the *sk*, Bob needs to provide a fresh biometric sample *w*.
  - If and only if the overlap between *w* and *w'* is greater than a threshold value, Bob will retrieve the corresponding private key of his identity and decrypt the ciphertext to get the *sk*.

$$sk \longleftarrow Dec(sk_{id}, sk) \tag{7.3}$$

  - Bob downloads the encrypted data stored in the cloud environment to his local device, and uses *sk*, to retrieve the original message/data

$$M \longleftarrow Dec(sk, \mathcal{E}_M) \tag{7.4}$$

The chapter argues that since the face biometric data, for example, is public between parties who knows each other, it can be obtained from many resources such as social media resources (e.g., Facebook, Instagram, etc.). Hence, face recognition is an ideal biometric trait for our proposal.

The above stages are illustrated in Figure 1, which shows the overall framework of the KE-BIBC system to bind traditional encryption key *sk* with user's biometric data to provide adequate access control mechanisms for cloud storage.

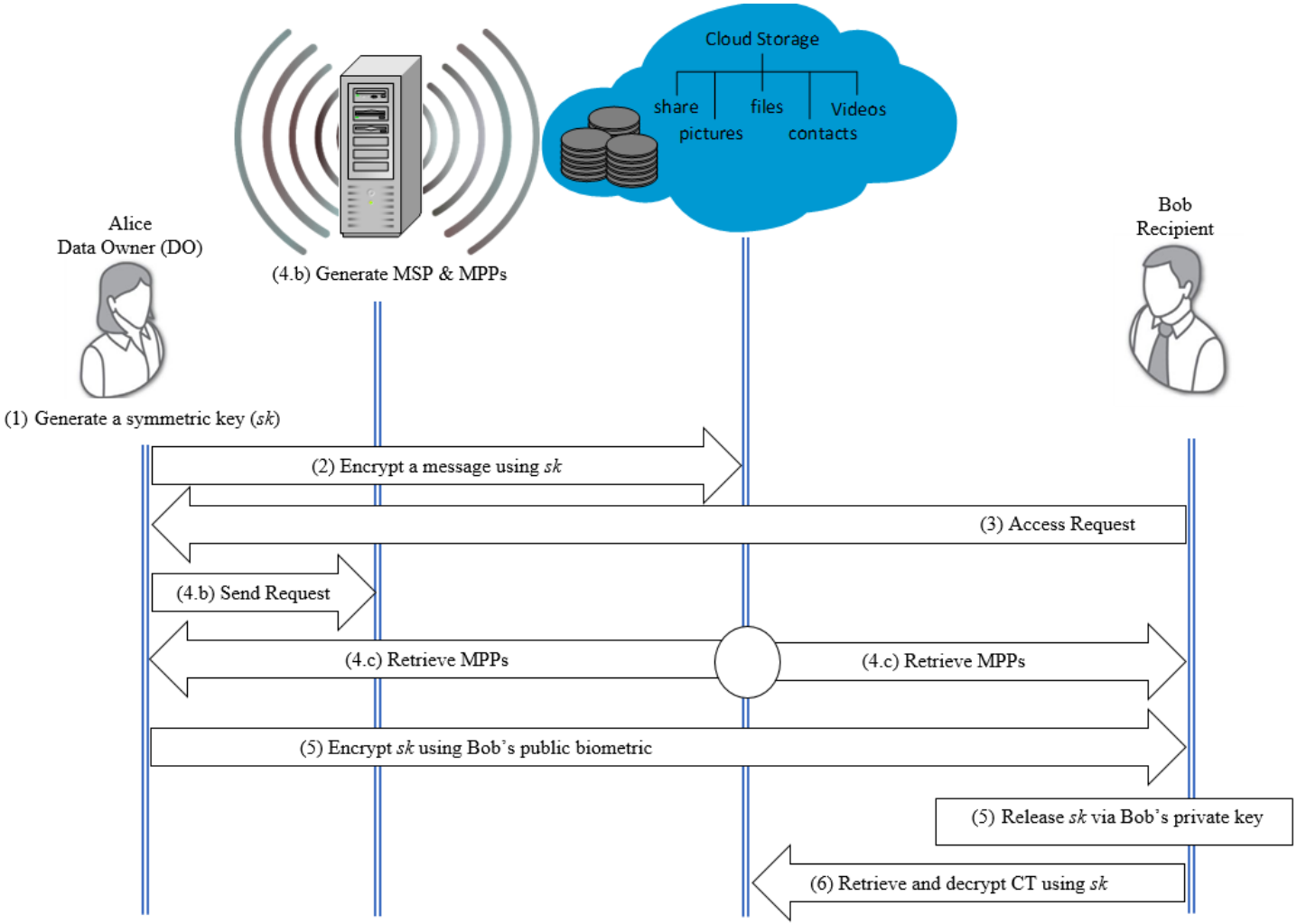


Figure 7.1: An overview of general KE--BIBC framework

7.3 KE-BIBC Implementation Details

Our protocol relies on the concept of fuzzy identity-based cryptosystems scheme proposed in [59] to bind the encryption keys with users' identity instead of using certificate authorities. The proposal has four main stages (*setup, Key extraction, Encryption and Decryption*) to implement the key exchange KE-BIBC explained in the previous section.

Let $\mathbb{G}_0$ be a bilinear group of prime order *p*, and $< g >$ be a generator of $\mathbb{G}_0$. Let also *e* be a bilinear map such that: $e: \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. In our proposal, each identity consists of a set of *n* strings of an arbitrary length. The collision-resistant hash function [59] is selected to convert each string in the identity into the corresponding integer in $\mathbb{Z}_p$. Eventually, the Lagrange coefficient $\Delta_{i,S}$is defined for $i \in \mathbb{Z}_p$ and a set of element S in $\mathbb{Z}_p$as follows:

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$$

As explained above, Alice generates an encryption key *sk* in order to encrypt her message *M* before storing it in a public cloud computing. The following four steps explain the four main steps of the protocol:

- ***Setup (n, d).*** PKG is responsible for the full implementation of this algorithm. We assume that Alice has Bob's public identity (i.e., Bob's face image), then she sends a request to a PKG to generate public and private parameters.
  - The elements $g_1 = g^y, g_2$ are chosen from$\mathbb{G}_1$
  - Picks uniformly at random $t_1, \dots, t_{n+1}$ from $\mathbb{G}_1$ **,** where *n* is the length of the identity.
  - Picks uniformly at random y in $\mathbb{Z}_p$.
  - Let *N* be the set $\{1, \dots, n+1\}$ and we define a function *T* as:

$$T(x) = g_2^{x^n} \prod_{i=1}^{n+1} t_i^{\Delta\, i,N(x)}$$

  The public parameters will be $g_1, g_2, t_1, \dots, t_{n+1}$ while *y* represents the master secret key *MSP*.

- **Key extraction.** The key extraction algorithm uses to generate a private key sk_{id} of Bob's identity *w* in order to decrypt the *sk.* As a verification stage, which precedes the process of giving Bob his decryption key, Bob needs to authenticate himself

using the proposed OTCR-MFA introduced in chapter 4.The process of extracting the private key components for identity *w* is as follows [59].

A random (d-1)-degree polynomial p is chosen with the constraint that all values at point zero equal to *MSP*, i.e. $p(0) = y$. The private key consists of two parts $\{D_i\}$and $\{d_i\}$ for each $i \in w$ so that

$D_i = g_2^{p(i)} T(i)^{r_i}$, and $d_i = g^{r_i}$ where r_i is randomly chosen from $\mathbb{Z}_p$ for each $i \in w$.

- **Encryption.** Alice encrypts the *sk* before sending it to Bob. To encrypt the $sk \in \mathbb{G}_2$using the public key of Bob's identity *w'*, she chooses random $k \in \mathbb{Z}_p$. The ciphertext consists of four parts:

$$CT = (w', E' = sk.e(g_1, g_2)^k, E'' = g^k, \{E_i = T(i)^k\}_{i \in w'})$$

- **Decryption.** Now, Bob needs to present his identity *w* (i.e., his fresh biometric) to get the corresponding private key of *w*. The decryption algorithm includes the following procedure: Assuming that the $\mathcal{E}_{sk}$ represents the encrypted *sk*, which was encrypted using the public key of identity *w'*, then another key of identity *w* would be able to decrypt the $\mathcal{E}_{sk}$ if and only if $|w \cap w'| \geq d$. If the overlapping between w and w' satisfies the threshold value *d*, then an arbitrary subset *S* of *d-elements* would be enough to decrypt *CT*, where *S* is a subset of $w \cap w'$. The symmetric key *sk* can be reconstructed using the following decryption formula:

$$sk = E' \prod_{i \in S} \left(\frac{e(d_i, E_i)}{e(D_i, E'')} \right)^{\Delta_{i,S}(0)}$$

7.3.1 Proving the Correctness of the Proposed Protocol.

The following steps prove the correctness of KE-BIBC:

$$sk = E' \prod_{i \in S} \left(\frac{e(d_i, E_i)}{e(D_i, E'')} \right)^{\Delta_{i,S}(0)}$$

$$= sk.e(g_1, g_2)^k \prod_{i \in S} \left(\frac{e(g^{r_i}, T(i)^k)}{e\left(g_2^{p(i)} T(i)^{r_i}, g^k\right)} \right)^{\Delta_{i,S}(0)}$$

$$= sk.e(g_1, g_2)^k \prod_{i \in S} \left(\frac{e(g^{r_i}, T(i)^k)}{e(g_2^{p(i)}, g^k).e(T(i)^{r_i}, g^k)} \right)^{\Delta_{i,S}(0)}$$

$$= sk.e(g_1, g_2)^k \prod_{i \in S} \left(\frac{e\left(g, T(i)\right)^{r_i k}}{e(g_2^{p(i)}, g^k).e\left(g, T(i)\right)^{r_i k}}\right)^{\Delta_{i,S}(0)}$$

by cancelling $e(g, T(i))^{r_i k}$

$$= sk.e(g_1, g_2)^k \prod_{i \in S} \frac{1}{e(g_2^{p(i)}, g^k)^{\Delta_{i,S}(0)}}$$

by interpolating the exponents, and since $p(0) = y$ using *d* points, the result be:

$$= sk.e(g, g_2)^{ky} \prod_{i \in S} \frac{1}{e(g_2, g)^{ky}} = sk$$

Where the *sk* is used to decrypt the encrypted data stored in public cloud computing. On the other hand, the proposed KE-BIBC is secure against Fuzzy-Selective Identity attack model [59] has been described in details chapters 3 and 4; this is why we will not discuss this model of attack again in this chapter

7.4 Chapter Summary

This chapter proposed a new protocol for key exchange using biometric identity-based cryptosystems (KE-BIBC) which enables parties, Alice and Bob, to securely exchange cryptographic keys even when an adversary is monitoring the communication channel. It has been shown that the proposed protocol combines biometrics with IBC in order to provide a secure way to access symmetric keys based on the identity of the users in the unsecured environment. It supports DOs by providing an appropriate access control mechanism by giving them the power of fine-grained sharing of encrypted data by controlling who can access their data.

In the proposed KE- BIBC protocol, the message is first encrypted by the data owner (Alice) using a traditional symmetric key before migrating the data to cloud storage. The symmetric key is then encrypted using public biometrics of the users selected by the data owner to decrypt the message based on fuzzy identity-based cryptosystems. We showed that only selected users (Bob as an example) were able to decrypt the message by providing a fresh sample of their biometric data. We argued that the proposed solution could eliminate the needs for a key distribution centre in traditional cryptography but

more importantly is to give Alice as a data owner the power of fine-grained sharing of encrypted data by controlling who can access her data.

Chapter 8

Conclusion and Future Work

This chapter summarises our contributions and explains how they serve in addressing the overall aim of the thesis in improving the security and privacy of IBCs by giving DOs much more control over their data in addition to solving the users' verification issue in before releasing the decryption keys.

8.1 Summary

The rapid growth of internet technologies (software and hardware) in recent years has led to making cloud Computing the most commonly used IT model for different clients (users, companies, and government agencies) by offering on-demand (paid as needed), inexpensive and IT services. This innovative model has radically revolutionised the IT industries' management by allowing new host multi-tenant plans. Despite these exciting offers, clients of Cloud Computing are facing some serious challenges.

In this thesis, we investigated the challenge associated with the adoption of Cloud-based data/ file storage and argued that once the clients upload their data to be hosted in Cloud Computing, the owners of the data/files will lose their control over them. The thesis provided a reflection on current best practices to protect the data by encrypting it before it is stored in the cloud computing environment to ensure that the data will not be exploited or manipulated by the providers/ or attackers.

Chapter 3 showed that traditional PKEs has an outstanding challenge related to keys management, e.g. creation, distribution, and storage cryptographic keys. We argued that F-IBCs (including IBCs) are a promising public key model proposed as alternative to PKEs in order to mitigate the burden related to traditional PKEs by introducing users' identities, e.g. email address, diver licenses, passport number, as well as public biometrics data, in the process of generating their corresponding encryption/decryption keys.

Despite the great potentials of F-IBCs, the thesis highlighted a range of security and privacy vulnerabilities. We then presented a number of contributions to address the limitation and unleash the full potentials of FBCs to be used in an open environment such as cloud computing. Herein, we provide a summary of the contributions.

1. Initially, we noted that the verification mechanisms used in existing F-IBCs for the delivery of decryption keys rely on impractical assumption. It was assumed that an impersonation attack cannot deceive PKG and get the decryption key of a genuine user. The thesis argued that social media platforms (e.g., Facebook, Instagram, WhatsApp, Twitter, LinkedIn, and Snapchat) are an accessible source that allows attackers to gain public biometric data (e.g., face image). Therefore, the adoption of F-IBCs on public biometrics/ identities in the release of decryption keys make them vulnerable to imposter attacks.

   To tackle this problem, we proposed a one-time challenge-response multifactor authentication, which is a hybrid approach that blends a user's biometric with cancellable biometrics relying on a user-based transformation to improve the security as well as the privacy of existing F-IBCs. The proposed scheme increases the difficulty for imposters to deceive the PKG in order to retrieve the decryption keys of any genuine user in F-IBCs. The proposal has two main stages: the enrolment and authentication stages. In our Experiments, we implemented a face recognition system and presented the results based on the ORL database to demonstrate the feasibility of the proposal.

2. The thesis argued that the way that the decryption keys in existing F-IBCs are managed gives the PKG full control of the encrypted data, with very little or no control from DOs. For that reason, we proposed a new decryption key generation based on engaging both the PKG and the DOs. To further improve the control of DOs, we proposed a new key-validity method based on exploiting Shamir Secret Sharing technique so that DOs can decide on the validity and the expiry of the decryption keys.

3. The thesis investigated the security of the MSPs of existing F-IBCs and highlighted two serious security vulnerabilities, namely PKG-based key escrow problem and the central point of attacks. Therefore, our third contribution is the proposal of binding the MSP to the users' biometric data so that the power is completely shifted from the

PKG to DOs. For error-tolerance that accompanies the discrepancy that may occur between features vectors presented in enrolment and verification stages, we use the Reed Solomon error correction code. In this our Experiments, we implemented Face and Fingerprint recognition systems that utilised the ORL and FVC2002_Db2_a database, respectively. Both results confirm the viability of the proposal.

4. The last proposed solution in Chapter 7 aims to introduce a new application of fuzzy identity-based cryptosystems (F-IBCs) by adopting it as a protocol in the exchange of symmetric keys. Typically, how to securely store and exchange the keys between two parties over open networks particularly in the open environment such as cloud computing are the main problem associated with using symmetric/ asymmetric. Currently, available key exchange protocols depend on employing trusted couriers or secure channels, which can be subject to a man-in-the-middle attack and various other attacks. We argued that the proposed key exchange-based biometric identity-based cryptosystem (KE-BIBC) permits parties to securely exchange cryptographic keys even an adversary is observing the communication channel between them.

8.2 Future research

Our future research will focus on further improving IBCs as a promising public key encryption model. This can be summarised as follows.

1. Using multi-modal biometric to bind the MSP to the user biometrics. Thus, to release the MSP, it requires each success in the verification process of two levels (in case of two modal biometric) instead of one level; this increases the difficulty of non-genuine users' capability to impersonate the PKG.
2. Investigating a new scheme of IBC in which MSPs are issued by DOs where only the MPPs are generated by PKG. This work aims to move control over issuing the decryption key to DOs instead of PKG to further prevent the central point of attacks as well as any possibility of manipulating the data on the PKG side as illustrated in Figure 8.1.

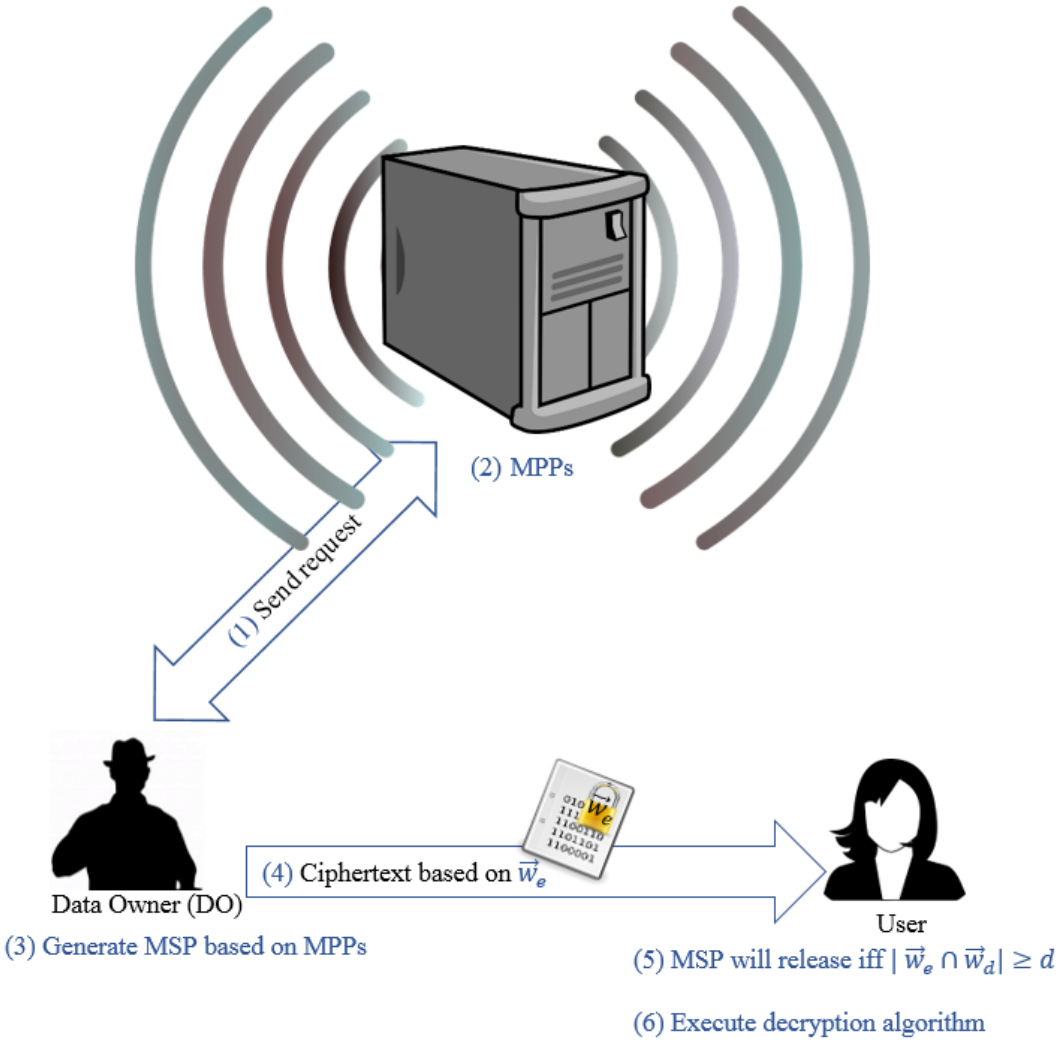


Figure 8.1: Shifting the establishment of MSP from DOs to PKG in the proposed future work

References

[1] M. D'Angelo and business.com writer, "The Best Cloud Computing Services of 2019," *business.com*, 2018. [Online]. Available: https://www.business.com/categories/cloud-computing-services/. [Accessed: 03-Dec-2019].

[2] T. Chou, "Security Threats on Cloud Computing Vulnerabilities," vol. 5, no. 3, pp. 79–88, 2013.

[3] T. E. Kezia, "The state of cloud storage in five charts," *IT PRO*. [Online]. Available: https://www.itpro.co.uk/cloud-storage/30599/the-state-of-cloud-storage-in-five-charts. [Accessed: 06-Nov-2018].

[4] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," *Proc. - 6th Int. Conf. Semant. Knowl. Grid, SKG 2010*, no. December, pp. 105–112, 2010.

[5] D. Riley, "Cloud data management firm Veeam exposes 200GB of data on AWS instance," *Silicon Angle*, 2018. [Online]. Available: https://siliconangle.com/2018/09/11/cloud-data-management-firm-veeam-exposes-200gb-data-aws-instance/. [Accessed: 06-Nov-2018].

[6] J. Rogers, "Dropbox data breach: 68 million user account details leaked," FOX News Network, USA, 2016.

[7] Identity Theft Resource Center (ITRC), "2017 Annual Data Breach Year-End Review," 2018.

[8] Identity Theft Resource Center, "Data breach reports 2018," *Identity Th. Resour. Cent.*, vol. 18, no. 9, 2018.

[9] Identity Theft Resource Center, "Data breach reports 2019," 2019.

[10] A. G. Rajkumar Buyya, James Broberg, *CLOUD COMPUTING:Principles and Paradigms*, no. 139. Hoboken, New Jersey: John Wiley & Sons, INC., 2014.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1–5.

[12] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over

encrypted data in cloud computing," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011, pp. 383–392.

[13] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 12, pp. 1947–1960, 2013.

[14] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," *Proc. - IEEE INFOCOM*, vol. 2016-July, 2016.

[15] B. Wang, W. Song, W. Lou, and Y. T. Hou, "Privacy-preserving pattern matching over encrypted genetic data in cloud computing," *Proc. - IEEE INFOCOM*, 2017.

[16] Z. Fu, L. Xia, X. Sun, A. X. Liu, and G. Xie, "Semantic-Aware Searching over Encrypted Data for Cloud Computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 9, pp. 2359–2371, 2018.

[17] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005, pp. 457–473.

[18] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[19] D. E. Denning and D. K. Branstad, "A taxonomy for Key Escrow Encryption Systems," 1996.

[20] H. Abelson *et al.*, "The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption," *World Wide Web J.*, vol. 2, no. 3, pp. 241–257, 1997.

[21] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 196 LNCS, pp. 47–53, 1985.

[22] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, 2004.

[23] E. A. Rua, E. Maiorana, J. L. A. Castro, and P. Campisi, "Biometric template protection using universal background models: An application to online signature," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 1 PART 2, pp. 269–282, 2012.

[24] W. Jansen, T. Grance, and P. D. Gallagher, "Guidelines on Security and Privacy

in Public Cloud Computing," 2011.

[25] S. Dustdar, T. U. Wien, N. Patrignani, and I. Kavathatzopoulos, "Cloud computing," *ACM SIGCAS Comput. Soc.*, vol. 45, no. 3, pp. 68–72, 2016.

[26] J. Rittinghouse and J. Ransome, *Cloud computing\nImplementation, Management, and Security*. 2016.

[27] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Secur. Priv.*, vol. 8, no. 6, pp. 24–31, 2010.

[28] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing," *Commun. Surv. Tutorials, IEEE*, vol. 15, no. 2, pp. 843–859, 2013.

[29] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," in *Advances in Cryptology--Crypto 2004*, 2004, pp. 443–459.

[30] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *Advances in Cryptology-CRYPTO 2006*, Springer, 2006, pp. 290–307.

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

[32] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography*, Springer, 2007, pp. 515–534.

[33] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *Parallel Distrib. Syst. IEEE Trans.*, vol. 25, no. 1, pp. 222–233, 2014.

[34] C. Paar and J. Pelzl, *Understanding Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg.https://doi.org/10.1007/978-3-642-04101-3, 2010.

[35] W. M. H. Company, *Modern Cryptography: Theory and Practice*, vol. 170, no. 2. 2003.

[36] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field," *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1639–1646, 1993.

[37] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[38] V. S. Miller, “Use of elliptic curves in cryptography,” in *Conference on the theory and application of cryptographic techniques*, 1985, pp. 417–426.

[39] F. Li, “Context-Aware Attribute-Based Techniques for Data Security and Access Control in Mobile Cloud Environment,” City University London, 2015.

[40] V. C. Hu *et al.*, “Guide to Attribute Based Access Control (ABAC) Definition and Considerations,” 2014.

[41] R. Shirey, *Internet Security Glossary, Version 2*, 2nd ed. The IETF Trust, 2007.

[42] A. AlZayer, “system access control,” *Courses Resources*, 2013. [Online]. Available: http://coursesdocs.blogspot.co.uk/2013/09/system-access-control.html. [Accessed: 15-Jun-2016].

[43] F. G. G. Meade, “a Guide To Understanding Discretionary Access Control in,” *Control*, no. September, 1987.

[44] jimmyxu101, “Discretionary Access Control vs Mandatory Access Control.” 2014.

[45] NIST Special Publication 800-53, “Security and Privacy Controls for Federal Information Systems and Organizations Security and Privacy Controls for Federal Information Systems and Organizations,” *Sp-800-53Ar4*, p. 400+, 2013.

[46] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli., *Role-Based Access Control*, 2nd Editio. ARTECH HOUSE, INC., 2007.

[47] X. Jin, R. Krishnan, and R. Sandhu, “A unified attribute-based access control model covering DAC, MAC and RBAC,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7371 LNCS, pp. 41–55, 2012.

[48] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-Based Access Control Models yz 1 INTRODUCTION,” vol. 29, no. 2, pp. 38–47, 1996.

[49] D. F. Ferraiolo and D. R. Kuhn, “Role-based access controls,” *arXiv Prepr. arXiv0903.2171*, 2009.

[50] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, “Attribute-Based Access Control,” no. February, pp. 85–88, 2015.

[51] V. C. Hu *et al.*, “Guide to attribute based access control (ABAC) definition and considerations (draft),” *NIST Spec. Publ.*, vol. 800, no. 162, 2013.

[52] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shah, “First Experiences using

XACML for Access Control in Distributed Systems," *XMLSEC '03 Proc. 2003 ACM Work. XML Secur.*, pp. 25–37, 2003.

[53] L. I. Repository, "City , University of London Institutional Repository," vol. 21, pp. 381–395, 2015.

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

[55] R. Canetti, S. Halevi, and J. Katz, "A Forward-Secure Public-Key Encryption Scheme," vol. 20, no. 3, pp. 265–294, 2007.

[56] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *Advances in Cryptology-EUROCRYPT 2004*, 2004, pp. 223–238.

[57] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[58] A. Burnett, A. Duffy, and T. Dowling, "A Biometric Identity Based Signature Scheme," vol. 5, no. 3, pp. 317–326, 2004.

[59] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005, pp. 457–473.

[60] N. D. Sarier, "Multimodal biometric Identity Based Encryption," *Futur. Gener. Comput. Syst.*, vol. 80, pp. 112–125, 2018.

[61] F. Guo, W. Susilo, and Y. Mu, "Distance-Based Encryption: How to Embed Fuzziness in Biometric-Based Encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 247–257, 2016.

[62] M. Pirretti, P. Traynor, P. Mcdaniel, and B. Waters, "Secure Attribute-Based Systems," 2006.

[63] J. Baek, W. Susilo, and J. Zhou, "New constructions of fuzzy identity-based encryption," p. 368, 2007.

[64] K. W. Bowyer, P. Yan, K. I. Chang, P. J. Flynn, E. Hansley, and S. Sarkar, "Multi-modal biometrics: an overview," no. September, pp. 1221–1224, 2006.

[65] S. Ma, "Identity-based encryption with outsourced equality test in cloud

computing," *Inf. Sci. (Ny).*, vol. 328, pp. 389–402, 2016.

[66] C. Gentry and A. Silverberg, "Hierarchical ID-Based Cryptography," pp. 1–21, 2002.

[67] N. D. Sarier, "A new biometric identity based encryption scheme," in *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, 2011, no. December 2009, pp. 23–32.

[68] N. Dottling and S. Garg, "Identity-Based Encryption from the Diffie-Hellman Assumption," *Crypto*, pp. 537–569, 2017.

[69] O. Blazy *et al.*, *(Hierarchical) Identity-Based Encryption from Affine Message Authentication To cite this version : (Hierarchical) Identity-Based Encryption from Affine Message Authentication*. 2016.

[70] D. Boneh and X. Boyen, "Efficient Selective Identity-Based Encryption Without Random Oracles," vol. 24, no. 4, pp. 659–693, 2011.

[71] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 54–65, 2015.

[72] B. Choudhury, P. Then, B. Issac, V. Raman, and M. K. Haldar, "A Survey on Biometrics and Cancelable Biometrics Systems," *Int. J. Image Graph.*, vol. 18, no. 01, p. 1850006, 2018.

[73] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, "Security and accuracy of fingerprint-based biometrics: A review," *Symmetry (Basel).*, vol. 11, no. 2, 2019.

[74] H. Al-Assam, "Entropy evaluation and security measures for reliable single/multi-factor biometric authentication and biometric keys," University of Buckingham, 2013.

[75] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2010.

[76] A. Adler, "Vulnerabilities in Biometric Encryption Systems," pp. 1100–1109, 2005.

[77] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," *Proc. 1994 IEEE Work. Appl. Comput. Vis.*, pp. 138–142, 1994.

[78] N. Hazim, S. Sameer, W. Esam, and M. Abdul, "Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 371–377, 2016.

[79] R. C. Fourier and E. Cedex, "Three factor scheme for Biometric-based cryptographic key regeneration using Iris," 2008.

[80] R. Canetti, S. Halevi, and J. Katz, "A Forward-Secure Public-Key Encryption Scheme," in *Advances in Cryptology-Crypto 2003*, Springer, 2003, pp. 255–271.

[81] D. A. N. Boneh and M. Franklin, "Downloaded 12 / 27 / 12 to 138 . 26 . 31 . 3 . Redistribution subject to SIAM license or copyright ; see http://www.siam.org/journals/ojsa.php," vol. 32, no. 3, pp. 586–615, 2003.

[82] H. Al-Assam and S. A. Jassim, "Multi-Factor Challenge/Response Approach for Remote Biometric Authentication," *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 8063, pp. 80630V-80630V–11, 2011.

[83] V. Revuelto and K. Socha, "Weaknesses in Diffie-Hellman Key Exchange Protocol," pp. 0–7, 2016.

[84] A. M. Bazen and R. N. J. Veldhuis, "Likelihood-Ratio-Based Biometric Verification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 86–94, 2004.

[85] J. Chen, H. W. Lim, S. Ling, H. Wang, and K. Nguyen, "Revocable Identity-Based Encryption from Lattices," pp. 390–403, 2012.

[86] J. H. Seo and K. Emura, "Efficient delegation of key generation and revocation functionalities in identity-based encryption," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7779 LNCS, pp. 343–358, 2013.

[87] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7778 LNCS, no. Pkc, pp. 216–234, 2013.

[88] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 425–437, 2015.

[89] S. Park, K. Lee, and D. H. Lee, "New constructions of revocable identity-based encryption from multilinear maps," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1564–1577, 2015.

[90] D. Chang, A. K. Chauhan, S. Kumar, and S. K. Sanadhya, "Revocable identity-based encryption from codes with rank metric," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10808 LNCS, pp. 435–451, 2018.

[91] G. Chunpeng, Z. Liu, J. Xia, and F. Liming, "Revocable Identity-Based Broadcast Proxy Re-encryption for Data Sharing in Clouds," *IEEE Trans. Dependable Secur. Comput.*, vol. PP, no. c, pp. 1–1, 2019.

[92] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," p. 417, 2008.

[93] D. Boneh, X. Boyen, and E. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," pp. 440–456, 2005.

[94] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3621 LNCS, no. 1, pp. 258–275, 2006.

[95] D. Boneh, B. Waters, and M. Zhandry, "Low overhead broadcast encryption from multilinear maps," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8616 LNCS, no. PART 1, pp. 206–223, 2014.

[96] P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor, "Ranksign: An efficient signature algorithm based on the rank metric," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8772, pp. 88–107, 2014.

[97] M. Co., "Key Distribution Center," *Microsoft*, 2016. [Online]. Available: https://msdn.microsoft.com/en-gb/enus/library/windows/desktop/aa378170(v=vs.85).aspx. [Accessed: 05-May-2017].

[98] W. Diffie and M. E. Hellman, "New Directions in Cryptography Invited Paper," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[99] V. Antonþiü, "Key Management in Identity Based Encryption schemes," pp. 1235–1239, 2010.

[100] S. Kwidama and T. Hassanmahomed, “Cryptographic Key Management,” *Univ. Amsterdam*, 2009.

[101] K. Paterson and G. Price, “A comparison between traditional Public Key Infrastructures and Identity-,” *Isg.Rhul.Ac.Uk*, vol. 8, no. 3, pp. 57–72, 2003.

[102] J. W. Lyons, “Key Management Using ANSI X9.17,” 1987.

[103] J. Bethencourt *et al.*, “Ciphertext-Policy Attribute-Based Encryption To cite this version : HAL Id : hal-01788815 Ciphertext-Policy Attribute-Based Encryption,” no. May 2007, 2018.

[104] H. Cheng, C. Rong, M. Qian, and W. Wang, “Accountable Privacy-Preserving Mechanism For Cloud Computing Based On Identity-Based Encryption,” *IEEE Access*, vol. PP, no. 4, pp. 1–1, 2018.

[105] Z. Cheng, R. Comley, and L. Vasiu, “Remove key escrow from the Identity-based encryption system,” 2004.

[106] Q. Wei, F. Qi, and Z. Tang, “Remove key escrow from the BF and Gentry identity-based encryption with non-interactive key generation,” *Telecommun. Syst.*, pp. 22–24, 2018.

[107] T. Document and P. Minimum, “Instruction for national security systems public key infrastructure x.509 certificate policy under CNSS policy no. 25,” vol. 2014, no. 1300, 2014.

[108] X. Li, T. Xiang, F. Chen, and S. Guo, “Efficient biometric identity-based encryption,” *Inf. Sci. (Ny).*, vol. 465, pp. 248–264, 2018.

[109] G. V. Respati, S. Tinggi, and S. Negara, “Secure Key Issuing in ID-based Cryptography,” vol. 32, no. Aisw, 2015.

[110] G. Sumalatha, “Analyzing the Key Escrow Problem in Identity Based Cryptography and Security Notions of Key Management Techniques,” vol. 6, no. 11, pp. 537–543, 2017.

[111] Y. Li, F. Qi, and Z. T. B, “Security, Privacy, and Anonymity in Computation, Communication, and Storage,” vol. 10658, pp. 108–120, 2017.

[112] R. Boussada, M. E. Elhdhili, and L. A. Saidane, “Toward privacy preserving in IoT e-health systems: A key escrow identity-based encryption scheme,” *2018 15th IEEE Annu. Consum. Commun. Netw. Conf.*, pp. 1–7, 2018.

[113] L. Chen, K. Harrison, D. Soldera, and N. Smart, "Applications of Multiple Trust Authorities in Pairing Based Cryptosystems We investigate a number of issues related to the use of multiple Applications of Multiple Trust Authorities in," 2003.

[114] J. H. Oh, K. K. Lee, and S. J. Moon, "How to solve key escrow and identity revocation in identity-based encryption schemes," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3803 LNCS, pp. 290–303, 2005.

[115] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Secure key issuing in ID-based cryptography," *Proc. Second Work. Australas. Inf. Secur. Data Min. Web Intell. Softw. Int. 32*, no. Gentry, pp. 69–74, 2004.

[116] V. Goyal, "Reducing Trust in the PKG in Identity Based Cryptosystems," *Crypto '07*, pp. 1–18, 2007.

[117] S. Chow, "Removing escrow from identity-based encryption: New security notions and key management techniques," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5443, pp. 256–276, 2009.

[118] I. T. Union, "Biometrics and Standards," *ITU-T Technol. Watch Reports*, pp. 1–22, 2009.

[119] C. Rathgeb and A. Uhl, "A Survey on Biometric Cryptosystems," pp. 1–25, 2011.

[120] A. Juels and M. Sudan, "A fuzzy vault scheme," *Des. Codes, Cryptogr.*, vol. 38, no. 2, pp. 237–257, 2006.

[121] Z. Jin, A. B. J. Teoh, B. M. Goi, and Y. H. Tay, "Biometric cryptosystems: A new biometric key binding and its implementation for fingerprint minutiae-based representation," *Pattern Recognit.*, vol. 56, pp. 50–62, 2016.

[122] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," *Proc. 6th ACM Conf. Comput. Commun. Secur. - CCS '99*, pp. 28–36, 1999.

[123] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," *Proc. - Int. Conf. Pattern Recognit.*, vol. 16, no. 3, pp. 811–814, 2002.

[124] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 846–859, 2000.

[125] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud

computing," *Proc. - 2012 Int. Conf. Comput. Sci. Electron. Eng. ICCSEE 2012*, vol. 1, no. 973, pp. 647–651, 2012.

[126] P. Partheeban and V. Kavitha, "A study with security concerns in service delivery models of cloud computing," *Int. J. Appl. Eng. Res.*, vol. 10, no. 21, pp. 42219–42230, 2015.

[127] Microsoft, "Key Distribution Center," *msdn.microsoft.com*, 2016. [Online]. Available: https://msdn.microsoft.com/en-gb/enus/library/windows/desktop/aa378170(v=vs.85).aspx. [Accessed: 05-May-2017].