AN EFFICIENT FRAMEWORK FOR DATA SECURITY IN MOBILE CLOUD COMPUTING PARADIGM

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

DOCTOR OF PHILOSOPHY

by

Tarunpreet Bhatia

(951403001)

Under the guidance of:

Dr. Anil Kumar Verma

Professor, Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA – 147004

July 2018

Dedicated To

The Almighty and My parents

Certificate

I hereby certify that the work being presented in this thesis entitled "**An Efficient Framework for Data Security in Mobile Cloud Computing Paradigm**", for the award of degree of "Doctor of Philosophy" submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Anil Kumar Verma and refers other researchers works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Tarunpreet Bhatia

(Tarunpreet Bhatia)

951403001

Place: Patiala

Date: 25 · 7 · 2018

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Approved by:

Anil K. Verma

(Dr. Anil Kumar Verma)

Professor

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala, Punjab, INDIA

Scanned by CamScanner

Acknowledgements

First of all, I would like to express a deep sense of gratitude to the Almighty, Who blessed me with the zeal and enthusiasm to complete this work successfully.

I am extremely thankful to my supervisor, Dr Anil Kumar Verma, Professor, Thapar Institute of Engineering and Technology, for his constant guidance, help and continuous support. I am grateful to him for motivating, inspiring and supporting me throughout the life cycle of my PhD work. His enthusiasm has been a constant source of inspiration for me to attain my goal. Especially, the extensive comments, discussions and the interactions with him had a direct impact on the final form and quality of my PhD work.

I am also thankful to Dr Maninder Singh, Head of Computer Science and Engineering Department, for his motivation, kind help and unstinted support. I am thankful to my Doctoral committee members, Dr Neeraj Kumar, Dr Shalini Batra and Dr Kulbir Singh for their constructive comments and ensuring the progress of my research work in the right direction. I would also like to thank Prof. P. Gopalan (Director, Thapar Institute of Engineering and Technology) and Prof. O.P. Pandey (Dean, Research and Sponsored Projects) for providing me all the research facilities which have been immensely helpful for the completion of my work. I wish to thank the faculty and staff members of Computer Science and Engineering Department for their co-operation and support.

This thesis would not have been possible without the support of my family. I offer the deepest gratitude to my grandfather Late. S. Harbans Singh Narang; my loving father, S Tejinder Singh Bhatia, whose love and affectionate blessings have been a constant source of inspiration in making my vision a reality. I am also thankful to my mother, Mrs Arvinder Bhatia and my brother, Mr Harmanpreet Bhatia for their untiring love, affection, motivation and blessings. I also acknowledge the cooperation and encouragement extended to me by my friends, especially Dr Karamjit Kaur, Dr Gaurav Sharma and Ms Neha Arora.

Abstract

The incessant spurt of research activities to augment the capabilities of resource-constrained mobile devices by leveraging heterogeneous cloud resources has created a new research impetus called Mobile Cloud Computing (MCC). However, this rapid relocation to the cloud has fueled security and privacy concerns as users' data leave the owners' protection sphere and enter the cloud. Cryptography is a conventional way of maintaining the privacy of data and authenticating users over an insecure network. Traditional Public Key Cryptography (PKC) is more expensive than symmetric cryptography as it requires the certificate for authentication. The scope for identity-based cryptography is no doubt high as it authenticates the sender without relying on certificates, but it suffers from the key escrow problem. Certificateless cryptography is a relatively new and attractive paradigm that provides the security against key escrow problem and solves certificate management problem in traditional PKC.

This work has proposed three different schemes for data security in MCC namely Certificateless Proxy Signcryption (CLPSc), Certificateless Proxy Re-encryption (CL-PRE) and Certificateless Incremental Proxy Re-encryption (CL-iPRE).

CLPSc is one of the most efficient security primitives for secure data transmission. The entrustment of signing rights to a proxy signcrypter at the behest of an original signcrypter imparts utility in various fields such as an online proxy auction, healthcare industry, ubiquitous computing, etc. Unlike the traditional sign-then-encrypt approach, signcryption primitive saves computational costs and bandwidth load. Recently, a pairing free CLPSc scheme has been proposed which has claimed to be secure against forgery under adaptive chosen-message attacks. It has been proved that it has failed to provide unforgeability. As an improvement to the scheme, a pairing-free CLPSc scheme has been proposed for e-prescription system in MCC. The proposed scheme is proven to be secure against indistinguishability under adaptive chosen-ciphertext attack and existential forgery under adaptive chosen-message attack in the random oracle model. The proposed scheme outperforms the existing schemes in terms of computational efficiency.

The ubiquitous and timely access to electronic health records (EHRs) and personal health records (PHRs) help physicians to take critical care decisions and save lives. Cloud computing has a potential to provide ubiquitous and on-demand instant access to common pool of shared resources and services to various stakeholders who are involved in e-healthcare industry. A Proxy Re-encryption (PRE) is a cryptographic scheme for delegation of decryption rights. The PRE schemes are useful in the scenarios where data are desired to be shared with the authorized users over the cloud. The cryptanalysis of recently proposed CL-PRE scheme has been performed. Further, a lightweight, pairing-free, single-hop unidirectional CL-PRE scheme has been proposed for secure sharing of mobile PHRs with public cloud. In CL-PRE, patients encrypt the data with their public keys before outsourcing to cloud and cloud resident semi-trusted proxy further re-encrypts into ciphertext under intended recipient's public key without learning anything about encrypted message. The security has been proved through formal analysis against chosen ciphertext attack in the random oracle model. The proposed CL-PRE scheme is more efficient and suitable for low power mobile devices in comparison to existing schemes.

Conventionally, any modification to uploaded record irrespective of its scale or frequency would compel the mobile client to encrypt and compute the hash value from the scratch. Such operations would drain battery and energy resources of mobile devices. One of the crucial factors for enabling fast and secure computations in the Zettabyte era is the use of the incremental cryptographic algorithm which runs in time proportional to the number of modifications instead of the document length. An efficient elliptic curve based pairing-free CL-iPRE scheme has been proposed for improving the file modification operations.

Keywords: Certificateless cryptography, Data security, Encryption, Incremental cryptography, Proxy signcryption, Proxy re-encryption

Table of Contents

Certificate i
Acknowledgements ii
Abstract iii
List of Figures viii
List of Tables ix
List of Abbreviations x
List of Symbols xiv
List of Publications xvi
Chapter 1 Introduction 1
1.1. Mobile Cloud Computing 2
1.1.1 MCC Architecture 4
1.1.2 MCC Applications 6
1.1.3 MCC Challenges 7
1.2 Security issues in MCC 9
1.2.1 Infrastructural issues 10
1.2.2 Privacy issues 13
1.2.3 Compliance Issues 14
1.3 Data Security in MCC 16
1.3.1 States of data 18
1.3.2 Security principles 19
1.3.3 Classification of data security schemes 20
1.3.4 e-health and m-health data 23
1.4 Thesis Contribution 27
1.4.1 Motivation 27
1.4.2 Research Objectives 29
1.4.3 Summary of Contribution(s) 30
1.5 Thesis Outline 32
1.6 Summary 33
Chapter 2 Literature Review 35
2.1 Cryptographic Techniques/Frameworks for MCC 35
2.1.1 Symmetric Cryptography 35

2.1.2 Asymmetric Cryptography 39
2.1.3 Hybrid Cryptography 53
2.1.4 Proxy re-encryption 55
2.1.5 Incremental cryptography 58
2.2 Cryptographic Preliminaries 60
2.2.1 Elliptic Curve Cryptography (ECC) 60
2.2.2 Computational Hardness Problems 62
2.3 Introduction to Certificateless Proxy Signcryption (CLPSc) 62
2.3.1 Formal model of CLPSc 63
2.3.2 Security model of CLPSc 65
2.4 Introduction to Certificateless Proxy Re-encryption (CL-PRE) 70
2.4.1 Formal model of CLPRE 70
2.4.2 Security model of CLPRE 72
Chapter 3 Certificateless Proxy Signcryption 81
3.1 Introduction 81
3.2 Related Work 82
3.3 Review of Yanfeng et al's scheme 84
3.4 Cryptanalysis of Yanfeng et al's scheme 87
3.5 Proposed CLPSc scheme 88
3.6 Security Analysis 93
3.7 Computational and Performance Analysis 103
3.8 Summary 105
Chapter 4 Certificateless Proxy Re-encryption 106
4.1 Introduction 106
4.2 Related Work 107
4.3 Review of Qin's et al scheme 109
4.4 Cryptanalysis of Qin's et al scheme 112
4.5 Proposed CLPRE scheme 113
4.6 Security Analysis 119
4.7 Computational and Performance Analysis 132
4.8 Summary 135
Chapter 5 Incremental Certificateless Proxy Re-encryption 136
5.1 Introduction 136
5.2 Related Work 138

5.3 Proposed CL-iPRE Scheme 140
5.4 Block Modification Operations 144
5.4.1 Incremental block updation operations 145
5.4.2 Incremental block insertion operations 146
5.4.3 Incremental block deletion operations 146
5.5 Results and Comparative analysis 147
5.6 Summary 151
Chapter 6 Conclusion and Future Scope 152
References 155

List of Figures

Figure 1.1: Cloud computing architecture 3
Figure 1.2: MCC architecture 4
Figure 1.3: Security issues in MCC 9
Figure 1.4: Security architecture in cloud 17
Figure 1.5: Possible states of data 18
Figure 2.1 Cryptography preliminaries and data security schemes in MCC 36
Figure 2.2: Symmetric cryptography 37
Figure 2.3: Asymmetric cryptography 40
Figure 2.4: PKI based cryptosystem 41
Figure 2.5: Identity based cryptosystem 45
Figure 2.6: Certificateless cryptosystem 52
Figure 2.7 Proxy re-encryption 56
Figure 2.8: Incremental cryptography 58
Figure 2.9: Point addition in elliptic curves 61
Figure 3.1: Overview of proposed CLPSc scheme for E-Prescription system in MCC 90
Figure 3.2: Comparison of existing schemes with proposed CLPSc scheme in terms of computation time 105
Figure 4.1: Architecture of cloud-based mPHRs sharing using proposed CL-PRE scheme 114
Figure 4.2: Comparison of existing schemes with proposed CL-PRE scheme in terms of computation time 135
Figure 5.1: Architecture of proposed CL-iPRE scheme for secure sharing of EHRs in MCC 141
Figure 5.2: Turnaround time while block updating operations on dataset (a) Updating 1 block (b) Updating 2 blocks (c) Updating 4 blocks 149
Figure 5.3: Energy consumption while block updating operations on dataset (a) Updating 1 block (b) Updating 2 blocks (c) Updating 4 blocks 150

List of Tables

Table 1.1: Comparison of cloud computing and MCC paradigm 5
Table 1.2: Summary of threats, associated attacks and mitigation techniques 15
Table 2.1: Comparable key sizes for equivalent security levels (in bits) 60
Table 3.1: Comparative analysis of different schemes 104
Table 4.1: Comparison of proposed scheme with other schemes 134
Table 5.1: Data set used for file modification operations 148

List of Abbreviations

Abbreviation	Description
ACL	Access Control List
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
AS	Authentication Server
BSS	Block-based Storage Scheme
CA	Certificate Authority
CCA	Chosen Ciphertext Attack
CIA	Confidentiality, Integrity, Authenticity
CL-iPRE	Certificateless Incremental Proxy Re-encryption
CL-PKC	Certificateless Public Key Cryptography
CLPRE	Certificateless Proxy Re-encryption
CLPSc	Certificateless Proxy Signcryption
CLSS	Certificateless Short Signature
CMReS	Cloud Manager-based Re-encryption Scheme
CS	Cryptographic Server
CSPs	Cloud Service Providers
COM	Centralized Owner Model
CoS	Coding based Scheme
CP-ABE	Ciphertext Policy Attribute-based Encryption
CReS	Cloud-based Re-encryption Scheme
DES	Data Encryption Standard

DNS	Domain Name Server
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie–Hellman
EC-CDH	Elliptic Curve Computational Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
EHRs	Electronic Health Records
EnS	Encryption based scheme
EUF-CMA	Existential forgery under adaptive Chosen-Message Attack
KP-ABE	Key Policy Attribute-based Encryption
HIBE	Hierarchical Identity-based Encryption
HIPAA	Health Insurance, Portability and Accountability Act
HMAC	Hash Message Authentication Code
IaaS	Infrastructure-as-a-Service
IBACC	Identity-based authentication protocol for cloud
IBE	Identity-based Encryption
IB-PRE	Identity-based Proxy Re-encryption
IND-CCA2	Indistinguishability of encryptions under adaptively chosen ciphertext attacks
IoT	Internet-of-Things
IND-CPA	Indistinguishability under Chosen-Plaintext Attack
I-PReS	Incremental Proxy Re-encryption
KGC	Key Generation Centre

L-DCG	Lightweight Dynamic Credential Generation
L-EDCG	Lightweight Enhanced Dynamic Credential Generation
LSD-ABAC	Lightweight Static and Dynamic Attributes Based Access Control
MAC	Message Authentication Code
MAC address	Media Access Control Address
MCC	Mobile Cloud Computing
MCDs	Mobile Communication Devices
MD	Message Digest
MNM	Mobility Node Model
mPHRs	Mobile Personal Health Records
MReS	manager based re-encryption
OOB	Out-of-Band
PaaS	Platform-as-a-Service
PDP	Public verifiable Data Possession
PE-AA	Privacy Enhanced Authentication and Authorization
PHRs	Personal Health Records
PKASSO	PKI based authentication based on Single Sign-on
PKC	Public Key Cryptography
PKG	Private Key Generator
PKI	Public Key Infrastructure
PP-CP-ABE	Privacy Preserving Cipher Policy Attribute Based Encryption
PP-DP-SS	Privacy Preserving Data Policy Sharing Service
PRE	Proxy Re-encryption

QoS	Quality of service
RSA	Rivest Shamir Adleman
SaaS	Software-as-a-Service
SAKE	Scalable Authenticated Key Exchange
SECO	Secure Collaborative
SeDaSC	Secure Data Sharing in Clouds
SDSM	Secure Data Service Mechanism
ShS	Sharing based scheme
SHA	Secure Hash Algorithm
SLA	Service Level Agreements
VTaPD	Virtual Trusted and Provisioning Domain
VM	Virtual Machine
WBAN	Wireless Body Area Network
WSDL	Web Services Description Language
XSS	Cross site scripting

List of Symbols

Symbol	Meaning
k	Security Parameter
E/F_p	Elliptic curve over prime finite field
G	Cyclic subgroup of the elliptic curve group
P	Generator of subgroup
q	Order
ψ	System parameters
s	Master secret key
P_{Pub}	Master public key
$\mathcal{M} = \{0,1\}^l$	Plaintext message space
$\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5$	Hash functions
ID_A	Identity A
S_A	Partial private key
K_A	Partial public key
k_A, k_{A_1}, k_{A_2}	Secret values of user A
Sk_A	Private key of user A
Pk_A	Public key of user A
$D_{A \to C}$	Delegation from user A to user C
$\mathcal{M}_W$	Message warrant
D_C	Proxy signing key
A	Sender/Data owner/Original signcrypter

C	Proxy signcrypter/Proxy server
B	Receiver
$\mathcal{M}$	Plaintext Message
$\mathcal{A}_1, \mathcal{A}_2$	Adversaries
$\mathbb{O}$	Oracle
$\mathbb{C}$	Challenger
$\mathbb{q}$	Oracle queries
$\mathcal{L}_{\mathcal{H}_i}$	Hash-lists maintained by challenger
$\mathcal{L}_i$	List i maintained by challenger
$\mathcal{C} = \{0,1\}^{l+n}$	Ciphertext space
$Rk_{A \to B}$	Re-encryption key from A to B
m	Plaintext message
c_A	First-level ciphertext
c_B	Second-level/Re-encrypted ciphertext
EHR_i	i^{th} block of EHR
n	Number of blocks
x	Size of each EHR
b_j	Size of j^{th} block of EHR
H_i	Hash value of i^{th} block of EHR
(c_A, c)	Encrypted EHR
(c_B, c)	Re-encrypted EHR
idx	Location index

List of Publications

Referred Journals

1. **Tarunpreet Bhatia** and A.K. Verma, "Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues", **The Journal of Supercomputing**, vol. 73, no. 6, pp. 2558–2631, 2017 **(SCIE indexed, Impact Factor: 1.3).**
2. **Tarunpreet Bhatia** and A.K. Verma, "Cryptanalysis and Improvement of Certificateless Proxy Signcryption Scheme for E-Prescription System in Mobile Cloud Computing", **Annals of Telecommunications,** vol. 72, no. 9, pp. 563-576, 2017 **(SCIE indexed, Impact Factor: 1.41).**
3. **Tarunpreet Bhatia,** A.K. Verma, Gaurav Sharma, "Secure Sharing of Mobile Personal Healthcare Records using Certificateless Proxy Re-Encryption in Cloud", **Transactions on Emerging Telecommunications Technologies**, doi: 10.1002/ett.3309 **(SCIE indexed, Impact Factor: 1.53).**
4. **Tarunpreet Bhatia,** A.K. Verma, "Towards a Secure Incremental Proxy Re-encryption for e-Healthcare Data Sharing in Mobile Cloud Computing", **Concurrency and Computation: Practice and Experience (SCIE indexed, Impact Factor: 1.11).** (Communicated)
5. **Tarunpreet Bhatia**, A.K. Verma, Gaurav Sharma, Suman Bala, "A Novel Defense Scheme against Flooding Attack in Mobile Adhoc Networks", **Recent Patents on Engineering,** vol. 11, 2017, doi: 10.2174/1872212111666170614115605 **(Scopus indexed).**
6. **Tarunpreet Bhatia,** A.K. Verma, "Detection and Isolation of Blackhole and Grayhole Nodes in Mobile Networks", **International Journal of Engineering and Technology**, 2018, **(Accepted, Scopus indexed).**

International/National Conferences

7. **Tarunpreet Bhatia** "Secure sharing of e-health data using certificateless proxy re-encryption" Grace Hopper Celebration of Women in Computing, Bengaluru, India (GHCI), 2017. (Presented)

Book Chapter

8. **Tarunpreet Bhatia** and A.K Verma, "Biometric Authentication for Cloud Computing", Handbook of Research on Securing Cloud-Based Databases with Biometric Applications, IGI Global, pp. 209-235, 2014. **(Scopus indexed)**

Chapter 1
Introduction

Due to novel accomplishments in the field of computing and in-hand technologies, we are heading towards a steady and seamless integration of networking, communication, hardware miniaturization, mobile computing, ubiquitous computing, pervasive computing, wireless sensors networks and a variety of algorithmic advancements. Mobile cloud computing (MCC) integrates mobile computing, cloud computing and wireless networks where data processing and storage occur outside the mobile device Mobile service is among the fastest-spreading technologies in the history. There are several applications such as mobile commerce, mobile learning, mobile health, etc. that are benefitted from it and made a huge impact on the global market. Cloud computing has a potential of providing ubiquitous and on-demand instantaneous access to a common pool of shared resources and services. MCC still in its infancy is different from cloud computing with regard to mobility, bandwidth utilization, fault tolerance, security, etc. The widespread evolution of MCC needs secure, reliable, and non-repudiated identification and authentication mechanisms.

"Security breaches usually entail more recovery efforts than acts of God. Unlike proverbial lightning, breaches of security can be counted on to strike twice unless the route of compromise has been shut off" – FedCIRC.

These days, users have dozens of different username and password pairs to remember so they compromise with safe password practices and put themselves at risk. Organizations and users have become victim of increasing cyber frauds, identity and data thefts which cost global society over USD 388 billion in the last year. Cloud and web services providers

---

Some work/contents of this chapter have been peer reviewed and published in:

- *Tarunpreet Bhatia and A.K. Verma, "Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues", The Journal of Supercomputing, vol. 73, no. 6, pp. 2558–2631, 2017.*

need to give their users a better and secure way of using these web services. The speedy evolution and adoption of cloud computing have inevitably raised concerns enveloping outsourced data in a myriad of safety issues. Protecting user privacy and data secrecy from an adversary is a crucial factor for the proliferation of MCC paradigm. Since mobile devices are resource constrained, providing security against various threats is more challenging than resourceful devices. Designing a secure data access for MCC platform is a huge challenge. There is a need for a secure mobile cloud framework which is capable of providing data security with low processing overhead on mobile nodes. Encryption would be hence an indispensable part of the security solution encompassing confidentiality and integrity.

1.1. Mobile Cloud Computing

Mobile cloud computing (MCC) incorporates mobile computing, cloud computing and wireless networks where data processing and storage occur outside the mobile device. In MCC, the computation functionality and data storage for user applications are moved away from mobile devices to more powerful cloud computing platforms [1]. MCC is one of the promising technologies for changing the world that provides pooled cloud resources toward unrestricted storage, utility and mobility to serve multiple mobile devices through Ethernet or Internet at any time and place in heterogeneous environments [2, 3]. In order to have in-depth understanding of MCC, one should have a clear understanding of cloud computing and its services. Cloud computing has redefined the meaning of computing by replacing a client-server, which is an organizational centric model with more scalable, efficient and flexible data centric model [4]. The cloud renders its services to consumers by providing on-demand access to a shared pool of several computing resources such as server, storage area, applications in pay-as-you-use manner. There is no need to have a powerful mobile device configuration, such as the processor and memory, and all the resource-intensive processing is performed on the cloud. Since mobile devices are resource-constrained in terms of battery life, memory, bandwidth, etc., Cloud Service Providers (CSPs) allow them to use Infrastructure-as-a-Service or IaaS (compute, network and storage), Platform-as-a-Service or PaaS (object storage, identity, queue, etc.) and Software-as-a-Service or SaaS (applications such as monitoring, finance, collaborative, enterprise resource planning) at

low cost and on-demand basis. In SaaS, the consumers can only use service provider's applications that are running on a cloud. They are not responsible for managing underlying cloud infrastructure and computing resources (network, servers, operating system, and memory) except little user-specific configuration settings of the running application, e.g., salesforce.com In PaaS, the consumer can deploy onto the cloud consumer-created applications using tools and languages as provided by the service providers. The consumer is not responsible for managing underlying cloud infrastructure but has some level of control over applications deployed by consumers and their hosting environment configurations, e.g., Google applications. In IaaS, the consumer can deploy and run software (operating system and applications) on the cloud. The consumer is not at all responsible for managing the cloud infrastructure, yet has a full level of control over the operating system, memory storage and extended applications, e.g., Amazon's EC2. Figure 1.1 depicts cloud computing architecture.

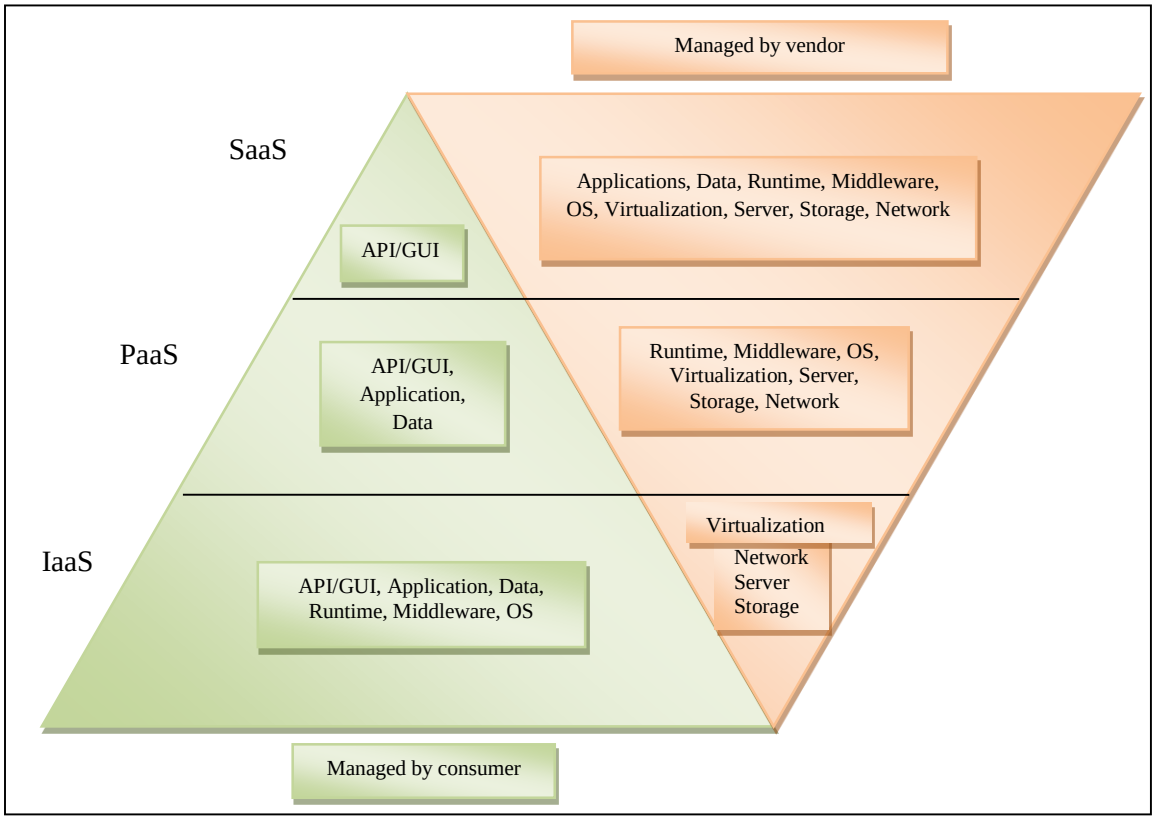

Figure 1.1: Cloud computing architecture

1.1.1 MCC Architecture

The detailed MCC architecture is depicted in Figure 1.2. The mobile devices such as laptops, PDA's, handheld devices can access cloud services either through mobile network or through Wireless Access Points (WAP). Mobile devices connect to the mobile networks via Base Transceiver Stations (BTS) or satellites which are responsible for controlling the connections and functional interfaces between the mobile networks and mobile devices. They transmit the mobile users' requests and data to the Base Station Controllers (BSC) which are further connected to the Mobile Switching Center (MSC) providing a wide range of mobile network services such as AAA (Authentication, Authorization and Accounting) based on the Home Location Register (HLR), Visitor Location Register (VLR), AAA center, Equipment Identity Register (EIR) and subscribers' data stored in databases. The subscribers' requests are then delivered to a cloud through the Internet. In WAP case, the mobile devices connect to the access points through Wi-Fi, which further connects to the Internet Service Providers (ISPs) to provide Internet connectivity. Wi-Fi-based connectivity is more efficient than mobile network GSM, GPRS, 3G, LTE, 4G connections as it provides low latency and consumes less energy. Inside the cloud, cloud controllers connect to data centers and application servers to process the requests and provide mobile users with the corresponding cloud services relying on ubiquitous computing, virtualization and service-oriented architecture.

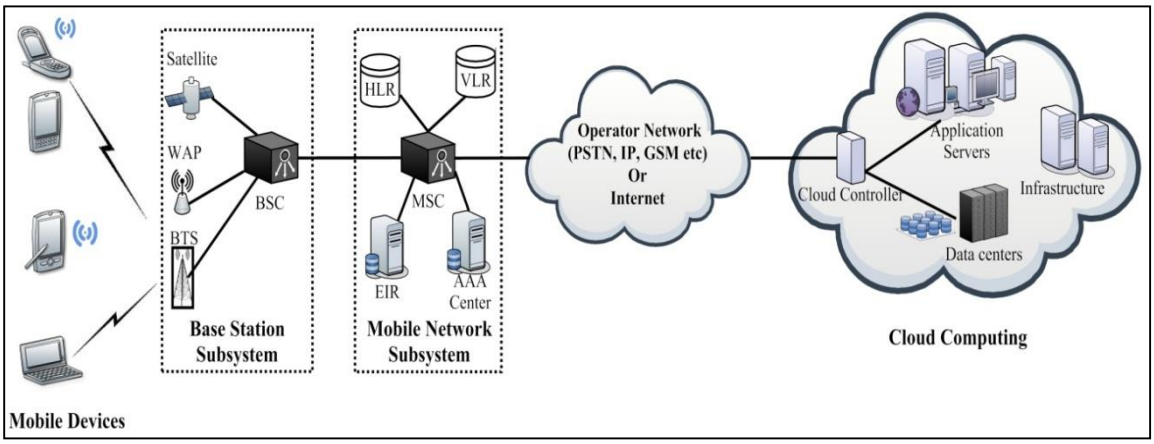

Figure 1.2: MCC architecture

Table 1.1 lists different parameters that differentiate mobile cloud with cloud computing. The thorough knowledge of these parameters helps in dealing with limitations associated with MCC to provide reliable, fault-tolerant and secure mobile app experiences.

Table 1.1: Comparison of cloud computing and MCC paradigm

Parameters	Cloud Computing	Mobile Cloud Computing
Device energy availability	Not required	Limited energy of mobile devices so resource-intensive operations are performed on cloud side instead on the mobile device
Bandwidth utilization	Not required	It should distribute the available limited bandwidth fairly among mobile nodes
Disconnected operation	Not required	Applications served from mobile cloud must support disconnections when mobile users/devices go out of range
Fault tolerance	The cloud needs to be fault-tolerant in events of hardware or software failures	The mobile cloud needs to monitor the connection strength continuously and needs to be fault-tolerant in events of frequent disconnections of mobile device or weaker connection signals
Network latency	Not a major issue	Mobile apps using mobile cloud are more sensitive to high network latencies and disconnected operations than cloud computing
Mobility	Not required	It must support high mobility of devices
Context and location awareness	Not required	Mobile apps must adapt to the frequent changes in the location of devices and collect context information from nearby devices. The applications must monitor the location and context to decide to offload or not to offload on the cloud
Security	It must provide secure cloud services with resourceful devices	It must provide both mobile device security and secure cloud services in the presence of resource-constrained mobile devices and insecure wireless communication medium

1.1.2 MCC Applications

Mobile service is among the fastest-spreading technologies in the history. There are several applications that are benefitted by MCC and made a huge impact on the global market.

- Mobile commerce (m-commerce) has changed the lives of people by providing various applications such as finance, online ticketing, shopping on the mobile devices. These applications were facing challenges such as low bandwidth, low battery power, complex mobile architecture and security risks. MCC integrates m-commerce applications with cloud to overcome above challenges.

- Mobile learning (m-learning) that combines e-learning with mobility also faces several challenges such as low transmission rate and high cost of mobile devices. These can be overcome by utilizing cloud for large storage and high processing capabilities.

- Mobile healthcare (m-healthcare) allows mobile users to access medical resources in an efficient way as opposed to traditional medical applications. The availability of on-demand services on the cloud overcomes mobile medical application's limitations such as low storage capacity, security and reliability of data.

- Mobile gaming (m-gaming) offloads game engine module to the resourceful cloud servers as it requires high computation and graphic rendering.

- Mobile apps can use the cloud for app development and app hosting. Mobile apps running on multiple mobile platforms such as Apple iOS, Android, Windows at the same time may rely on the cloud for storage, performing heavy computations, and fault tolerance to evade the repeated development and maintenance efforts.

Out of the above, the major focus of this work is on electronic health (e-health) and mobile health (m-health) as discussed in Section 1.3.4.

1.1.3 MCC Challenges

Even though migrating to the mobile cloud is a tempting trend, there are several other factors that must be taken into account which are as follows:

- **Heterogeneous environment:** MCC is deployed in heterogeneous networks where mobile nodes access the cloud through various network interfaces and radio technologies like GPRS, CDMA, WCDMA, WLAN, WiMAX. Due to heterogeneity, there arises an issue of handling wireless connectivity which should satisfy certain MCC requirements such as always-on connectivity, on-demand scalability and energy efficiency of mobile nodes.

- **Low bandwidth:** Bandwidth is a major hurdle with the increase in the number of cloud users at an alarming rate so there is a need to share and distribute the available limited bandwidth fairly among mobile nodes located in close proximity, within an organization, workplace, etc. or involved in same content. 4G and Femtocell are the emerging technologies which can overcome this bandwidth limitation.

- **Security and privacy issues:** Protecting user privacy and data secrecy from an adversary is an important factor for the prosperity of MCC paradigm. Since mobile devices are resource-constrained, protecting them from numerous security threats is more challenging than securing resourceful devices.

- **Identity management and access control:** The cloud servers being highly virtualized and federated are not visualized as trustworthy, so they need identity management and fine grained access control policies allowing encrypted data to be disclosed to authorized users only.

- **Key management:** Before outsourcing data in the cloud, data encryption using cryptography can prove to be an efficient way for ensuring confidentiality and protecting data. Key management (key generation, distribution and monitoring) is a critical element for data encryption in MCC environment.

- **Searchable encryption techniques:** Since data is encrypted before outsourcing to cloud servers, searching encrypted data in the cloud has gained researchers' attention these days [5].

- **Remote integrity check:** Designing efficient remote integrity check protocols for verifying integrity of data stored on the cloud without downloading the copy of data on the mobile device is an important security issue gaining attention of researchers' community.

- **Context-aware applications:** Designing resource-efficient and context-aware applications to improve the QoS is a cardinal requirement for unleashing the power of MCC in the heterogeneous environment towards unrestricted ubiquitous computing. In order to optimize mobile access and to provide context-aware mobile services to different users, it is important to monitor the relevant information about the device such as location, capabilities, environment, user profiles and preferences.

- **Energy management:** The major issue in mobile cloud environment is to lessen the energy utilization and make the cloud and mobile devices more energy-efficient. To make this happen, CSPs must use energy efficient scheduling, carefully designed transmission protocol and server consolidation so that resource wastage is minimized.

- **Live virtual machine migration:** It enables the highly robust and responsive provisioning in the data centers. It is done using virtualization techniques by balancing the load across the datacenters. The main issue is how to migrate the whole virtual machine (VM) in the current state from one server to another within a few milliseconds securely and without any data loss.

- **Availability:** Service availability is one of the major issues in MCC than normal CC paradigm. There are several disconnections due to traffic congestion, network or link failures, node mobility, and signal attenuation in MCC paradigm. These disconnections make mobile users unable to connect to cloud for a service. There must be some mechanism for allowing mobile user to connect to cloud through neighboring nodes in the events of disconnections.

- **Standard interface:** Customers are reluctant in moving their data centers to the cloud due to the absence of standards. Absence of portability and interoperability also makes it difficult for an application or data to be transferred or scaled over multiple CSPs. The current web interface for communication between mobile users and cloud incurs more overhead so a standard interface is required to ensure seamless connection between mobile users and CSPs. HTML5 Web Sockets can provide an efficient interface in the near future**.**

1.2 Security issues in MCC

The widespread evolution of MCC needs secure, reliable and non-repudiated identification and authentication mechanisms. Security has emerged as an obstacle in the adoption of cloud and mobile cloud computing [6] even though it delivers a wide range of resources. In this section, several security issues, threats and possible attacks have been addressed. These issues have been classified in Figure 1.3.

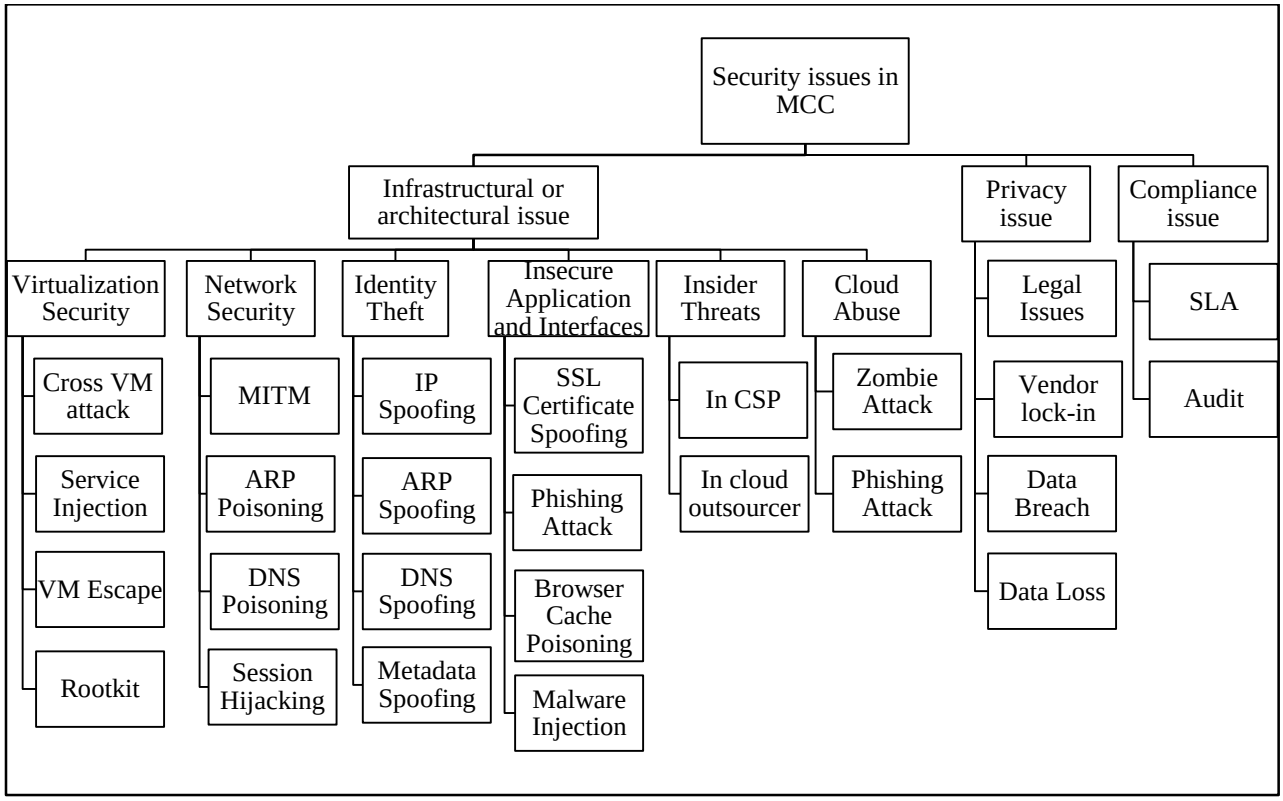

Figure 1.3: Security issues in MCC

1.2.1 Infrastructural issues

This dimension includes virtualization security, network security, data segregation issues, insider attacks and administrative interface issues.

- **Virtualization security:** Virtualization, one of the major components of cloud computing, allows multiple users to store data using applications provided by SaaS providers and share on-demand services. This poses high potential of intrusion into tenants' data if their data are not segregated properly at physical as well as application level. It leads to many risks such as isolation between multiple VMs running on the same physical machine, Cross-VM attacks, injecting malicious code into the application, virtualization software known as hypervisor vulnerabilities that can be exploited to bypass authentication and escalate privileges [7]. For example, vulnerable software Microsoft Virtual PC and Microsoft Virtual Server allow guest machines to run malicious code on the host machine or other guest machines. VM Escape attack allows an attacker to break an isolation layer to run an application with the hypervisor's root privileges and get access to the host operating system and other virtual machines running on the host machine. Rootkit is a set of programs or tools used by an attacker to gain administrator-level privileges on a machine either by cracking passwords or by exploiting hypervisor's vulnerabilities. VM-based rootkits allow an attacker to execute malicious code to destroy anti-malware programs to escape detection and control over any VM running on the physical compromised machine. To defend against such attacks, comprehensive monitoring of hypervisors and strong isolation between VMs are required, which curbs an attacker to inject malicious code into the neighbor's VM.

- **Network security**: It deals with network communications and configurations. The inherent vulnerabilities of Internet protocols such as ARP, HTTP, TCP allow an attacker to exploit the cloud system and its resources through Man-In-The-Middle (MITM) attack, ARP poisoning, DNS poisoning, session hijacking, etc. Data is obtained from an enterprise and stored on cloud so the strong network encryption techniques such as SSL and TLS are required to protect against such attacks.

Improper configuration of SSL leads to a MITM attack in which an attacker can access the data communication between two parties such as data centers, VMs. The malicious attacker can redirect all inbound and outbound traffic from other VMs to its own machine through ARP poisoning attack since ARP does not require proof-of-origin. DNS poisoning attack is tricking the domain name server (DNS) to send traffic in the wrong direction by modifying DNS cache content maliciously. The cloud customers must ensure that CSPs are taking proper steps to secure their DNS infrastructure. In session-hijacking attack, an attacker exploits improper implementation of session ID's in HTTP and assumes the user identity for further communication. Anonymous authentication and encrypting traffic can thwart such attacks. Firewalls and protocols must be configured to provide the required level of security for cloud environment.

- **Identity thefts:** It is a form of fraudulent act in which an entity pretends to be someone else to access resources or obtain banking and other critical information. DNS spoofing, IP spoofing, ARP spoofing, metadata spoofing, phishing attacks are all forms of identity thefts. An intruder obtains the IP address of a legitimate user and alters TCP/IP packet headers to masquerade as a trusted host and hides its identity in order to launch IP spoofing attack. This attack can be used to hijack browsers, overload targets with traffic and steal information. In an ARP spoofing attack, an intruder binds its MAC address to an IP address of a legitimate VM in a network and sends spoofed ARP messages that resulted in data intended for the legitimate host's IP address sent to the intruder VM attached to same virtual switch instead. Malicious attackers use ARP spoofing to gain access to sensitive information or network resources, modify data-in-transit or block traffic on a LAN. In DNS spoofing attack, an attacker machine supplies false DNS information to a VM so that when it browses a particular site, browse request is redirected to a fake IP address created by an attacker to steal banking credentials. Web services description language (WSDL) file contains metadata or descriptions about services offered so that an individual can access those services electronically. In metadata spoofing attack, intruders can steal information or inject malicious code by modifying a service's WSDL file at a service delivery time. In a phishing attack,

an adversary sets up a fake URL identical to the real Web application fooling the users to enter a valid credentials and certificates. According to Financial Fraud Action UK, financial fraud largely driven by identity thefts corresponds to £755m in 2015, 26% rise from 2014. Recently in 2016, Collins of Lancaster, Pennsylvania, has been accused of gaining access to more than 100 Apple iCloud and Gmail accounts illegally by sending phishing mails to clients. After this incident, Apple recommended to enable two-factor authentication as passwords can be easily compromised by hackers. Strong remote authentication and authorization mechanisms should be enforced for accessing sensitive information stored on the cloud.

- **Insecure application and interfaces:** CSPs rely on user interfaces for exploring resources and tools available by CSPs; administrative interfaces for VM management, deployment, coding, testing, monitoring, user access control and configuration and programming interfaces for accessing virtualized resources and service provisioning. Weak user interfaces and API can expose an organization to several security risks. Somorovsky et al. [8] had provided a security analysis of control interfaces of Amazon and Eucalyptus that they can be easily compromised through signature wrapping and cross-site scripting (XSS) techniques. Defects in the design and architecture of applications lead to malware injection attacks such as SQL injection, OS injection, XSS injection and LDAP (lightweight directory access protocol [9]) injection attacks. An adversary compromises cloud system by injecting malicious code in a service or malicious VM instance, which may modify or block service functionalities. The legitimate requests are thus redirected to malicious services. For example, in the Amazon EC2 API, an adversary can introduce malicious command instead of security group name [10]. SSL protocol ensures secure communication between a user's browser and server through the use of SSL certificates which can be verified by user's browser. If the server's SSL certificate is vouched by any of the trusted CA's pre-listed in the browser, then secure communication is established otherwise warning to the user is issued. SSL secure connection can be broken by impersonating legitimate server through SSL certificate poisoning attack to intercept sensitive information. In browser cache

poisoning attack, an attacker performs MITM attack on HTTPS session of a user and replaces the cached content with malicious data. A reliable end-to-end encryption, data validation checks, secure user interfaces, strong authentication and access control policies can avoid such attacks.

- **Insider attack:** Mostly, organizations and CSPs focus more on protection against external attackers and less on insider intruders. An insider attacker can easily harm the network or system as he has an authorized access to system and much familiar with the network architecture and system security procedures. The insider employee can be either a malicious employee working for the cloud provider or an employee of an organization which is utilizing cloud services. Coherent demarcation of duties and transparent employee management policies can reduce the extent of damage caused by insider attacks.

- **Cloud abuse**: It means using cloud services for malicious intent, such as for breaking encryption keys, sharing pirated software or propagating malware launching DDoS attacks and phishing attacks as it was difficult to perform this using standard computer. Proper validation/verification checks during initial registration phase and constant monitoring of network traffic can prevent such attacks.

1.2.2 Privacy issues

This dimension includes data security, data loss and legal issues. The data of customers and services by providers are hosted at different geographic locations so it is affected directly or indirectly by multiple jurisdictions or subpoena law enforcement measures. Governance issues such as vendor/data lock-in, data and security control arise due to dissimilarities in underlying cloud architectures and loss of administrative controls of a customer in a cloud environment. The availability of large variety of cloud servers and mobile devices on the one hand and lack of well-established uniform standards and APIs on the other hand aggravate vendor lock-in issue in MCC. Law enforcement also results in disclosure of hardware device as demanded by a court of law for a particular customer, but this will in turn affect all the customers whose data were stored on that particular device

[11]. Breaching into the cloud platform that stores data of various users and organizations can attack the data of all users compromising confidentiality, integrity and authenticity (CIA). The major data security challenges for achieving CIA are key management, identity management, access control and remote integrity check. Data confidentiality ensures that information is available only to authorized users. Data integrity means that data should not be tampered or altered accidently or intentionally. Data authenticity ensures that both parties involved in the communication are who they claim to be. Data loss means access to data in the events of CSP's carelessness, malicious attack, power outages, natural disasters such as earthquake, fires or floods. It can turn into a problematic situation for CSPs when data storage complies with certain laws such as HIPAA (Health Insurance, Portability and Accountability Act). The efforts put to mitigate data loss can exacerbate data breach and vice versa. We can encrypt data to reduce the impact of data breach, but if encryption keys are lost, data are gone. On the other hand, if the offline backup is kept for valuable data to reduce data loss, there is a high potential of exposure to data breaches. Solutions include strong encryption policies, data integrity checks, secure storage of encryption keys and efficient data backup strategies.

1.2.3 Compliance Issues

This dimension deals with service availability and audit capabilities. Service level agreements (SLAs) ensure required service availability and procedures to be adopted to guarantee a certain level of security [12]. Auditing is done by customers, service providers and third parties to continuously assess security and availability services. SaaS providers should consider certain regulatory and legislative frameworks while storing privacy-related information in the cloud such as HIPAA, Gramm-Leach-Bliley (GLBA), the Fair and Accurate Credit Transaction Act (FACTA), Privacy Act of 1974, Computer Security Act of 1987.

Table 1.2 summarizes above-mentioned potential security issues or threats to MCC, their effects on various cloud services and relevant mitigation techniques.

Table 1.2: Summary of threats, associated attacks and mitigation techniques

Threats	Effects	Cloud service affected	Possible Attacks	Attack Surface	Mitigation Techniques
Virtualization Security	Allow an attacker to gain access over another user's VM, provide malicious service to user	PaaS, IaaS	Cross VM, VM Escape, Rootkit, Service Injection	Compromising hypervisor and service identification files	Securing hypervisor, comprehensive monitoring at hypervisor level, VM isolation, IDS/IPS (Intrusion Detection/Prevention System), Data validation checks
Network Security	Affects the data security and privacy	SaaS, PaaS, IaaS	MITM, ARP Poisoning, DNS Poisoning, Session Hijacking	Accessing data communication between two parties	Anonymous authentication, Encryption, SSL, TLS
Identity Theft	Allow an attacker to be someone else or access their resources	SaaS, PaaS, IaaS	IP Spoofing, DNS Spoofing, ARP Spoofing, Metadata Spoofing	Weak password recovery methods, key loggers	Strong authentication mechanisms
Insecure Applications and Interfaces	Clear text authentication and improper authorization	SaaS, PaaS, IaaS	SSL Certificate Spoofing, Phising, Attack on browser cache, Malware Injection	Unauthorized access to services and management interface	Strong authentication and access control policies, Hash based service integrity check, Secure browsers and API
Insider Threats	Affect confidentiality, integrity and availability of resources	SaaS, PaaS, IaaS	In CSP or cloud outsourcer	Gaining access over confidential data and cloud services	Transparent employee management process including compliance reporting and breach notification, segregation of duties
Cloud Abuse	Affect service availability and privacy of user's	SaaS, PaaS, IaaS	Zombie, Phishing	Allow intruder to compromise valid user's VM, allow user to access fake	Strong authentication and authorization, constant monitoring

	sensitive information.			link, lack of validation	of network traffic against intruders
Data breach and loss	Unauthorized disclosure of private data behind applications	SaaS, PaaS, IaaS	MITM	Weak encryption algorithms, lack of authentication, authorization, weak keys and their storage, unreliable data centre	Strong encryption policies, secure storage of keys and API's, efficient data backup and retention policies

1.3 Data Security in MCC

The security architecture in cloud environment has been divided into three layers as illustrated in Figure 1.4. The inner layer deals with different deployment models discussed above. The layer above it constitutes different delivery models: SaaS, PaaS and IaaS is forming the core architecture of cloud and exhibiting characteristics like scalability, on-demand self-service, location dependency, multi-tenancy etc. as shown in outer layer. Cloud security depends on the deployment and delivery model used, and the characteristics exhibited by it [13]. The fundamental security challenges involved in adopting virtualization and cloud computing encompasses data storage security, hardware security, software security, access control, data transmission security, application security and security related to third-party sources.

The main questions that come up in the mind while migrating to the cloud are:

- How secure our data is on the cloud?
- Where will our data be stored on the cloud?
- Who all have access to our stored data in cloud?
- How much a client can trust the company or third party?
- Are the cloud providers keeping data of different clients inaccessible from other clients?

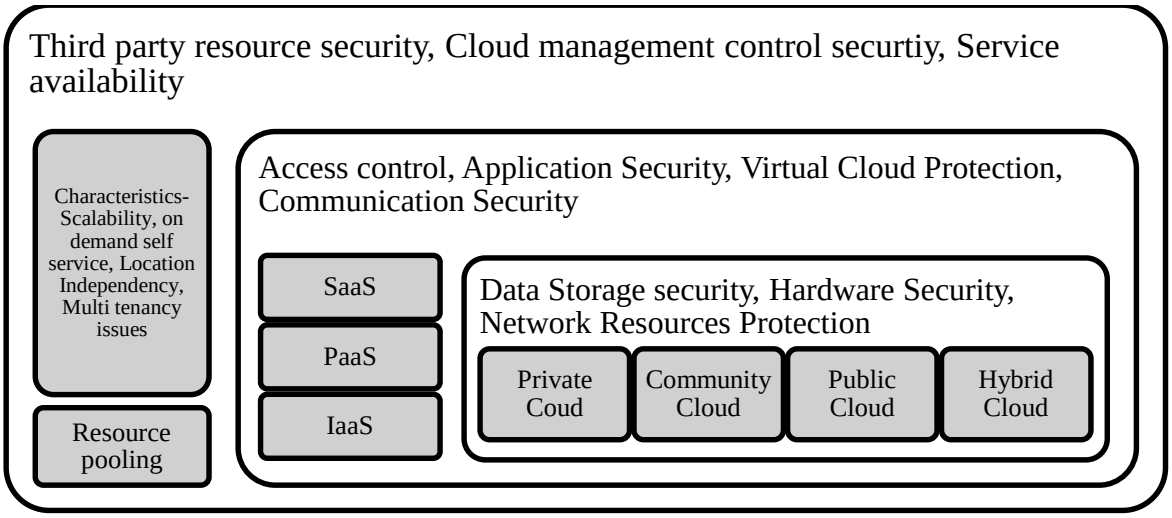

Figure 1.4: Security architecture in cloud

One of the major issues, still unresolved, for CSPs and their clients, is the difficulty in figuring out who is responsible for what security measures and controls. The service providers are responsible for creating services and features compliant with data protection and privacy standards on one side, and a customer can configure and use those services in a way compliant with its industry and location on the other side. The service providers can create operational controls to protect customer's data on a cloud, and customers have to use those controls to prevent unintended data sharing. The service providers are responsible for obtaining certifications and signing SLAs, whereas customers are responsible for verifying service providers' audit reports and certificates according to their organizational data privacy requirements.

The boundary between these responsibilities is not clear and thus depends on the agreement signed between the customer and service providers and on the cloud service and deployment model used. For example, PaaS and IaaS providers share the security responsibility equally for everything above the virtual machine layer with the clients. But with SaaS and cloud apps, the customer becomes more responsible for things such as access control and monitoring. In private cloud, customer is responsible for security at all the levels, whereas in public cloud, security responsibilities are shared between CSPs and customers. Customer is responsible for securing applications and data deployed on cloud platform and cloud service.

1.3.1 States of data

While securing data in the cloud, we need to figure out possible states in which the data may occur and available controls for that state. With the proliferation of Internet and cloud computing in recent years, protecting data-at-rest is considered as important as protecting data-in-transit [14]. Encryption protects the data in the cloud from security breaches, third-party disclosures and compliance violations [15]. Encryption is considered important for all the three possible states of data: in-transit, at-rest and in-use as shown in Figure 1.5.

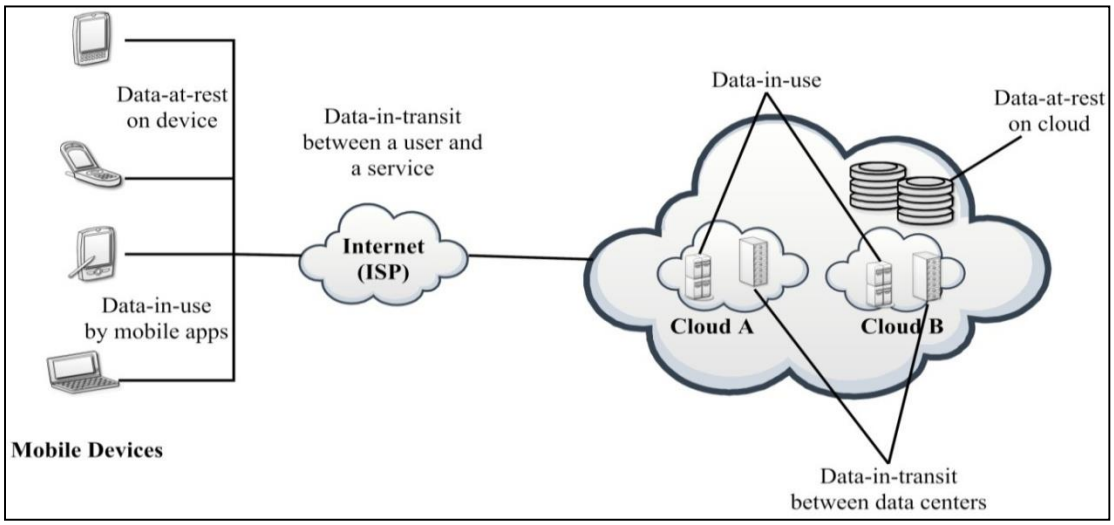

Figure 1.5: Possible states of data

The choice of strong encryption algorithms and key management policy is critical for success of any encryption policy. For better security, customers can adopt their own key management infrastructure instead of adopting CSP's default key management infrastructure. Skyhigh Networks Inc., recently analyzed encryption controls of 12000 cloud providers and made a statement that 81.8% of CSP encrypt data-in-transit, protecting data from MITM attacks as it moves through the Internet, but only 9.4% of CSP encrypt data-at-rest, making it vulnerable to unauthorized access, data breaches and blind government subpoenas. Among them, only 1.1% of CSP use customer-managed encryption keys. The most popular apps such as Facebook, Twitter, PayPal, Gmail, Linkedin, eBay, etc. store data (user credentials, payment card numbers, bank account

numbers) in an unencrypted form. Ebay suffered biggest data breach in 2014 when 145 million account credentials were stolen. The possible states of data are as follows:

- **Data-in-transit:** Data including voice, video, text, metadata is thought to be in motion once it leaves an enterprise control and moves across the network or to cloud and vice-versa, and thus, its encryption is vital. It involves not only communication with a component outside the cloud service, but also communication between virtual networks. It has to be protected against eavesdropping attack through cryptographic protocols such as SSL or TLS by establishing an encrypted and authenticated channel.
- **Data-at-rest:** It refers to inactive data which is stored physically on NAS, SAN, file servers in the form of databases, data warehouses, off-site backups, etc. In addition to encryption, strong access control policies and data federation should be used to thwart attacks.
- **Data-in-use:** It refers to the dynamic data which is stored in non-persistent state, e.g., data or encryption keys in cache, main memory, transactions in a message queue, data currently processed by an application. These data are generally in clear text form for performing value-added functions such as searching, retrieval of the data, but cloud security alliance now recommends encryption of data-in-use for better security. Fully homomorphic encryption allows computations on ciphertext and the results thus obtained when decrypted match the computations performed on plaintext. Enclaves are used to secure data-in-use in which data is in encrypted form in RAM, but available as clear text inside CPU.

1.3.2 Security principles

The basic security principles that any data security framework must comply are confidentiality, integrity, authentication and non-repudiation. Apart from these, there are two more principles, availability and access control which are linked to the entire system as a whole and not to particular data or message.

- **Confidentiality:** This principle ensures that only sender and recipient(s) should access the message. It prevents unauthorized access to data, and loss of confidentiality leads to interception.

- **Integrity:** It ensures correct delivery of message to the intended recipient(s) without any alteration. Loss of integrity leads to modification attack.

- **Authentication:** It helps in establishing proof of identities that the communicating node is what it claims to be. Absence of authentication measures leads to fabrication attack.

- **Non-repudiation:** This principle makes sure that the sender cannot deny later on for not sending the message.

- **Access control:** It ensures the use of network's resources and services by only the authorized users. It acts like a bridge between confidentiality, integrity and authenticity. It begins with authentication and then identifies who can "access" what, where access comprises reading of data (confidentiality) and writing (integrity).

- **Availability:** It ensures that authorized parties are able to access the information when desired. Denying access to information leads to DoS attack in which legitimate users are denied access to resources.

1.3.3 Classification of data security schemes

There is a need of securing data of an organization or enterprise on cloud to ensure confidentiality and privacy that facilitates the need of efficient key management schemes while using cloud SaaS. With the widespread use of sophisticated hacking tools, the data stored in the cloud is at increasing risk of malicious attack [16]. So, there should also be robust authentication schemes for cloud computing to ensure access to data by legitimate and authorized users only. SaaS providers must verify the identity of every single user attempting to access the cloud system. Data security schemes in the MCC are classified as:

- **Password based schemes**: Most of the current systems are based on password-based authentication in which server maintains a database for passwords or their hash values after salting. Lamport [17] proposed first one-time password scheme in 1981 for remote authentication. They are easily deployable, but an attacker can easily steal, guess or modify the passwords stored in the server. It also suffers from scalability issue. Recently, Ren et al. [18] proposed a lightweight, fine-grained and flexible scheme for access control (F2AC) in multiparty file sharing in mobile cloud computing environment. It supported dynamic operations such as adding or deleting users within an ad hoc group, authorizing or revoking privileges for members transitively and segregating access authentication from system authentication. It consisted of two lists: access control and user authentication list, both maintained at cloud. The Access Control List (ACL) is comprised of filename, users, privileges (read, update, modify, create) and conditions (location, MAC addresses, etc.), whereas user authentication list contained the username and corresponding token or password. The creator of the file assigned and sent token to other users with whom it wants to share files. It is vulnerable to impersonation, password guessing and replay attack. Problems with passwords highlight the need for another system of user identification [19].

- **Smart cards based:** Smart cards possess an individual's identity-related information in encrypted form on an embedded chip. On-chip defense measures minimize identity thefts. Individual smart cards can be programmed for multiple uses, e.g., banking transactions, medical entitlement. But the information stored on smart cards can be revealed [20]. If a smart card is compromised by an adversary, then the user is also compromised. Jangirala et al. [21] proposed an efficient remote user authentication scheme which is based on dynamic identity using smart cards and is applicable for multi server environment. Halevi and Krawezyk's protocol [22] is used for password-based one-way authentication protocol. Each client shared symmetric key and password with server, which is saved by the server in two different tables and smart card is used by the client to store symmetric key. The password-based one-way authentication protocol is transformed to password-based

mutual authentication and key exchange protocol (PWAKE) which is further upgraded to two-factor smart card-based, password mutual authentication and key exchange protocol by using pseudorandom functions and collision-resistant hash functions. Hwang and Li [23] proposed smart card-based authentication scheme relying on an ElGamal public cryptosystem. But it provides one-way authentication.

- **Cryptography based**: It is a conventional way of maintaining the privacy of data and authenticating users over an insecure network. The individual or group of users possessing correct cryptographic key have access to encrypted data. It not only protects data from theft or alteration, but can also be used for user authentication. A brief classification of various cryptography-based data security schemes in MCC is given in Chapter 2.

- **Biometrics based:** In cryptography-based schemes, an attacker can obtain keys in illegal ways and pretend to be a genuine user. A very reliable and natural solution to prevent such attacks is to use biometric traits, which recognize users through physiological and behavioral characteristic possessed by them. Your fingers, your eyes and voice are always with you, and these cannot be imitated or possessed by others. But it might be possible that biometric templates such as fingerprint are acquired stealthily and used by a malicious user. There are several cloud-based biometric recognition solutions available in the market such as Eyeprint ID [24], BioID [25], Face.com [26], Animetrics FaceR [27], Calsoft Labs [28]. According to recent research by New York-based firm, 240 million users or businesses will continue to use cloud services through mobile devices by 2015, increasing the revenue of mobile cloud computing to $5.2 billion.

- **Muti-factor authentication schemes**: They are a combination of two or more schemes in an effort to provide more efficient data security and authentication schemes. There is always a threat of password being cracked even if biometric information is protected using password in some multifactor authentication schemes. It is feasible to integrate biometrics to cryptographic infrastructure.

Biometric authentication appearing to be a more reliable solution than other traditional authentication measures is also vulnerable to attacks when it comes to remote authentication over open networks.

- **Intrusion detection based:** Cryptographic solutions are proactive and computation-intensive. They perform well for anticipated attacks, but may collapse under unknown and unanticipated attacks. The increasing popularity of smartphones by offering advanced computing has given birth to both generic malware (viruses, Trojans, worms) and smartphone-specific malware that may exploit Bluetooth interface or voice-recognition algorithms. So, we need a second wall of defense, i.e., an intrusion detection mechanism for detecting and reporting unusual behavior. An effective intrusion detection mechanism should save all malware signatures in phone, but it will consume large computation and memory resources of mobile devices. Zonouz et al. [29] developed Secloud, a cloud based lightweight security solution for smartphones. It emulates a smartphone in a cloud and keeps it synchronized by sending the device inputs and network connections continuously to the cloud. It consists of 3 main entities: client agent which is a lightweight software running on smartphones for collecting user and sensor inputs from the device and passing to emulated replica on a cloud; proxy server for mirroring network traffic between smartphones and Secloud's replica; and emulator in the cloud running various host-based and network-based security solutions such as virus scanners, a file integrity checker, network-based IDS, snort. It performs security analysis using intrusion detection techniques on emulated replica instead of the device itself, thereby reducing the energy and processing power requirement of mobile devices.

1.3.4 e-health and m-health data

Due to novel accomplishments in the field of computing and in-hand technologies, we are heading towards a steady and seamless integration of networking, communication, hardware miniaturization, mobile computing, ubiquitous computing, pervasive computing, wireless sensor networks, cryptography, and a variety of algorithmic advancements for

smarter healthcare. The aforementioned technologies and healthcare informatics have led to the foundation of e-health and m-health, leading to advancements in on-the-fly communication, connecting previously remote locations along with an improvement in lifestyles and enhancement in decision making in the crucial sectors of healthcare.

Conventionally, a patient's health information remains fragmented at all those health centers where a patient visits for probable treatment. These fragments of data, when integrated together, holds a bigger picture of a patient's health status. This is an imperative key and could significantly assist physicians in prescribing treatments at times of regular checkups, consultations or even during emergencies. An Electronic Health Record (EHR) is a centralized repository storing all the fragments of a patient's health information from all physicians and clinicians ever consulted and this valuable record can be easily accessed by all authorized healthcare providers (laboratories and specialists) at all times.

Personal Health Records (PHRs) resemble the EHRs on the informational level. However, the user's role in accessing, updating and managing the PHRs with their health information distinguishes the PHR from the EHR. Clinicians, home monitoring devices and patient's self-judgment constitutes PHR. On comparing, the PHRs fare better than the EHRs in their assistance for monitoring a patient's health information and in the effective communication of information. Nowadays, the healthcare delivery paradigm is gradually evolving from organization-centered towards patient-centered approach. A demand in the upkeep of EHRs and PHRs along with easily accessible information is driven by the need for a patient centered approach in healthcare industry. A ubiquitous and well integrated PHR allows the storage and time-bound access to a patient's health information that has brought the patient at the core and hence pushed the fragmented patient health information out of the foray.

With the coming of age, iPads have become so prevalent that they have very smoothly taken over as ubiquitous roles as the stethoscopes coming in handy for healthcare providers in hospitals and medical offices. The same level of prevalence and visibility holds true for mobile devices with the coming of patient portals enabling the patients to schedule appointments, view reports and instant follow-ups by asking questions missed during regular checkups. Technology aids the patients especially those suffering from chronic

diseases in monitoring and managing their health, leading to improved results with lower expenses. Pocket friendly portable technologies of mobile communication devices (MCDs) such as cell phones, tablets collect environmental data, and the information fed in by the patient and transfer it to the PHR using the Internet. Coupled with an active decision support system, MCD-PHR pair can examine the data and send notifications on the patient's MCD as timely medication reminders, bill memos and health warnings.

PHRs can be made accessible to healthcare consumer by healthcare providers, commercial vendors or health record banking associations. A major objection discouraging the users from adapting to PHRs was the fact that they dreaded their medical data being used without their consent. The PHRs in offering from healthcare providers for users went past this obstacle. This proved that majority of the patients were comfortable in trusting doctors with their personal information and gladly accepted its accessibility online. To this, the major EHR vendors' response was to make the patient accessible PHR options available along with the traditional EHR software meant for physicians and hospitals for their conventional roles in medical information management.

EHRs or PHRs are distinguished from generic cloud data via their characteristic set of security, regulatory, and operational constraints.

- EHRs or PHRs must adhere to HIPAA, GDPR (General Data Protection Regulation) etc security guidelines following its strict regulatory policies (administrative, technical and physical) for it to be law compliant. HIPAA mandates the protection and confidential handling of medical documents and personal health information.

- Individual EHRs or PHRs consist of massive data in the form of X-Rays, MRIs, CT scans, pathology lab test reports, physician diagnosis and medical transcripts, etc. which keep on increasing over the life span of the respective patient. The security solution for e-healthcare data storage in the cloud should work efficiently on large data sets.

- Individual EHRs or PHRs needs to be modified frequently, depending on the respective patient's medical history. The security solution for e-healthcare data storage in cloud should be able to securely support dynamic insert, delete, update operations on cloud data.

- Presently, a major part of EHRs or PHRs is examined and updated using low battery-powered mobile devices. It is imperative to design an energy-efficient security solution for e-healthcare data storage in the cloud for a judicious use of the battery resources of energy-starved mobile devices operated by patients, physicians and medical personnel.

- EHRs or PHRs should remain accessible for long periods of time spanning for about an entire lifetime of the patient all the while maintaining its confidentiality and integrity of ever-increasing EHRs or PHRs during entire detainment period.

E-health applications such as personal health monitoring generate an enormous amount of data for a single patient even, and it becomes cumbersome for healthcare units to store such data of all the patients on a single computer because of catastrophic failures. A growth in the amount of information leads to the requirement of efficient storage and management. Many hospitals and healthcare units are making use of on-cloud storage services for patient records that would give them an upper hand in terms of saved expenses and faster access. Cloud computing enriches and facilitates healthcare by providing an easy to use web-based or mobile-based framework for managing and sharing personal health data. The clients owing resource-constrained mobile devices can outsource the heavy computations and massive data to the untrusted cloud servers and take benefit from the unlimited computing and storage services in a pay-per-use manner. The cloud computing paradigm comes tailor-made with advantages of storage, speed of computation along with many other pros.

Inspite of the inherent advantages of cloud computing in e-healthcare industry, the herculean task of migrating e-heath data on the public cloud impedes its widespread adoption. A crucial aspect to be addressed concerning the storage and processing of data on the cloud is to safeguard the confidentiality and integrity of the patient's medical information. The concerns regarding the confidentiality and integrity of patients' data is of

utmost priority since any malicious modification on EHRs could result in erroneous medical decisions. Even though role played by cloud computing is intermediary yet the health care institutes employing such services are vulnerable as they lose the ownership, control or management of the EHR data which is indirectly stored and processed on untrusted servers. Any malicious intentions on part of the CSP would irrevocably jeopardize the integrity of the patients' medical records.

1.4 Thesis Contribution

MCC will be a main information infrastructure in the near future consisting of various large, geographically distributed and heterogeneous data centers. Designing a secure data access for mobile cloud computing platform is a huge challenge. MCC inherits all the security issues that exist in cloud computing with an additional limitation of mobile devices' resources like bandwidth, processing power, memory, etc. so security algorithms for cloud computing cannot be used directly on mobile cloud platform. There is a need for a secure mobile cloud framework which is capable of providing data security with low processing overhead on mobile nodes. Security mechanisms should satisfy the constraints of real-time response under frequent disconnections and low communication and computation overhead.

1.4.1 Motivation

Even though migrating to the mobile cloud is a tempting trend from a financial perspective, there are several other aspects that must be taken into account by CSPs for providing cloud services and organizations before outsourcing their sensitive data to CSPs. According to Cisco VNI Mobile Data Traffic Forecast [30], "Global mobile data traffic will grow 8-fold nearly from 2015 to 2020 with a compound annual growth rate (CAGR) of 53%, reaching 30.6 exabytes per month by 2020. 67% of mobile devices will be smart devices by 2020 as compared to 36% in 2015. People all over the world are consuming more and more wireless bandwidth to manage various tasks. The vast majority of mobile data traffic (98%) will originate from these smart devices by 2020, up from 89% in 2015". According to a report from Markets and Markets [31], "the mobile market is expected to grow from USD 1.18 billion in 2015 to USD 3.26 billion by 2020, at a CAGR of 22.5% from 2015 to 2020. The

cloud security market is expected to be worth $8.7 billion by 2019". The huge increase in demand for smartphones and tablets has simultaneously impacted, IT solutions to scale up application development for mobile computing, also ensuring the security is in place.

The researchers have proposed several data security schemes, but still the field is open to find a lightweight approach suitable for resource-constrained mobile devices without compromising security. Symmetric cryptography is simple and efficient in resource utilization. The main benefit to use symmetric key system is low computing cost, but it needs a prior key distribution. This approach is also impractical for cloud computing, as it fails to provide scalability. This cryptography fails to provide non-repudiation and authentication mechanisms.

Traditional public key (asymmetric) cryptography is more expensive than symmetric cryptography as it needs certificates for authentication, which requires high computations. There is a scope of identity-based key agreement, which can authenticate the sender and, the adversary cannot able to take the advantage by impersonating as a user. But it suffers from the key escrow problem. The research investigation can also be focused on designing a certificateless scheme, which can provide the security against key escrow problem and solves certificate management problem in traditional Public Key Infrastructure (PKI). The research can be directed to find the vulnerabilities in the existing schemes and countermeasures can be provided, if any.

In terms of computation overhead, pairing operation is the most expensive operation among other public key operations, such as elliptic curve point multiplication, elliptic curve point addition, etc. The major intention is to reduce the number of computational operations so that the approach can be applied to MCC.

Proxy signature, a variant of the digital signature, allows an original signer to entrust his signing authority to a third party known as a proxy signer in the events of his absence for signing messages on his behalf. A large number of certificateless proxy signature schemes have been quoted in the literature, however, design of certificateless proxy signcryption schemes stands neglected. Such schemes have a dual benefit of saving storage space and communication bandwidth as a message needs not to be sent separately with its signature.

A conventional way of incorporating data confidentiality involves a significant effort by the data owner in encrypting the data with key (symmetric or public key) before outsourcing to the cloud. The traditional asymmetric cryptography is feasible only under the circumstance of the data possessor requiring an access to the uploaded data as it can be decrypted by his/her private key. This approach hampers flexible sharing of data with multiple users as it necessitates sharing of keys by the genuine data possessor for decryption with authorized target users. Another alternative is to download the data, re-encrypt with intended recipient's public key and re-upload on cloud by data owner. This strategy incurs heavy processing overhead, extensive computation and communication cost at the data owner's site, defeating the very purpose of cloud computing. Proxy re-encryption (PRE) can deliver a propitious solution to the aforementioned problems.

One of the crucial factors for enabling fast and secure computations in the zettabyte era is the use of incremental cryptographic primitives. For files ranging from several megabytes up to hundreds of gigabytes, incremental cryptographic primitives can offer speedup factors measured in multiple orders of magnitude.

1.4.2 Research Objectives

On the basis of literature review and research gap mentioned above, the following objectives have been identified to be investigated under the research entitled "**An efficient framework for data security in mobile cloud computing paradigm**".

i. To study the existing data security schemes for mobile cloud computing.

ii. Designing and developing a secure framework for providing data security in mobile cloud computing environment.

iii. Verification and validation of proposed framework.

iv. Comparison and analysis of proposed framework with existing schemes.

1.4.3 Summary of Contribution(s)

To achieve the **first objective**, a comprehensive review and thorough study of existing security schemes/frameworks/protocols etc. has been done. These techniques include cryptographic solutions such as identity based, attribute based or biometric-based encryption, etc. Various cryptographic solutions including symmetric, asymmetric, proxy re-encryption and incremental cryptography have been reviewed. It was found that the computation cost (in terms of number of operations), running time (in Sec), energy consumption (in Joules), memory consumption are possible parameters to compare the performance of data security schemes. Asymmetric cryptography suffers from certificate management problem and Identity-based Encryption (IBE) suffers from the key escrow problem. In terms of computation overhead, pairing operation is the most expensive operation, among other public key operations, such as elliptic curve point multiplication, elliptic curve point addition, etc.

To fulfill the **second objective**, three algorithms have been proposed. The major contribution of the thesis is to reduce the number of computational operations so that the approach can be applied to resource-constrained mobile devices. The proposed algorithms are:

i. Cryptanalysis and improvement of Certificateless Proxy Signcryption Scheme for E-Prescription system in mobile cloud computing.

ii. Cryptanalysis and improvement of a pairing-free Certificateless Proxy Re-Encryption Scheme for secure sharing of mobile personal healthcare records in Cloud.

iii. A Secure Incremental Proxy Re-encryption for e-Healthcare Data Sharing in Mobile Cloud Computing.

The first proposed algorithm is based on Certifcateless cryptography and elliptic curve cryptography. Recently, a pairing free Certificateless Proxy Signcryption (CLPSc) scheme has been proposed by Yanfeng et al. [32] which claims its security against forgery under adaptive chosen-message attacks. In this thesis, it has been unveiled that the

aforementioned scheme has failed to provide unforgeability. As an improvement of their scheme, a novel pairing-free Certificateless Proxy Signcryption scheme using Elliptic Curve Cryptography (ECC) has been proposed for e-prescription system in mobile cloud computing. The proposed scheme is proven to be secure against indistinguishability under adaptive chosen-ciphertext attack and existential forgery under adaptive chosen-message attack in the random oracle model against Type 1 and Type 2 adversaries through formal analysis. The proposed scheme outperforms the existing schemes in terms of computational efficiency, making it suitable for futuristic mobile cloud computing applications.

The second algorithm is also based on certifcateless cryptography and ECC. First of all, cryptanalysis of Qin's scheme [33] is performed breaching confidentiality of their scheme. As an improvement to their scheme, a lightweight and pairing free single hop unidirectional Certificateless Proxy Re-encryption (CL-PRE) scheme based on elliptic curves has been proposed for secure sharing of mobile PHRs with public cloud competent for low-power mobile devices. The proposed scheme has been proved secure against chosen ciphertext attack in the random oracle model through formal analysis. Also, the proposed scheme is compared to existing proxy re-encryption schemes and found to be the most efficient.

The third algorithm is also based on certifcateless cryptography and ECC. A novel elliptic curve based pairing-free certificateless incremental proxy re-encryption scheme has been proposed for improving the file modification operations. The proposed scheme shows significant improvement in file modification operations in terms of turnaround time and energy consumption on mobile devices.

For **third objective**, verification of the proposed schemes is done by making use of mathematical theorems and lemmas. The same is validated using experimental simulation setup. The simulation setup includes the following:

i. Android SDK
ii. Firebase
iii. MIRACL cryptographic library [34]

For **fourth objective**, comparison of the proposed schemes is done with existing schemes in terms of computation time of various phases, size of first level ciphertext, second level ciphertext, re-encryption key length, turnaround time, energy consumed etc.

1.5 Thesis Outline

The thesis has been organized into 6 chapters. The content of each chapter is briefly described below:

Chapter 1 Introduction of thesis is divided into 5 sections. Section 1.1 gives an overview of Mobile Cloud Computing and its architecture, applications and challenges, followed by section 1.2 in which security issues are discussed in detail. Section 1.3 deals particularly with data security in MCC, security goals and classification of various security schemes. Section 1.4 provides significance of the research done as part of the doctoral thesis and motivation behind carrying this research. The brief overview about thesis organization is presented in Section 1.5.

Chapter 2 Literature Survey of thesis presents historical perspective and review of literature relevant to this research including survey of existing cryptographic solutions/frameworks for MCC in Section 2.1, followed by mathematical theory and preliminaries of ECC, bilinear pairing, computational problems etc. in Section 2.2. The strengths and weaknesses of various data security schemes are emphasized to provide better insight into research niche. The formal and security models of proxy signcryption and re-encryption primitives are discussed in Section 2.3 and 2.4 respectively.

Chapter 3 Proxy Signcryption of this thesis presents a pairing free Certificateless Proxy Signcryption for E-Prescription System in MCC which is based on elliptic curve discrete logarithm problem. Section 3.2 throws some light on existing proxy signcryption schemes, followed by a review and cryptanalysis of Yanfeng's et al. [32] pairing free certificateless proxy signcryption scheme in Section 3.3 and 3.4 respectively. The proposed CLPSc scheme is discussed in Section 3.5. For verification and validation of the proposed scheme, cryptographic library MIRACL is used. It also includes mathematical security proof in

Section 3.6. The comparative analysis has been made against existing protocols in Section 3.7.

Chapter 4 Proxy Re-encryption of this thesis presents a pairing free Certificateless Proxy Re-encryption (CLPRE) for secure sharing of mPHRs (Mobile Personal Health Records) in MCC. Section 4.1 gives an overview of proxy re-encryption primitive. Section 4.2 throws some light on the existing proxy re-encryption schemes in the literature, followed by a review and cryptanalysis of Qin's et al. [33] pairing free certificateless proxy re-encryption scheme in Section 4.3 and 4.4 respectively. The proposed CLPRE scheme is discussed in Section 4.5. For verification and validation of the proposed scheme, cryptographic library MIRACL is used. It also includes formal security analysis in Section 4.6. The comparative analysis has been made against existing protocols in Section 4.7.

Chapter 5 Incremental Certificateless Proxy Re-encryption of this thesis presents a pairing free Certificateless Incremental Proxy Re-encryption (CL-iPRE) for secure sharing of secure sharing of EHRs and m-PHRs in MCC. Section 5.1 gives an overview and need of incremental proxy re-encryption primitive. Section 5.2 throws some light on existing incremental proxy re-encryption schemes. The proposed CL-iPRE scheme is discussed in Section 5.3, followed by block updating operations in Section 5.4. The performance comparison has been evaluated in terms of turnaround time and energy consumption using MIRACL as cryptographic library, Android SDK for mobile cloud client application and Firebase for hosting web instant in Section 5.5.

Chapter 6 Conclusion and Future Scope of this thesis presents the conclusion drawn from the experimental results carried out in the thesis. Some indicators for the future research on the topic under consideration are also discussed briefly.

1.6 Summary

MCC will be the main information infrastructure in the near future consisting of various large, geographically distributed and heterogeneous data centers. Designing a secure data access for mobile cloud computing platform is a huge challenge. MCC inherits all the security issues that exist in cloud computing with an additional limitation of mobile

devices' resources like bandwidth, processing power, memory, etc. Consequently, many innovative security protocols and techniques have been developed to meet this challenge. There are several ways to provide security, one of them being cryptography. Selecting the appropriate cryptographic technique having low resource consumption holds the key for the future.

Chapter 2

Literature Review

This chapter investigates various cryptographic based data security frameworks and taxonomy of the state-of-the-art data security frameworks. Having dwelled into a brief insight into MCC, its challenges and applications, it is logical that MCC need lightweight cryptographic algorithms to achieve desired level of security with reduced computation and communication overhead. This chapter also presents an introduction of two novel cryptographic primitives, namely, Proxy Signcryption and Proxy Re-encryption and their formal and security models.

2.1 Cryptographic Techniques/Frameworks for MCC

Cryptography is a technique of storing and transmitting data over an insecure channel in a particular form so that data is intelligible only for intended recipient(s). It deals with constructing and analyzing protocols/algorithms that prevent third parties or adversaries from reading secret messages. Figure 2.1 summarizes the cryptographic preliminaries and building blocks associated with each data security scheme in MCC environment, proposed year and interdependencies between the schemes.

2.1.1 Symmetric Cryptography

It was developed in the 1970s in which encryption and decryption are done by common shared key [35] (as shown in Figure 2.2). It is relatively easy and fast to implement, but secret key agreement between parties is a problematic issue.

Some work/contents of this chapter have been peer reviewed and published in:

- *Tarunpreet Bhatia and A.K. Verma, "Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues", The Journal of Supercomputing, vol. 73, no. 6, pp. 2558–2631, 2017.*

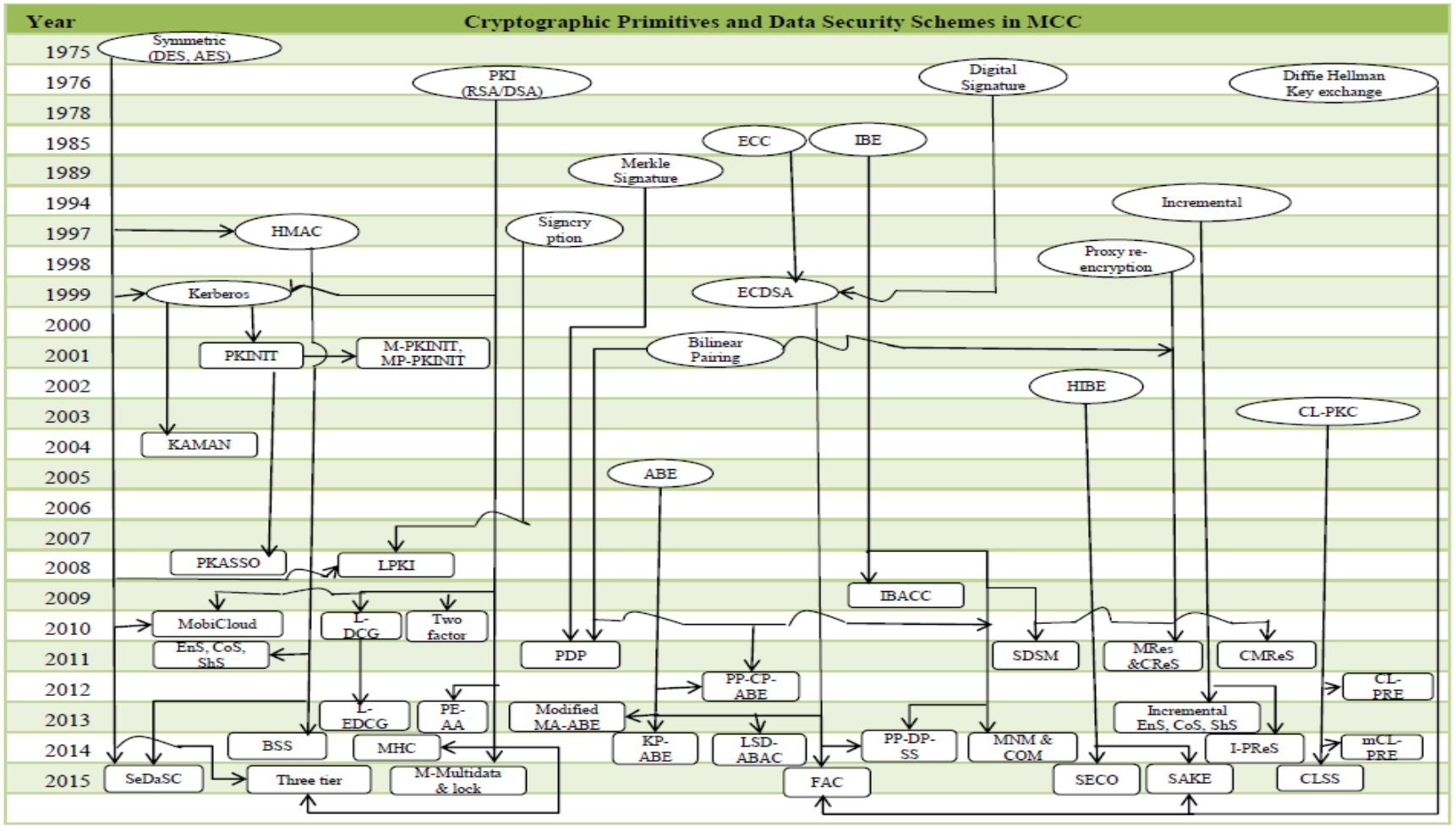

Figure 2.1 Cryptography preliminaries and data security schemes in MCC ⬭ represents cryptographic building blocks and ▭ represents data security protocols in MCC environment

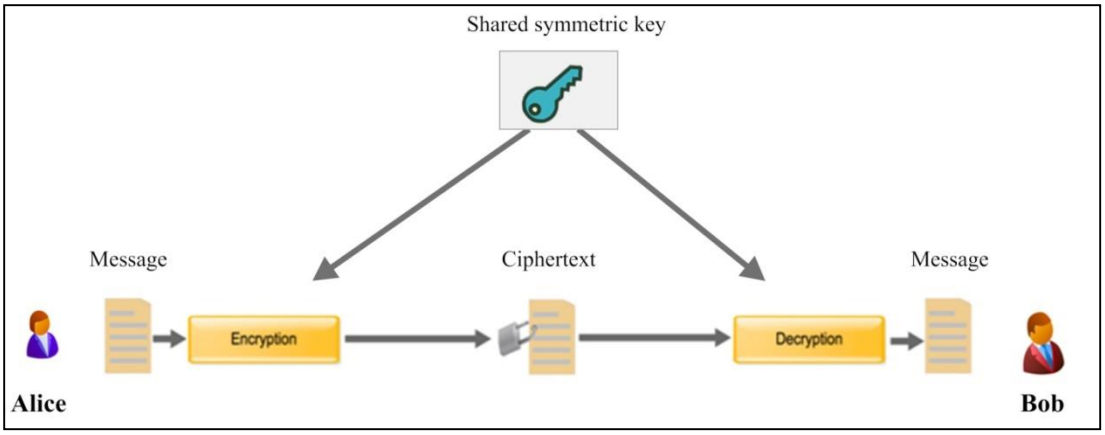

Figure 2.2: Symmetric cryptography

The robustness and efficiency of symmetric key depend on the key length used and secure transmission of keys between the parties. This cryptography fails to provide non-repudiation and authentication mechanisms. If an attacker is successful in compromising symmetric key, then both sides of conversation get exposed and all the messages encrypted with that key are exposed.

Kerberos is an authentication protocol based on symmetric key cryptography to allow workstations to share resources in a secure manner [36]. Since symmetric key operations incur less overhead, it provides shorter authentication latency as compared to PKI. In Kerberos, a trusted third party is responsible for certifying communicating parties to one another. The popular symmetric key algorithms [37] are Data Encryption Standard (DES), Advanced Encryption Standard (AES), blowfish etc. Message Authentication Code (MAC) relies on symmetric encryption to authenticate the originator of a message. MAC-based algorithms takes message and shared symmetric key as input and outputs a MAC value. This allows an individual to verify whether an adversary has tampered with the contents of a message or file. MACs can also be built from keyed hash functions (MD, SHA) known as HMAC [38]. Encryption ensures confidentiality, and MAC ensures integrity or authentication, so a combination of these cryptographic primitives is required to achieve the security principles.

Pirzada and Donald [39] proposed a secure key exchange protocol called KAMAN based on four-pass Kerberos applicable in mobile adhoc networks [40]. KAMAN employs

replication and election mechanisms to ensure stable and prolonged connectivity between clients and servers. It consists of multiple servers for authentication and load distribution and eliminates the use of the ticket granting server to make it suitable for mobile environment. Mobile clients know the password or secret key and servers store hash value of these passwords. The servers share a secret key among each other to replicate their databases periodically or on demand to withstand network or system failures. The mobile client C1 wants to communicate with C2 in a secure way so it sends authentication request to one of the server S1. S1 will reply back C1 with a ticket containing the session key for requested secure connection. C1 sends this ticket to C2. This ticket acts as a certificate to C2 that a C1 is really who it claims to be. The C2 acknowledges the ticket by sending a timestamp, and a secure communication session is established between C1 and C2 using session key provided by S1.

Ali et al. proposed secure data sharing in clouds (SeDaSC) [41] methodology applicable in both conventional and MCC environment based on symmetric encryption without the use of computation-intensive operations. It provides confidentiality, integrity, forward and backward access control. It involves three entities, namely the data owner, trusted third party known as a Cryptographic Server (CS) on the cloud and cloud for storage. The data owner sends the data file, the list of the users sharing the data file and the parameters required for generating an ACL to the CS. CS is responsible for encrypting, decrypting, key management and maintaining access control lists. CS generates the symmetric key and encrypts the data before storing on the cloud. CS splits the symmetric key into two key shares for each user: One share is transmitted to the user, and other key share is stored within the ACL related to the data file. Whenever a person leaves a group, CS removes its records from the ACLs of the related files without re-encrypting the files. As the whole key is not possessed by that member, he cannot decrypt any of the data files. The entire key is not shared with the group members by CS to thwart against insider attack so that malicious users within a group cannot decrypt, modify and re-encrypt the data.

Ren et al. [42] proposed three different distributed schemes: encryption-based scheme (EnS), coding-based scheme (CoS) and sharing-based scheme (ShS) for providing confidentiality and integrity of mobile users' files which are stored on cloud servers. The

three entities involved are mobile device, cloud server and user operating mobile device. These schemes assume that the device is semi-trusted, i.e., computations are trusted, but storage is not trusted, and stored data may be compromised. Cloud servers are distrusted, but wired/wireless link security between mobile device and cloud servers is provided by traditional MAC or IP layer protocols such as IEEE 802.11, IPSec, TLS, etc.

Khan et al. [43] proposed BSS for MCC in which files are logically divided into blocks and lightweight XOR operations are used for achieving confidentiality and extending the battery lifetime of the mobile node without compromising the security. A mobile user M will choose a secret password consisting of ASCII characters of length $'k'$. Each character of the password is encoded into binary string of length 8 bits which is further extended in number of steps say M. It also ensures the integrity of the uploaded file by using hash-based MAC. The mobile user first calculates integrity key IK by applying hash filenames on filename, extended password and file size. Thereafter, MAC is generated from file contents and IK generated above. The mobile user is finally responsible for uploading file, hash function and generated MAC to the cloud. When a mobile user wants to download a file from cloud, he will send a request along with $H(FN)$ for unique identification of file to CSP. The mobile user generates an extended password and integrity key again for decryption. The downloaded MAC is matched with newly calculated MAC, and their similarity ensures integrity of files.

2.1.2 Asymmetric Cryptography

Diffie and Hellman introduced the concept of another cryptography, popularly known as asymmetric or Public key cryptography (PKC) [44]. The two different keys are involved, a public key known to everyone used for encrypting and verifying digital signatures and corresponding private key owned by the receiver only used for decrypting and digitally signing the message, as shown in Figure 2.3. The private key has to be kept confidential while the public key is published freely.

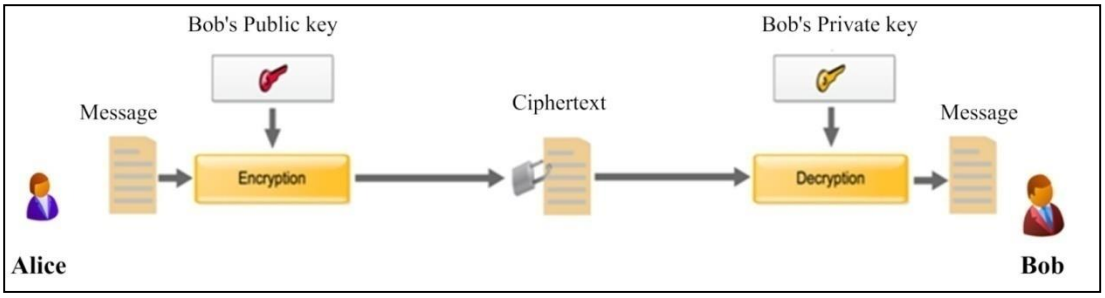

Figure 2.3: Asymmetric cryptography

2.1.2.1 Public Key Infrastructure (PKI)

PKI is a set of software, hardware, policies, procedures and people that work in collusion to facilitate the secure e-transfer of data over an insecure network. PKI authentication framework consists of public key certificates and digital signatures from a trusted certificate authority (CA) [45, 46]. The traditional authentication method based on user ID and password though cost-effective is inadequate for e-commerce transactions, so more rigorous proof is desired to confirm the identities involved in communication. In PKI authentication as shown in Figure 2.4, CA issues a secure digital certificate to the user which binds public keys with a user's identity and credentials for a given period of time. In such a way, trust in the user public key relies on one's trust in the CA issuing that digital certificate.

There are several variants of PKI-based cryptosystems such as RSA [47], ElGamal [48] and elliptic curve cryptography (ECC) [49]. Each of these cryptosystem can be used for encryption and digital signatures. There are inherent limitations of MAC or HMAC symmetric key algorithms for verifying authenticity. They fail to provide non-repudiation and involve a cumbersome key exchange procedure. Digital signatures employ asymmetric cryptography for providing authenticity of a message. Digital signature binds a sender to digital message using sender's private key which can be verified by the receiver using the sender's public key. The signature algorithm takes as input hash value of a message and sender's private key to produce cryptographic value known as digital signature as output. The message along with signatures is sent to receiver end for verification. Diffie and Hellman proposed the concept of digital signatures in 1976 [44]. Thereafter, many digital signature schemes were proposed such as RSA [47], Lamport signatures [50], Merkle

signatures [51], Digital Signature Algorithm (DSA) [52], ElGamal signatures [47], elliptic curve DSA (ECDSA) [53].

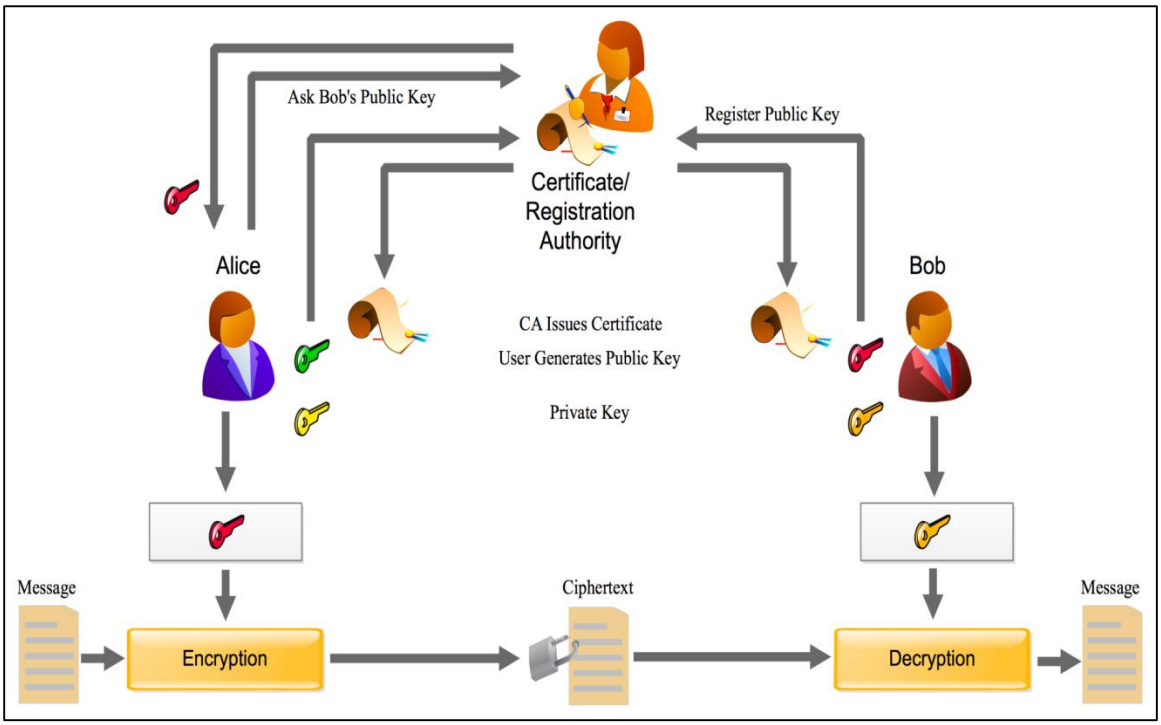

Figure 2.4: PKI based cryptosystem

Lee et al. [54] proposed a two-factor authentication framework for cloud computing based on PKI authentication and mobile out-of-band (OOB) authentication. Two-factor authentication scheme integrates two methods of authentication to enhance the level of assurance that a valid user has been authorized to access a service making it more difficult for unauthorized users to access the service. PKI provides the first step of authentication which is a combination of public/private keys, digital certificates and trusted third party. OOB adds another layer of authentication in which user's credential information and one-time random code are transferred to another party through OOB channel and sensitive data on another channel. This solution provides security against phishing and replay attacks preventing fraudulent users to access cloud services. Many online banking services are utilizing OOB authentication for verifying users' identities through a separate channel to prevent real-time threats. User provides PKI certificate and password to prove his/her identity to cloud server. The cloud server sends a certificate to corresponding CA for

verification. On receiving a positive reply from cloud server, it sends authentication request to the authentication server (AS). AS generates one-time authentication code using NLM-128 generator and sends code to the cloud server, user's mobile phone via a secure OOB channel and workstation. The user accepts the session if the code on the phone matches with the session code on the workstation. The cloud server also matches the code received from users and AS and grants access if a match occurs.

In MCC environment, an adversary can easily hack digital credentials such as a password or digital certificates. Xiao and Gong [55] proposed a first lightweight dynamic credential generation (L-DCG) scheme for identifying the mobile users based on randomly generated dynamic credential. Dynamic credentials are updated with mutual coordination between mobile user and cloud when a mobile user requested for a new communication channel or updates has reached a pre-decided threshold value. This scheme suffers from certain limitations. It assumes cloud as a fully trusted entity, and mobile user has to update dynamic secret frequently resulting in power consumption of mobile device.

Khan et al. [56] enhanced L-DCG [55] and proposed lightweight, enhanced dynamic credential generation scheme (L-EDCG) in MCC environment for protecting user's identity. L-DCG treats cloud as fully trusted component and allows mobile users to update dynamic credentials frequently, which may result in high processing overhead, communication delay and increased energy usage of mobile device. The proposed L-EDCG scheme involves 3 main entities, namely a mobile user or device, CSP and manager. It performs well in fully distrusted cloud environment and offloads credential generation operations to fully trusted entity called manager to reduce processing overhead on mobile devices. The manager entity is under full control of the organization and generates public-private key pairs, mobile and cloud secret and also current credential for each mobile user. The current credential is calculated by the manager as [57] and the communication between user and cloud occurs through the manager.

Khalid et al. [58] proposed a cloud-based secure and privacy enhanced authentication and authorization (PE-AA) protocol that allows anonymous communication. It comprises of 4 different servers, namely strong authentication server for authentication, authorization

sever based on XACML policy for authorizing and granting an access token to user for a particular service, identity management server for storing identity information and distributing identity information and keys, local certificate authority server for generating and verifying anonymous certificates. The protocol employed a Web application interface for communication between mobile users and different servers. This protocol is resilient against Web application attacks such as SQL injection, phishing, XSS, session hijacking, cookie tampering etc.

Annappaian and Agrawal [59] proposed a customer end multilevel cryptography with metadata and lock algorithm (M-Metadata & Lock) for secure data storage on the cloud. The number of levels and ways of encryption are pre-decided by customer based on its data confidentiality requirement and organizational structure. The authors have proposed 3-level and 3-way multilevel cryptography framework in which three levels are chief data officer, cryptography officer and data designers and three ways are data lock, data encryption and metadata, respectively. The responsibilities of three entities/levels involved are as follows:

- Chief data officer: Responsible for locking the data using some authentication mechanisms, such as passwords, biometric trait.
- Cryptography officer: Responsible for encrypting the data with private and public key cryptographic algorithms.
- Data designer: Responsible for determining and storing metadata, i.e., data pattern or structure of the data to be stored. The sensitive data is converted to metadata before encryption.

In 1985, Miller introduced the use of elliptic curves in PKC [49] but ECC algorithms such as Elliptic Curve Diffie–Hellman (ECDH) for key exchange, ECDSA were deployed after 2002. Before the introduction of ECC, most of the algorithms using PKC for encryption or digital signatures are based on RSA/DSA which incurs a heavy processing overhead with increasing the key length in providing better security. This overhead has affected electronic commerce sites to a large extent as they conduct number of secure transactions. ECC has

gained attention these days as it promises to provide equal security with much smaller key length, thereby reducing processing overhead [60].

Toorani and Beheshti proposed LPKI, a lightweight PKI for resource-constrained mobile platforms [61]. LPKI reduces computational overhead and communication cost by employing ECC and signcryption [62, 63]. ECC-based approaches are better than modular exponentiation-based systems. Traditional signature-then-encryption scheme digitally signs a message then encrypts the message, whereas signcryption scheme generates digital signatures and encrypts a message in one logical step with one pair of private-public keys assigned to entities. LPKI consists of six components such as registration authority for registering users and issuing them unique identifier, certificate authority for managing and issuing X.509v3 certificates, digital certificates for binding user's identity to his/her public key, certificate repository for storing certificates, Key Generating Server for generating public-private keys otherwise end entities are responsible for generating and storing key pairs on their smart cards or SIMs, online certificate status protocol [64] for handling inquiries about certificate revocation status, validation authority for performing validation tasks for end entities using Delegated Path Validation protocol [65] and timestamp server for generating valid timing information for other components.

2.1.2.2 Identity-based Encryption (IBE)

IBE, proposed by Shamir in 1985, is based on asymmetric cryptography. It allows an individual/sender to compute a public key of a user/receiver by combining public system parameters with unique identity information such as user's email address for encryption, and corresponding private key of receiver is generated by PKG, as shown in Figure 2.5.

Boneh and Franklin proposed first practical IBE system in 2001 [66]. It is an application of Weil pairings over elliptic curves and finite fields. Later in 2003, they introduced the concept of bilinear mapping and Weil pairing is an example of such a map [67]. Hierarchical IBE (HIBE) [68] is a generalization of IBE that maps an organization hierarchy. In HIBE, an identity at upper level issues private keys to identities at lower levels without compromising the security as it cannot decrypt messages intended for other identities.

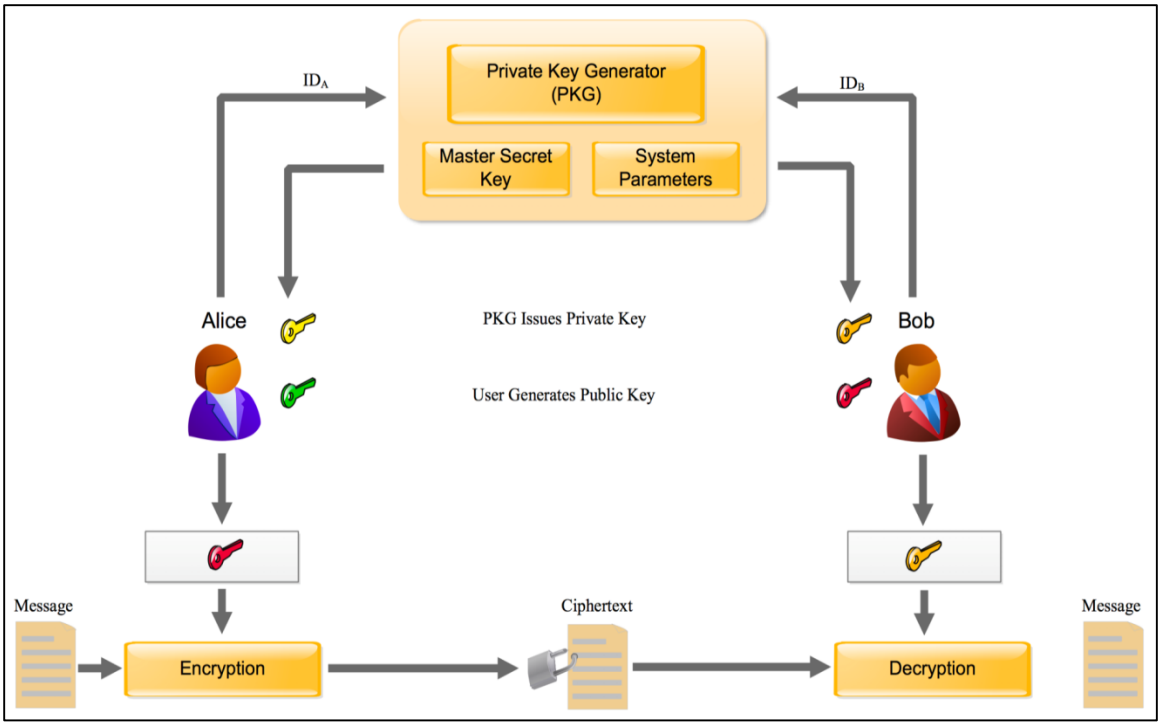

Figure 2.5: Identity based cryptosystem

Rajni et al. [69] proposed two models namely, Mobility Node Model (MNM) and a Centralized Owner Model (COM) based on IBE for privacy and security in mobile cloud environment. In MNM, data owner provides access to mobile client coming from the external environment via a proxy server that generates key shares without revealing clients' identity to the cloud. COM provides a centralized control mechanism which suits well for large networks where the same key is used for all mobile clients belonging to the same group, unlike MNM where a new key is used for all mobile clients increasing computation overhead. In COM, data owner is connected with trusted leaders of each group and requests of internal or external mobile client propagates through trusted leader to data owner and cloud. Li et al. [70] proposed a lightweight hierarchical Identity-based authentication protocol for cloud (IBACC) with corresponding IBE encryption and signatures. In IBACC, communication and computation cost of client is one IBE ciphertext and one IBE signature, while that of server is one IBE decryption and one IBE signature verification.

Dong et al. [71] proposed a secure and scalable data collaboration services (SECO) to allow availability of shared data among multiple users in a consistent manner. It leverages

multilevel HIBE for ensuring data confidentiality in an untrusted cloud environment. It allows one-to-many encryption, data writing operation, fine-grained access control, collusion resistance and backward secrecy. SECO consists of 5 different entities: root PKG (R-PKG) which stands at topmost level possesses a master key and generates private keys and system parameters for low level PKGs (L-PKG); L-PKGs request private keys from their upper level PKGs and generate corresponding private keys for lower level PKGs; domain PKG (D-PKG) requests private keys from upper L-PKG and generates private keys for their domain entities or users; users collaborate with each other to share data, receive private keys from D-PKG and interact with cloud server for storing, accessing and updating data dynamically, cloud server for providing storage-as-a-service to all domain entities. Each domain is comprised of D-PKG and multiple users working in collaboration. D-PKG keeps a user list which contains public keys of all authorized users within the domain. Within a domain, a user can encrypt data using multiple recipient public keys and system parameters before storing data in the cloud. The decryption can only be done by intended recipients and corresponding D-PKG using their private keys.

Liu et al. [72] proposed SAKE, a scalable authenticated key exchange for mobile e-health networks (MHN) which refines existing hierarchical identity-based signature and Diffie-Hellman key exchange scheme. In large-scale MHN, there are large number of hierarchies, such as NHIO (National Health Information Organization), RHIO (Regional Health Information Organization) of state, city, street and community and MHN users so there may be long paths between users belonging to different domains. SAKE employs virtual hierarchical network architecture of maximum three levels to reduce heavy communication overhead, mutual authentication costs and authentication path length. The key exchange process consists of 2 phases: Preparation phase involves processing at NHIO and RHIO and generates secret keys for mobile health network units, and handshake phase, which is run by individual mobile units generates a shared key for communication between individual mobile units.

2.1.2.3 Attribute-based Encryption (ABE)

ABE, one of the promising cryptographic techniques, was designed by Sahai and Waters in the year 2005 to achieve data confidentiality in a distributed cloud environment [73, 74]. Using ABE, the data owners can enforce fine-grained access policies, according to the nature of the data. In ABE, the ciphertext can be decrypted by a particular key only if the attribute set is validated by the access policy. ABE has been widely utilized in cloud-based storage scenarios for granting fine-grained access control [75, 76]. For example, an employer can upload an encrypted file on the cloud and defines the access policy of that file using the following static attributes and functions (NOT, OR, AND): "Manager" OR "Director" AND "Enterprise ABC". Hence, an employee who is a "Manager" or "Director" employed at "Enterprise ABC" can decrypt the file. It is further divided into two parts: key policy ABE (KP-ABE) and ciphertext policy ABE (CP-ABE) [77, 78]. In the KP-ABE scheme, the private keys of users are associated with an access policy while, the ciphertext is related to a set of attributes [78]. Decryption is allowed if the access policy defined in the private key matches with the attributes associated with the ciphertext. Yu et al. [79] proposed a fine-grained access control scheme for cloud storage utilizing KP-ABE. It introduces a minimal computation overhead on the data owner as computation-intensive tasks are delegated to cloud servers. However, this scheme incurs a significant computation cost in key generation and decryption phase. However, in CP-ABE, the situation is reversed. The ciphertext is encrypted with access policy and each user possesses a certain set of attributes which are associated with the private key of users [80]. A user can decrypt the ciphertext if he has respective private keys. CP-ABE is preferred in the MCC environment as it is closer to role based access control [81] model.

Huang et al. [82] proposed MobiCloud that provides a secure framework with enhanced connectivity and processing power to support MANET functions such as routing, information dissemination and trust management. MobiCloud utilizes cloud computing technology to create a virtualized MANET communication layer by transforming each physical mobile node to a service node on the cloud as ESSI (Extended Semi-Shadow Images) to address the communication and computational deficiencies of mobile nodes. MobiCloud launches a new service, Virtual Trusted and Provisioning Domain (VTaPD)

for isolating information flow belonging to multiple virtual security domains through programmable routers. Each mobile device is equipped with application manager and sensor manager. Application manager manages software agents (SAs) running on mobile device and cloud platforms, whereas sensor manager manages the sensing data about the device itself such as processor type, battery state, location information and information about neighboring mobile nodes such as neighbor's identity, link quality etc. Within each VTaPD, there are multiple SAs corresponding to each ESSI which are loaded/unloaded by node manager. The core component for providing Security-as-a-Service in MobiCloud comprises VTaPD manager and Trust Manager Server (TMS). MobiCloud resource and application manager construct VTaPDs as directed by VTaPD manager and TMS. VTaPD manager collects the context-aware sensed information from a mobile device and uses it for risk management and intrusion detection. MobiCloud TMS provides attribute-based key distribution and encryption for secure data access control, identity search and federation services.

Zhou and Huang [83] designed a privacy preserving CP-ABE (PP-CP-ABE) and attribute-based data storage (ABDS) for efficient data storage in MCC environment. In PP-CP-ABE, computation-intensive encryption and decryption related tasks are outsourced to CSPs without degrading the security level as compared to traditional CP-ABE in which operations are performed locally. It consists of data owner, data requester, Encryption Service Provider (ESP), Decryption Service Provider (DSP), Storage Service Provider (SSP) and Trusted Authority (TA). PP-CP-ABE provides encryption and decryption services to data owners and data requesters, respectively through a third party (ESP and DSP) with storage of encrypted data in SSP without revealing the data content and secret keys to them. ABDS allows lightweight and resource-constrained mobile devices to access and manage encrypted data stored on cloud through frequent upload, download and update operations. Yu et al. [84] devised a security framework for cloud computing relying on CP-ABE. But their framework requires users to reveal a part of secret key to the cloud, while PP-CP-ABE forwards blinded private keys to CSPs.

Li et al. [85] proposed modified Multi-Authority Attribute-based Encryption (MA-ABE) scheme for resource-constrained mobile devices. They have extended Chase and Chow's

scheme [86] for mobile users by introducing a semi-trusted cloud server between the mobile user and attribute authorities for offloading communications and computations from mobile users to this server. The semi-trusted cloud server interacted with the different attribute authorities on behalf of the mobile user and obtained the masked shared decryption keys. It combines all the keys and gets one masked key which can be unmasked by a mobile user only to decrypt the message. The cloud server cannot decrypt the message and determine the attributes of the mobile user, thus preserving the security and privacy of the user.

Lv et al. [87] proposed the first KP-ABE scheme for mobile cloud storage system which outsources encryption and decryption operations to cloud servers. It delegates the tasks of attribute-based access control and updating public keys, ciphertext and private keys to the cloud servers from attribute authority and mobile users without disclosing data contents. This scheme consists of 4 entities: Attribute Authority (AA), data owners, cloud server and mobile users. AA is responsible for generation public key and master key. Whenever a new user arrives, AA generates a partial component of private key SK_{p1} using master key and transmits to the user. Data owner is responsible for defining attribute set and encrypting data before outsourcing to the cloud. Cloud server stores the data and provide access to stored data. It also generates other component of the private key SK_{p2}. Cloud sever transmits a partially decrypted ciphertext using SK_{p2} to the mobile user instead of the whole ciphertext when attribute set of user satisfies the access tree. Mobile user further decrypts partially decrypted ciphertext to get plaintext using SK_{p1}.

Fei et al [88] proposed a Lightweight Static and Dynamic Attributes Based Access Control (LSD-ABAC) scheme for secure data access in mobile computing environment. This scheme includes both static and dynamic attributes such as location of a mobile user, risk level associated in using mobile app currently to enhance security by exploiting features of smart mobile devices. LSD-ABAC consists of 4 phases: setup, key issuing, encryption and decryption. The setup algorithm takes the security parameters as an input and outputs a bilinear group and a set of public-private key pairs for communication with attribute authorities. The semi-trusted cloud authority communicates with each attribute authority on behalf of user in key issuing phase. The data owner encrypts the file using set of

attributes maintained by attribute authorities as well as a set of real-time context attributes obtained from smart mobile device and uploads on cloud. The decryption algorithm takes the decryption credentials received from attribute authorities, hash value of context-related attributes, partial decryption key as received from semi-trusted cloud authority, ciphertext from the cloud as inputs and gives the original data as an output.

Dong et al. [89] designed a Privacy-preserving Data Policy Sharing Service (PP-DP-SS) based on CP-ABE and IBE techniques that ensure collision resistance, data confidentiality and privacy-preserving by not disclosing user attributes to the cloud. It consists of data owner, data consumer, cloud server and private key generator (PKG). Data owner refers to an enterprise whose data are stored on cloud, and data consumers are users having an access to stored data as decided by data owner and can decrypt the desired data using secret keys. It is the responsibility of data owner to compute secret key for each user based on all the attributes accessible to that user. PKG computes private keys for users and distributes private keys to users and corresponding public keys (user ID) to the data owner. A cloud server provides services to users and consists of authorized user IDs in the cloud. Each file has certain attributes attached to it. The users also possess certain attributes as provided by data owner and user ID. This scheme is an integration of four algorithms, namely system initialization that generates parameters for all system entities, an encryption phase in which data owner encrypts files and uploads to cloud, key generation in which data owner generates and distributes secret keys to each user through cloud after encrypting it with the public key of that user, and the decryption phase in which a user can decrypt data in cloud server if and only if he possess matched set of attributes.

Wu et al. [90] proposed secure and cost-effective fine-grained fuzzy access control protocol, which allowed small and medium enterprises to offload access control computations to the cloud. It relies on ABE to avoid communication between cloud and enterprise during the authentication stage. The proposed protocol consists of 3 stages: setup, user registration and user authentication stage. During the setup phase, enterprise and cloud server agrees upon shared secret key K using Diffie-Hellman key exchange protocol. The enterprise selects a signature key SK and signature verification key PK known to all. The user U with username UN, password pwd and set of attributes $ATTRIB$

chooses a public-private key pair PK_U, SK_U respectively during the registration stage and sends to an enterprise. Enterprise generates 2 signatures as follows:

$$S_1 = SIGN_{SK}(UN||PK_U||H(pwd||PSG(K||\text{UN}))$$

where PSG is pseudorandom generator and H is a cryptographic hash function

$$S_2 = SIGN_{SK}(UN||ATTRIB)$$

During the authentication stage, mobile user sends all above parameters to the cloud. The cloud server first verifies S_2 and then S_1. Thereafter, it sends a random nonce to the user which user sign using his private key and send back to the cloud. After verifying, cloud accepts the user.

2.1.2.4 Certificateless cryptography

Traditional PKI requires deployment of infrastructure for issuing and managing certificates to guarantee the authenticity of public keys. IBE eliminates need of certificates, but suffers from the key escrow problem as the KGC can access the private keys of all users. ABE suffers from both key escrow problem and revocation problem as the private keys of existing users have to be updated whenever a user is revoked. Al-Riyami and Paterson proposed a new cryptosystem called Certificateless Public Key Cryptography (CL-PKC) [91] to solve key escrow and key revocation problem.

CL-PKC involves Key Generation Center (KGC) for generating partial private keys to users using their identities and master key, as shown in Figure 2.6. KGC securely supplies partial private keys to users and has no access to actual private keys of users. A user combines some secret information with partial private key received to compute its actual private key and KGC's system parameters to compute its public key.

Xu et al. [92] proposed CL-PRE scheme for data sharing in the cloud. CL-PRE involves data owner, data recipients, cloud resident proxy server and storage server. This scheme lowers computation overhead of data owner by leveraging cloud for encryption and key management. The data owner is responsible for encrypting data using symmetric data encryption key before outsourcing to cloud and generating proxy re-encryption keys with all the recipients. The owner sends encrypted data, access control list and data encryption key encrypted with an owner's public key to the cloud. The cloud further transforms his

encrypted key to the re-encryption key which can be easily decrypted by recipient's private key.

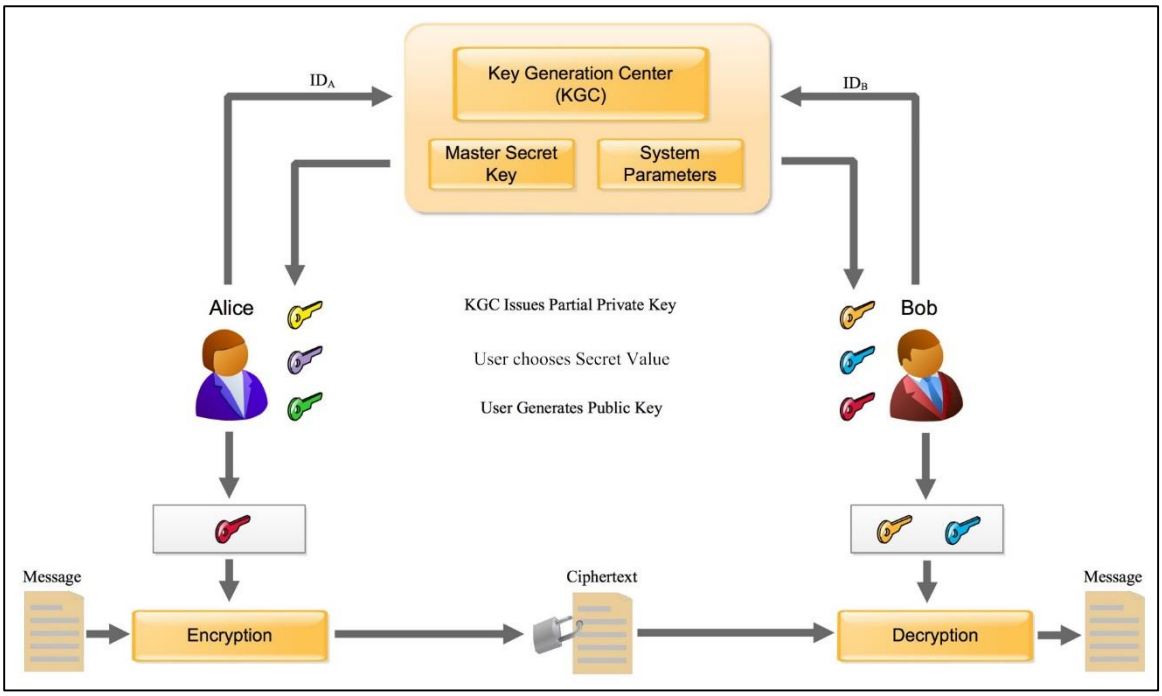

Figure 2.6: Certificateless cryptosystem

Seo et al. [93] proposed first mediated CL-PRE (mCL-PRE) for secure sharing of data in public clouds without pairing operations. This scheme involves data owner, user, cloud as storage center, KGC and security mediator. Each user possesses its own private key and public key pair. The user has to send its public key and identity to KGC in the cloud which generates two partial keys (SEM-key stored in the cloud security mediator and a U-key given to the user) and public key (KGC-key) for the user [94]. The public key is a pair of the user-generated public key and KGC-generated public key. The data owner encrypts the data using a random session key based on access control policies and then further encrypts the session key using his public key i.e. KGC-key. It further sends encrypted data along with ACL to be stored on the cloud. Whenever a user requests for data, security mediator verifies the user, retrieves the encrypted data from cloud and decrypts the data partially using SEM-key and user decrypts partially decrypted data fully using his private key and U-key.

Tsai proposed an efficient Certificateless Short Signature (CLSS) scheme [95] based on bilinear pairing suitable for devices with low-bandwidth environment. In CLSS, signature length is smallest and it is impossible for any certificateless scheme to generate signatures smaller than this length. The signature generation requires only one multiplication point operation, and verification requires one pairing and two multiplication point operations. Initially, KGC takes a secure parameter as an input and generates master private-public key pair and public parameters. It further computes partial private keys for each user based on its identity. The user chooses a random secret value and runs an algorithm to calculate its full public key. The user takes as inputs public parameters, its full public key, a message and returns a computed signature. The signature verifier algorithm, on the other side, takes the system parameters, the master public key, its full public key, received signature, message as its input and outputs true when the signature is valid and false otherwise.

2.1.3 Hybrid Cryptography

Symmetric cryptographic schemes have lower computational cost and faster encryption/decryption speed, but less secure as compared to asymmetric key schemes which are computationally expensive but more secure. The combination of these schemes can provide an efficient security solution. A hybrid cryptosystem including symmetric and asymmetric algorithms provides an efficient security solution without compromising on any of their features. PKINIT is an extension of Kerberos-based authentication [96] to allow public key-based authentication instead of symmetric key authentication between users and KGC to simplify key management issues in Kerberos. It integrates asymmetric key cryptography into the initial authentication exchange in pre-authentication data fields. The symmetric key cryptography is used afterward for better performance.

Harbitter and Menasce [97] proposed mobile, public key-enabled Kerberos protocol without and with proxy, namely M-PKINIT and MP-PKINIT which are lightweight version of PKINIT suitable for mobile platforms. They are a combination of public key-based Kerberos and Charon [98], which involves the use of proxy to assist mobile device in computation-intensive operations and develops trust relationship between mobile device and proxy. It enhances the security of Kerberos by involving minimal public key operations

and proxy for load distribution. The less expensive private key operations are performed on mobile devices and expensive public key operations on KGC to improve overall performance on a mobile platform. It differs from Kerberos, since the client generates a session key and encrypts it with the private key of KGC if proxy is absent or with the private key of proxy if present and signs it with its own secret key. KGC returns the standard Kerberos response to the client. It requires significant key operations when a mobile user moves to another Kerberos realm.

Park et al. [99] proposed PKASSO (PKI-based authentication based on Single Sign-On) protocol, an enhancement over the PKINIT protocol to solve resource- constrained problem of mobile devices by keeping the hardware and software less complex. PKINIT protocol failed to provide digital signature and non-repudiation. It suffers from high authentication latency as complex public and private key operations are required on mobile devices. PKASSO offloads the complex PKI operations from a mobile device onto a remote server having ample resources. PKASSO employed delegation server and referee server to provide SSO capability. The delegation server offloaded complex PKI operations on the behalf of mobile device and stored proxy certificates containing public and private keys signed by users. The referee server generated binding information between mobile devices and authentication messages which can be later used to provide non-repudiation mechanism. The authentication latency with PKASSO got improved to 0.082 second than conventional PKI-based, in which authentication latency is about 5.01 seconds.

Yang et al. [100] modified the Public verifiable Data Possession (PDP) scheme [101] for storage security in cloud computing for resource-constrained mobile devices to ensure privacy, confidentiality, integrity of mobile users' data stored on the cloud. It involves a trusted third party auditor for providing secure services like encryption, decryption, authentication and signature verification to overcome processing overhead on mobile devices. The mobile users just have to generate passwords and random numbers. The proposed scheme relies on Diffie-Hellman Key Exchange [102] for distributing symmetric keys over unsecured channel and thereafter, data files, asymmetric keys or secret information can be encrypted using this key and transferred safely. The bilinear mapping

[67] and Merkle Hash Tree [51] are used for providing integrity with minimum communication overhead and storage.

Nagaty [103] presented a secure mobile health application based on hybrid cloud platform which integrates cryptographic techniques with role-based access control to protect healthcare data of patients against intruders and authenticate valid users. It allows the mobile clients to access the information stored on cloud securely through token generation in three different access modes such as same hospital access when a doctor from the same hospital who owned a particular patient's medical data at a hospital wants to access it, cross-hospital access when a doctor from some hospital wants to access the medical data of a patient owned by another hospital, and third is emergency mode when a patient entered an emergency center and the doctor wants to access the patient's medical data owned by a hospital. IBE is used to generate patient's public and private key pairs. The patient's medical data and corresponding ACL containing patient id, access rights and patient's secret key are double-encrypted firstly by using patient public key and then using a hospital's public key. The signatures of patient's medical data file and access control list are generated using SHA1 or SHA2 and sent along with the files from hospital's private cloud server to public cloud server over secure transmission channel for storage.

Sujithra et al. [104] proposed a hybrid cryptographic approach for mobile device data security by outsourcing mobile data to remote cloud with minimal performance degradation. The three-tier hybrid approach has been designed. The first tier includes data encryption using MD5 algorithm, second tier encrypts the data again using AES algorithm, and third tier is twofold; either it re-encrypts data using ECC or key using the RSA algorithm. The comparative analysis of individual cryptographic algorithms such as DES, AES, ECC, MD5 [37] is done with hybrid approach, i.e., asymmetric with digital signatures or symmetric algorithm on both local environment and remote cloud environment.

2.1.4 Proxy re-encryption

It is a cryptographic primitive which enables the re-encryption of ciphertext from one secret key to another without knowing the actual secret keys. In a PRE scheme, a semi-honest proxy agent of Alice re-encrypts the ciphertext, on the message intended for Bob, on behalf

of Alice, without learning anything about the message (Figure 2.7). The PRE schemes are useful in the scenarios where data are desired to be shared with the authorized users over the cloud. A weaker form of proxy re-encryption is that proxy server possesses keys of both users': one for decrypting and other for encrypting. This was the only solution available till 1997. But Alice wants that Bob should be able to read the contents without revealing message contents and keys to trusted third party say proxy. Later, researchers suggested more efficient approaches for proxy re-encryption.

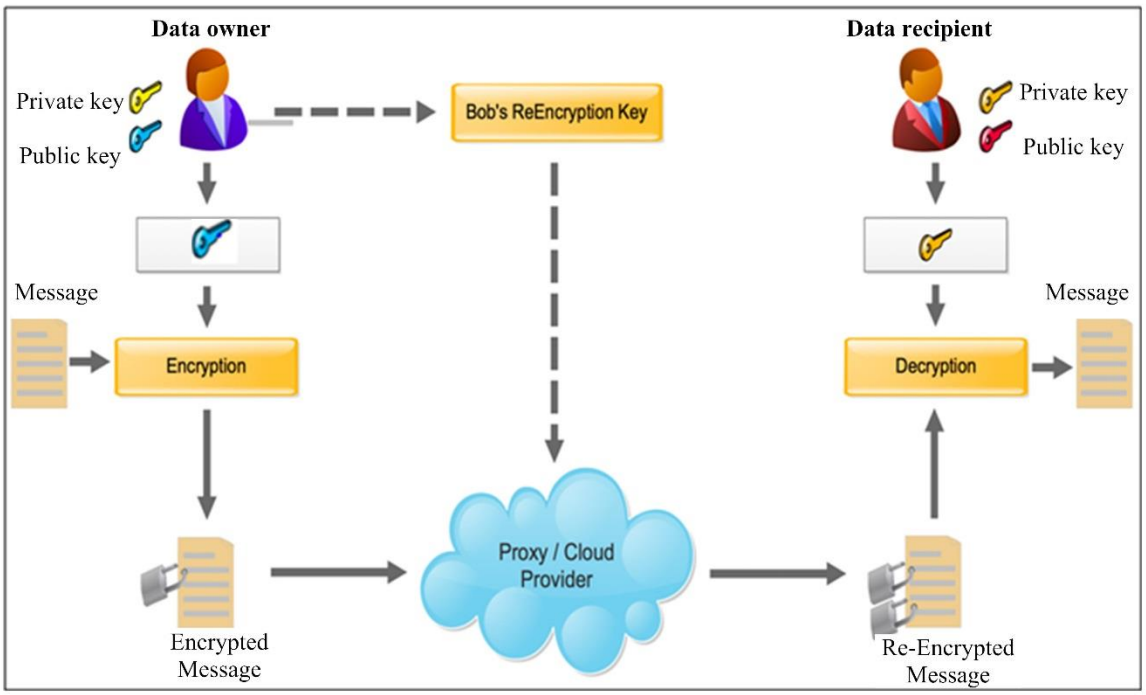

Figure 2.7 Proxy re-encryption

In 1998, Blaze et al. [105] proposed BBS (Blaze, Bleumer, and Strauss), an atomic re-encryption approach based on ElGamal cryptosystem [47] and re-encryption key $RK_{A\to B}$. Using $RK_{A\to B}$, a proxy can re-encrypt ciphertext without knowing the plain text. In order to compute $RK_{A\to B}$, one party must share his/her secret key with other party or rely on trusted entity in BBS scheme. Green and Ateniese [106] proposed first unidirectional and collision-resistant re-encryption scheme which avoids pre-sharing of any secret between communication parties. It is based on bilinear mapping. Therefore, proxy re-encryption is very useful in MCC environment as computationally intensive data access operations can

be offloaded from resource-constrained mobile devices to trusted third party such as proxy, manager or cloud.

Jia et al. [107] proposed Secure Data Service Mechanism (SDSM) which ensures data privacy as well as fine-grained access control with minimum communication overhead based on IBE and proxy re-encryption. It allows mobile users to encrypt the data with his identity to protect from data leakage and simultaneously outsourcing data and security management related task to the cloud without disclosing any information. It involves three entities data owner as a mobile device, data sharer and CSP rendering cloud services to the data owner. The data owner is responsible for sharing data or files and granting various access rights to data sharer. Data sharer and data owner communicate with the CSPs for data storage and retrieval. The cloud can grant access to an authorized user by re-encrypting the text with data sharer's identity which was earlier encrypted by data owner's identity. Data owner needs to forward re-encryption key with a size much smaller than the actual file to cloud and the rest of the re-encryption tasks are done on the cloud. SDSM is flexible enough to scale up as the number of data sharers increase.

Tysowski and Hasan [108] proposed manager-based re-encryption (MReS) and an efficient and scalable cloud-based re-encryption model to address the demands of mobile users. In MReS, all read requests initiated by mobile users are normally serviced through the manager which limits the scalability of the system. The manager under an organizational control maintains an access control list of authorized users and generates public and private keys for each user in the system and each partition on the cloud. During download request from user, manager has to generate re-encryption key and transformed ciphertext into re-encrypted ciphertext to be forwarded to the user. In CReS, the overall key management is done by client for security purpose, but computation-intensive data re-encryption is done by CSPs and key redistribution is kept minimal to reduce communication cost on mobile devices. All the members of the group receive a private key of partition initially through manager and are able to decrypt the message encrypted by any member of the group. When a new user enters the group and wants to access data, current partition key becomes invalidated and new version of the partition key is generated by applying a hash function

to a combination of current key and random salt. The manager then generates re-encryption key and sends it to a CSP to re-encrypt the data.

Khan et al. [109] proposed CMReS an enhancement over MReS and CReS. In CMReS, the data owner is able to download and decrypt the message directly without intervention of any trusted entity unlike manager in MReS. The mobile users are responsible for generating shared key pairs and publishing the public key information. The re-encryption responsibilities are offloaded to the cloud without disclosing any private keys in contrast to MReS and CReS. In CReS, the whole system gets affected if a particular group member is compromised because data partitions are encrypted by partition public key, unlike in CMReS where the message is encrypted by user's private key.

2.1.5 Incremental cryptography

Incremental cryptography involves designing of cryptographic algorithms in such a way that having applied algorithm once to a document, the result (say hash value) can be updated rapidly for a modified document without re-computing the result from scratch as shown in Figure 2.8.

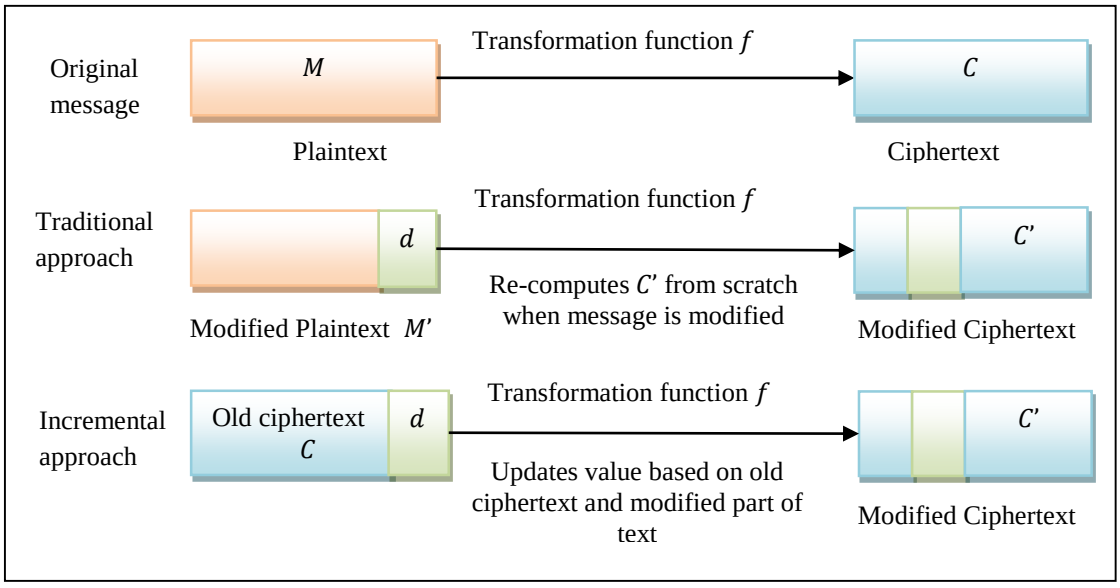

Figure 2.8: Incremental cryptography

Consider a modification in small part d of the original document M resulting in modified plaintext M'. In the traditional approach, transformation function is proportional to size of message M' whereas in incremental cryptography, transformation function is proportional to size of modified part d. The concept of incremental cryptography was given by Bellare et al. in 1994 [110] and later modified in 1995 using pseudorandom functions [111] and in 1997 using collision-free hash functions such as modular multiplication and addition [112]. So, it turns out to be a useful tool when the contents of a file change too frequently.

Itani et al [113] proposed first incremental cryptographic based energy-efficient protocol for mobile cloud computing environment that provides integrity of mobile users' data stored in the cloud based on a concept of proxy re-encryption and trusted computing. The trusted third party is responsible for installing coprocessors on a cloud which distribute secret keys and generate message authentication code for clients.

Khan et al. [57] proposed an incremental version of popular security schemes such as EnS, CoS and ShS for improving the block modification operations (insertion, deletion or updating) on resource-constrained mobile devices. The original file is divided into b blocks of equal size. The mobile user provides a password to be transformed into encryption and integrity keys as EnS [42] to provide confidentiality. DES and SHA-1 are used in EnS, and coding vector matrix is multiplied with blocks in CoS and XOR-based secret sharing in ShS for encryption and decryption tasks. The final encrypted file is generated by concatenation of individually encoded blocks and stored on the cloud. Similarly, final MAC of file is generated by applying a cryptographic hash function to the concatenation of individual blocks' MAC values. The mobile user uploads encrypted file, H(FileName), individual blocks' MAC and final MAC. It stores only file name and corresponding number of blocks in that file in its local memory. The proposed scheme is efficient for block modification operation. The mobile user encrypts only modified block not the entire file and generates a corresponding MAC. Thereafter, the mobile user has to specify a block number for modification and sends encrypted block and MAC to CSP. CSP re-calculates the final MAC of file based on the updation received from the mobile user.

Khan et al. [114] proposed Incremental Proxy Re-encryption (I-PReS) as an extension of BSS which uses the concept of incremental cryptography for improving the file modification operations without compromising the confidentiality and integrity services. The information about the file and the number of blocks into which a file is divided is stored locally on the mobile device. MAC for each block is calculated by applying hash function, and these codes are concatenated to calculate final hash code of the file. I-PReS shows significant improvement in terms of CPU processing overhead, energy consumption, turnaround time and memory storage of mobile device. I-PReS shows improvement in results in comparison to PReS while performing block deletion and modification operations.

2.2 Cryptographic Preliminaries

2.2.1 Elliptic Curve Cryptography (ECC)

Koblitz and Miller proposed ECC in 1985 which challenged RSA for both encryption and digital signatures as it incurs low processing overhead, making it viable for resource-constrained battery-powered mobile devices with limited computational and storage capabilities. ECC offers equal security level for smaller key lengths in comparison to RSA as shown in Table 2.1.

Table 2.1: Comparable key sizes for equivalent security levels (in bits)

Security Bits	Symmetric Encryption algorithm	RSA and Diffie-Hellman (modulus size in bits)	ECC Key (size in bits)
80	Skipjack	1024	160
112	3DES	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

There are two families of elliptic curves: binary curves $F(2^p)$ in which variables and coefficients all take on values in $F(2^p)$, best suitable for hardware applications and prime

curves over F_p (best suitable for software cryptographic applications) which consists of a set of points (x, y) which satisfies the Eq. (2.1):

$$y^2 \, mod \, p = (x^3 + ax + b) \, mod \, p \tag{2.1}$$

where $a, b \in F_p$ are integers in the range $[0, p-1]$ and Ð $=$

$$(4a^3 + 27b^2) mod \, p \neq 0$$

The corresponding points on curve form an additive group G with special point O (additive identity of the group) known as point at infinity. The two possible geometrical operations defined over elliptic curve groups are point addition and point doubling. Consider $P = (x_P, y_P)$,$-P = (x_P, -y_P)$, $Q = (x_Q, y_Q)$ and $R = (x_R, y_R)$ be points on the elliptic curve group such that R is a point of intersection of line PQ on curve E. The following operations can be performed:

$$P + O = P$$

$$P + (-P) = O$$

$$P + Q = R$$

The scalar multiplication of a point P on curve E, sP is performed by adding together s copies of point P, where random number $s \in Z_q^*$. A point P has order q if s is the smallest positive integer such that $qP = O$. Figure 2.9 shows point addition over an elliptic curve $y^2 = x^3 - 7x + 10$ in $\mathbb{R}$ with $P = (1, 2)$, $Q = (3, 4)$ resulting in $R = (-3, 2)$.

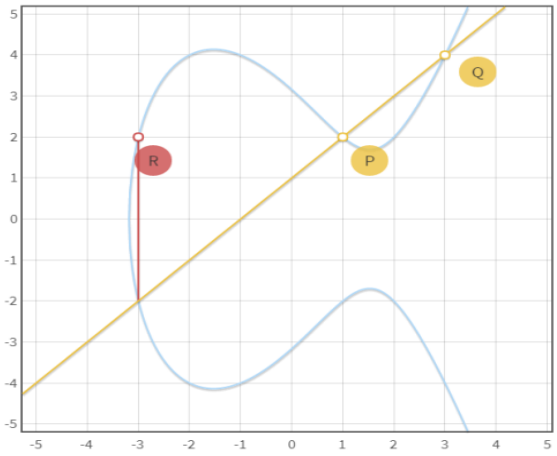

Figure 2.9: Point addition in elliptic curves

2.2.2 Computational Hardness Problems

When using standard cryptographic groups, the security of algorithms relies on hardness assumption.

Definition 2.1 *Discrete Logarithm Problem (DLP)*: Let G be a multiplicative cyclic finite group with generator g of order q. DLP in G states that given an element $X \in G$, find the smallest positive integer a such that $X = g^a$.

Definition 2.2 *Computational Diffie-Hellman Problem (CDHP)*: Let G be a multiplicative cyclic group with generator g of order q. Let $CDH: G\ X\ G \rightarrow G$ be a map defined by:

$$CDH(X, Y) = Z \text{ where } X = g^a, Y = g^b \text{ and } Z = g^{ab}$$

CDHP is to evaluate $CDH(X, Y)$ given $X, Y \in G$ without knowledge of $a, b \in Z_q^*$

Definition 2.3 *Elliptic Curve Discrete Logarithm Problem (ECDLP)*: It states that given two points $P\ and\ Q$ on elliptic curve group E defined over a prime finite field F_p with P as a generator of order q such that $Q = sP$ where $s \in Z_q^*$, computing s is intractable. The security of ECC is based on the discrete logarithm problem that is computationally infeasible to solve.

Definition 2.4 *Elliptic Curve Computational Diffie-Hellman Problem (EC-CDH)*: With the given two points A and B of elliptic curve E defined over a prime finite field F_p with P as a generator of order q, where $A = cP$ and $B = dP$ without knowing c and d, it is difficult to find out another point $K = cdP$.

2.3 Introduction to Certificateless Proxy Signcryption (CLPSc)

The traditional sign-then-encrypt approach provides confidentiality and authentication in two separate steps. Firstly, sender Alice signs a message using her private key for authentication and then encrypts the message and signature using symmetric encryption under randomly chosen session key which is further encrypted with recipient Bob's public key. At the receiver end, Bob verifies the signature, then decrypts the message. This method incurs more processing overhead and increases message length. A relatively new public key cryptographic primitive Signcryption was introduced by Zheng [115] in 1997,

which provides the encryption and digital signature in a single logical step with a cost lower than the traditional sign-then-encrypt approach. Signcryption schemes require only a single key pair for each user in comparison to traditional approach requiring two key pairs one for encrypting and one for signing. In signcryption scheme, sender Alice generates one time secret key by using the recipient Bob's public key for symmetric key encryption. Then the sender sends the ciphertext to recipient Bob. Bob after receiving ciphertext derives the same secret key using his private key.

Proxy signature is a signature scheme where an original signer delegates his/her signing ability to a proxy signer to enable the proxy signer to generate a signature on behalf of the original signer. Proxy signcryption is a novel cryptographic primitive that combines the functionality of a proxy signature with that of a signcryption. CLPSc is a technique to signcrypt a message to provide confidentiality and integrity by using the certificateless PKC concept. The primary benefit of CLPSc scheme is that it provides confidentiality, integrity and authentication and is free from key escrow problem.

2.3.1 Formal model of CLPSc

In general, a CLPSc scheme consists of 3 different entities: A as an original signcrypter, C as a proxy signcrypter, and B as a receiver. In proxy signcryption, the original signcrypter A delegates its signing authority to proxy signcrypter C by issuing a delegation warrant. The delegation or message warrant comprises of a message space, the identities of the delegator and the delegatee coupled with the validity epoch of the delegation. Publishing the signature on the warrant requires the delegator to sign the warrant with its private key. C then generates a signcrypted ciphertext at the behest of A and forwards it to B. B verifies the proxy signature and unsigncrypts the ciphertext. The CLPSc scheme consists of 10 polynomial-time algorithms which are as follows:

- **Setup:** This algorithm is executed by KGC. It accepts security parameter k as an input and generates the system parameters ψ and a master secret key s as outputs.

- **Extract-Partial-Private-Key:** In this algorithm, KGC accepts the system parameters ψ, the master secret key s and the identity ID_e of entity e as inputs and generates a partial private key S_e corresponding to the identity ID_e as an output.

- **Set-Secret:** This algorithm takes the system parameters ψ and the identity ID_e as inputs and generates the corresponding secret value k_e for entity e as an output.

- **Set-Private Key:** This algorithm accepts the system parameters ψ and the identity ID_e as inputs and generates the full private key $Sk_e = (S_e, k_e)$ for entity e as an output.

- **Set-Public Key:** This algorithm accepts system parameters ψ and user's private key $Sk_e = (S_e, k_e)$ as inputs and generates a public key pair $Pk_e = (R_e, P_e)$ corresponding to the private key Sk_e as an output.

- **Gen-Delegation:** This algorithm is executed by the original signcrypter A. It takes the security parameters ψ, the private key Sk_A and the message warrant $\mathcal{M}_w$ as inputs and generates as output the delegation $D_{A\to C}$ where C is designated as the proxy signer by the original signcrypter A.

- **Verify-Delegation:** This algorithm is executed by C for verifying the delegation $D_{A\to C}$ received from A.

- **Gen-Proxy-Key:** The proxy signer C executes this algorithm accepting the delegation $D_{A\to C}$ as an input. It generates the corresponding proxy key D_C for signcryption as an output.

- **Proxy-Signcryption:** This algorithm accepts the delegation $D_{A\to C}$, the system parameters ψ, the message warrant $\mathcal{M}_W$, the proxy key D_C and the receiver's public key as inputs and generates as output a signcrypted ciphertext for the proxy signer which is then further sent to the receiver B.

- **Proxy-Unsigncryption:** This algorithm is executed by the receiver B. It accepts the security parameters ψ, the message warrant $\mathcal{M}_W$, the signcrypted ciphertext, the receiver's private key, the original signer's and the proxy signer's public keys as inputs and outputs a plain text message m depending on the validity of the verification step.

2.3.2 Security model of CLPSc

In the CLPSc scheme, there are 2 different types of adversaries: Type 1 and 2 each with different capabilities as discussed below.

- **Type 1 Adversary $\mathcal{A}_1$:** It is a roguish entity which can replace public keys of various entities involved in a communication such as an original signcrypter, a proxy signcrypter or a receiver with the values of its choice but it can't access the master secret key.
- **Type 2 Adversary $\mathcal{A}_2$:** It is an inquisitive and malicious KGC, which possesses the master key but does not have permission to substitute a public key of any entity involved in a communication with a key of its own choice.

The following 8 oracles are accessed by adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$:

- **Create-User:** This oracle takes the identity ID_A of user A as an input. If ID_A already exists, nothing is to be done otherwise the oracle generates the user's private (S_A, k_A) and public key pairs (R_A, P_A) and then stores $(ID_U, S_A, k_A, R_A, P_A)$ into a list $\mathcal{L}$. It returns the public key (R_A, P_A).

- **Reveal-Partial-Private-Key:** This oracle probes list $\mathcal{L}$ to lookup an entry corresponding to the identity ID_A. If requested identity is not found, it returns nothing otherwise it returns the corresponding partial private key S_A.

- **Reveal-Secret-Key:** This oracle probes list $\mathcal{L}$ to lookup an entry corresponding to the identity ID_A and returns the corresponding k_A if matched otherwise returns null.

- **Replace-Public-Key**: Upon receiving the identity ID_A, this oracle probes $\mathcal{L}$ to lookup an entry corresponding to the identity ID_A. If found, it updates the public key pair of user A (ID_A, P_A, R_A) to a new value (ID_A, P_A^*, R_A^*).

- **Generate-Delegation:** This oracle takes the system parameters ψ, the original signcrypter $A's$ private key (S_A, k_A) and the message warrant $\mathcal{M}_w$ as inputs and generates a delegation as an output which is sent to proxy signcrypter.

- **Reveal-Proxy-Key:** This oracle generates a valid proxy key D_C for a given pair of identities (ID_A, ID_B) and the message warrant $\mathcal{M}_w$, if both ID_A and ID_B have been created and the corresponding public keys have not been replaced, otherwise null is returned.

- **Proxy-Signcrypt:** This oracle takes a plain text message m, the delegation from A, parameters from Reveal-Proxy-Key oracle, a public key of the intended recipient (R_B, P_B), and a private key of proxy signcrypter as inputs and generates a signcrypted message δ as an output.

- **Proxy-Unsigncrypt:** This oracle takes the signcrypted message δ, the message warrant $\mathcal{M}_w$, the system parameters ψ, the public keys and the identities of original and proxy signers, the identity and private key of the intended receiver as inputs and generates a decrypted plain text message v as an output if the signcrypted message is valid or an error message if it is invalid.

The two major security requirements for certificateless proxy signcryption primitive are unforgeability and confidentiality.

2.3.2.1 Unforgeability

The existential unforgeability against chosen message attacks (EUF-CMA) for certificateless proxy signcryption scheme is acquired by two games: Game 1 against Type 1 adversary $\mathcal{A}_1$ and Game 2 against Type 2 adversary $\mathcal{A}_2$.

$\boldsymbol{Game}^{\mathbf{1}}_{\boldsymbol{EUF-CMA}}$: A challenger and an adversary $\mathcal{A}_1$ participate in this game. It necessitates a level of difficulty for an adversary to masquerade a valid ciphertext for which it has not obtained any delegation warrant to sign.

- Initialization: On receiving a security parameter k as an input, the challenger executes Setup algorithm and generates the system parameters ψ and the master key as outputs. It then sends the system parameters to $\mathcal{A}_1$ and retains the master secret key with itself.

- Queries: $\mathcal{A}_1$ may carry out a polynomial bounded number of queries adaptively to the oracles of Create-User, Reveal-Partial-Private-Key, Reveal-Secret-Key, Replace-Public-Key, Generate-Delegation, Reveal-Proxy-Key, Proxy-Signcrypt and Proxy-Unsigncrypt.

- Forgery: Finally, the adversary $\mathcal{A}_1$ outputs a new signcryption δ^* on message $\mathcal{M}$ for ID_A^*, ID_B^* under the warrant $\mathcal{M}_w$, which is not composed by Proxy-Signcrypt oracle, unlike the normal case. The adversary $\mathcal{A}_1$ succeeds in the game if the outcome of the Proxy-Unsigncrypt oracle is not an error message and $\mathcal{A}_1$ has not queried both Reveal-Partial-Private-Key oracle and Replace-Public-Key oracle or Reveal-Secret-Key oracle of the original signer at any instant of the game.

$\boldsymbol{Game}^{\mathbf{2}}_{\boldsymbol{EUF-CMA}}$: This game engages the challenger and an adversary $\mathcal{A}_2$ in the CLPSc scheme as follows:

- Initialization: The challenger executes Setup algorithm and sends the system parameters ψ and the master secret key s to the adversary $\mathcal{A}_2$.

- Queries: $\mathcal{A}_2$ may carry out a polynomial bounded number of queries adaptively to the oracles of Create-User, Reveal-Secret-Key, Generate-Delegation, Reveal-Proxy-Key, Proxy-Signcrypt and Proxy-Unsigncrypt through the challenger.

- Forgery: Finally, the adversary $\mathcal{A}_2$ outputs a new signcryption δ^* on message $\mathcal{M}$ for ID_A^*, ID_B^* under the warrant $\mathcal{M}_w$, which is not composed by Proxy-Signcrypt oracle, unlike the normal case. The adversary $\mathcal{A}_2$ defeats the opposite party if the result of the Proxy-Unsigncrypt oracle is not an error message and $\mathcal{A}_2$ cannot extract user $A's$ secret value s_A at any instant of the game.

The CLPSc scheme is said to be EUF-CMA secure if no polynomial bounded adversary $\mathcal{A} \in (\mathcal{A}_1, \mathcal{A}_2)$ has an inconsequential advantage in winning the aforementioned games: $Game^1_{EUF-CMA}$ and $Game^2_{EUF-CMA}$.

2.3.2.2 Confidentiality

$Game^3_{IND-CCA}$ and $Game^4_{IND-CCA}$ capture the confidentiality requirement based on indistinguishability of encryptions under adaptively chosen ciphertext attacks (IND-CCA2) against the adversary $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively.

$\boldsymbol{Game^3_{IND-CCA}}$: This game involves the participation of the challenger and the adversary $\mathcal{A}_1$ in the CLPSc scheme:

- Initialization: On receiving an input k, the challenger executes the Setup algorithm and generates the system parameters ψ and the master key as outputs. It then sends the system parameters to $\mathcal{A}_1$ and retains the master secret key with itself.

- Phase-I: $\mathcal{A}_1$ adaptively may run a polynomial bounded number of challenger queries to oracles as in $Game^1_{EUF-CMA}$.

- Challenge: Once Phase-I is over, an adversary $\mathcal{A}_1$ submits 3 distinct identities: ID_A of original signcrypter, ID_C of proxy signcrypter and ID_B of verifier and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it desires to be confronted. The challenger chooses a random bit ß $\in \{0,1\}$ and proxy signcrypts $\mathcal{M}_ß$ to produce a signcrypted ciphertext δ^*. The challenger transmits δ^* as a challenge to the adversary $\mathcal{A}_1$.

- Phase-II: $\mathcal{A}_1$ can continue to probe the challenger with similar queries as in Phase-I.

- Output: $\mathcal{A}_1$ has to guess whether δ^* is signcrypted ciphertext of $\mathcal{M}_0$ or $\mathcal{M}_1$. For this, it generates an output ß′ ∈ {0,1} and succeeds in the game if ß′ = ß. $\mathcal{A}_1$ is not permitted to call Reveal-Partial-Private-Key oracle and Reveal-Secret-Key oracle on receiver's identity at any instant of the game. It is also not permitted to call the Proxy-Unsigncrypt query on the challenge signcrypted ciphertext under the identities ID_A, ID_B, ID_C unless public key of the original signcrypter, the proxy signcrypter or the verifier has been altered after the challenge was issued.

$\boldsymbol{Game^4_{IND-CCA}}$: This game engages the challenger and the adversary $\mathcal{A}_2$ under CLPSc scheme:

- Initialization: On receiving an input k, the challenger executes the Setup algorithm and sends the system parameters ψ and the master secret key s to $\mathcal{A}_2$.

- Phase-I: $\mathcal{A}_2$ adaptively may run a polynomial bounded number of challenger queries to oracles as in $Game^2_{EUF-CMA}$.

- Challenge: Once Phase-I is over, the adversary $\mathcal{A}_2$ submits 3 distinct identities: ID_A of original signcrypter, ID_C of proxy signcrypter and ID_B of verifier and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it desires to be confronted. The challenger chooses a random bit ß ∈ {0,1} and proxy signcrypts $\mathcal{M}_ß$ to produce a signcrypted ciphertext δ^*. The challenger transmits δ^* to the adversary $\mathcal{A}_2$.

- Phase-II: $\mathcal{A}_2$ can continue to probe the challenger with similar queries as in Phase-I.

- Output: $\mathcal{A}_2$ generates the output ß′ ∈ {0,1} and succeeds in the game if ß′ = ß. It is not allowed to extract a secret value of receiver at any instant of the game and make the Proxy-Unsigncrypt query as before.

The CLPSc scheme is said to be IND-CCA secure if no polynomial bounded adversary $\mathcal{A} \epsilon (\mathcal{A}_1, \mathcal{A}_2)$ has an inconsequential advantage in winning the aforementioned $Game^{3}_{IND-CCA}$ and $Game^{4}_{IND-CCA}$.

2.4 Introduction to Certificateless Proxy Re-encryption (CL-PRE)

Secure data sharing is an important concern when we have abundant e-health data stored in data centers. Proxy re-encryption (PRE) is latest solution to provide the security of this data before storing this to cloud. It is a cryptographic scheme for delegation of decryption rights. It is used when one user A wants to reveal the contents of message M encrypted with his public key Pk_A to another user B without revealing his private key Sk_A to user B. The foundation of classical public key cryptosystem includes certificate management and hence, is being considered computationally expensive. The introduction of an identity-based approach shifted the paradigm. Here, the public key is directly associated with the identity of the user and therefore, no further need of certificates. The evolution of certificateless public key cryptography eliminated the problem of key escrow in identity-based cryptosystems. The need of proxy re-encryption in various recently developed real-time applications leads to the combination of the concept of certificateless cryptography and proxy re-encryption. In CL-PRE, patients encrypt the data with their public keys before outsourcing to cloud and cloud resident semi-trusted proxy further re-encrypts into ciphertext under intended recipient's public key without learning anything about encrypted message.

2.4.1 Formal model of CLPRE

The general syntax of CL-PRE scheme consists of following 9 algorithms.

- **System setup:** This algorithm is run by KGC. It takes as an input security parameter k and outputs master secret key s, master public key P_{Pub} and a list of public parameters Ψ.
- **Extract partial key:** A deterministic algorithm run by KGC, which takes public parameters Ψ, cloud user identifier ID_u, master secret key s, master public key P_{Pub} as inputs and generates a partial public and private key pair (R_u, S_u) corresponding to user's identity ID_u.

- **Generate user key:** An algorithm run by every cloud user which takes public parameters Ψ, user's identity ID_u as inputs and outputs a user's secret key k_u and public key P_u.

- **Generate private key:** This algorithm is executed by every cloud user u with identity ID_u, accepts public parameters Ψ, partial private key S_u and user's secret key k_u as inputs and generates a full private key Sk_u for user u.

- **Generate public key:** This algorithm is executed by every cloud user u with identity ID_u, which accepts public parameters Ψ, partial public key R_u, user's secret key k_u and public key P_u as inputs and generates a full public key Pk_u for user u.

- **Encrypt:** The data owner $\mathcal{o}$ runs this algorithm for encrypting a plaintext message $m \in \mathcal{M}$ using its public key $Pk_{\mathcal{o}}$ and generates first level ciphertext $c_{\mathcal{o}} \in \mathcal{C}_1$ to be uploaded on cloud.

- **Generate re-encryption key:** This algorithm takes as inputs data owner's $\mathcal{o}$ public/private key pair $(Pk_{\mathcal{o}}, Sk_{\mathcal{o}})$ and data recipient's $\mathcal{r}$ public key $Pk_{\mathcal{r}}$. It is executed by the data owner to generate re-encryption key $Rk_{\mathcal{o}\to\mathcal{r}}$, which is forwarded to cloud resident proxy server $\mathcal{p}$.

- **Proxy re-encrypt:** It is executed by proxy server $\mathcal{p}$ which transforms first level ciphertext $c_{\mathcal{o}} \in \mathcal{C}_1$ received from the data owner $\mathcal{o}$ into second level ciphertext $c_{\mathcal{r}} \in \mathcal{C}_2$ for data recipient $\mathcal{r}$ using $Rk_{\mathcal{o}\to\mathcal{r}}$ or an error symbol Υ if $c_{\mathcal{o}}$ is invalid.

- **Decrypt:** This algorithm is executed by cloud user $u \in \{\sigma, r\}$ that takes ciphertext $c_u \in \mathcal{C}_{level}$ where $level \in \{1, 2\}$ corresponding to user u as an input and generates corresponding plain text message $m \in \mathcal{M}$ or an error symbol Υ if c_u is invalid.

The correctness of CL-PRE scheme requires the validity of following 2 equations.

$$Decrypt\left(\Psi, Sk_{\sigma}, Encrypt\left(\Psi, Pk_{\sigma}, ID_{\sigma}, m\right)\right) = m$$

$$Decrypt\left(\Psi, Sk_{r}, Proxy\ re-encrypt(\Psi, Rk_{\sigma \to r}, m)\right) = m$$

2.4.2 Security model of CLPRE

It has been assumed that two types of adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ exists with different capabilities which can breach confidentiality in CL-PRE scheme. Type 1 adversary $\mathcal{A}_1$ acts as an outer dishonest user who tries to steal information about an underlying message. Adversary $\mathcal{A}_1$ is not allowed to access master secret key s but is authorized to replace public keys of any entity with a valid value of its choice. Adversary $\mathcal{A}_2$ acts as an honest but curious KGC who has full access to s but is not allowed to replace public keys of any entity. Secure CL-PRE scheme must ensure that adversaries are not permitted to access the confidential message unless they have an authorized access to full secret key. The separate security models for original first level ciphertext and transformed second level ciphertext have been considered. The two security games $Game_1$ and $Game_2$ have been considered to prove security of CL-PRE scheme in which adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ interact with challenger $\mathbb{C}$ to violate confidentiality of the scheme.

2.4.2.1 Security against Type 1 adversary

Security of original first level ciphertext

In this section, original first level ciphertext security game played between adversary $\mathcal{A}_1$ and the challenger $\mathbb{C}$ is described. Before starting the game, $\mathbb{C}$ maintains a list of public keys of entities $\mathcal{L}_{Pk}$ and a list of corrupted identities $\mathcal{L}_{CI}$. $\mathcal{L}_{Pk}$ consists entries of the form $\{ID_i, Pk_i, \widetilde{Pk}_i\}$ where ID_i signifies user's identity, Pk_i signifies public key of entity i generated honestly by $\mathbb{C}$ through generate public key algorithm and $\widetilde{Pk}_i$ denotes public key

replaced by an adversary $\mathcal{A}_1$. An entity ID_i is called corrupted whose full secret key Sk_i has been extracted by $\mathcal{A}_1$, or public key has been replaced to $\widetilde{Pk_i}$ and partial key (R_i, S_i) has been extracted by $\mathcal{A}_1$.

- Initialization and Setup phase: $\mathbb{C}$ initializes $\mathcal{L}_{Pk}$ with $Pk_i = \widetilde{Pk_i}$ for all identities and $\mathcal{L}_{CI} = \phi$. $\mathbb{C}$ runs system setup algorithm with security parameter k and computes s and a list of public parameters Ψ. It gives Ψ to $\mathcal{A}_1$ and keeps s secret to itself.

- Phase-I: The adversary $\mathcal{A}_1$ can query oracles $\mathbb{O}_1, \mathbb{O}_2, \ldots, \mathbb{O}_m$ adaptively where$\{\mathbb{O}_j | \, j \in 1,2.., m\}$ is one of the following oracles and $\mathbb{C}$ responds to all the oracles correctly except the one in which $\mathcal{A}_1$ queries partial private key oracle $\mathbb{O}_{p_key}$ or user key extract oracle $\mathbb{O}_{u_key}$ for an identity whose public key has already been replaced by it.

    - Partial key extract oracle $\mathbb{O}_{p_key}$: This oracle takes $ID_i, \; s, P_{Pub}, \Psi$ as inputs and computes a partial public/private key pair (R_i, S_i) as an output by running extract partial key algorithm. If the adversary $\mathcal{A}_1$ has queried beforehand for user keys for corresponding identity ID_i, $\mathbb{C}$ adds ID_i to $\mathcal{L}_{CI}$.

    - User key extract oracle $\mathbb{O}_{u_key}$: Upon receiving ID_i as an input, it computes user's public/private key pair (P_i, k_i) and returns (P_i, k_i) to adversary $\mathcal{A}_1$. If $\mathcal{A}_1$ has queried beforehand for partial keys for corresponding identity ID_i, $\mathbb{C}$ adds ID_i to $\mathcal{L}_{CI}$.

    - Public key request oracle $\mathbb{O}_{Pk}$: Upon receiving input ID_i from $\mathcal{A}_1$, $\mathbb{C}$ searches list $\mathcal{L}_{PK}$ to check whether a tuple $\{ID_i, Pk_i, \widetilde{Pk_i}\}$ exists. If it exists, $\mathbb{C}$ returns Pk_i to $\mathcal{A}_1$; otherwise, it runs generate public key algorithm to compute public key Pk_i. $\mathbb{C}$ adds $\{ID_i, Pk_i, \widetilde{Pk_i}\}$ to $\mathcal{L}_{PK}$ and returns Pk_i to adversary $\mathcal{A}_1$.

- Private key extract oracle $\mathbb{O}_{Sk}$: Upon receiving ID_i as input, it calculates user private key (S_i, k_i) and returns (S_i, k_i) to adversary $\mathcal{A}_1$. $\mathbb{C}$ adds ID_i to $\mathcal{L}_{CI}$.

- Public key replace $\mathbb{O}_{Pk_rep}$: The adversary $\mathcal{A}_1$ can replace public key Pk_i corresponding to an identity ID_i with any value of its choice $\widetilde{Pk}_i$ by replacing the third component of the corresponding entry in $\mathcal{L}_{CI}$ with $\widetilde{Pk}_i$.

- Re-encryption key generate oracle $\mathbb{O}_{Rk}$: Upon receiving inputs $(ID_i, ID_j, \widetilde{Pk}_j)$ from $\mathcal{A}_1$, $\mathbb{C}$ runs generate re-encryption key algorithm where $\widetilde{Pk}_j$ is current public key of target identity to generate $Rk_{i\to j}$ which is returned to adversary $\mathcal{A}_1$.

- Re-encryption oracle $\mathbb{O}_{re-enc}$: Upon receiving inputs (ID_i, ID_j, C_i) where C_i is first level original ciphertext, $\mathbb{C}$ runs proxy re-encrypt algorithm to convert first level ciphertext C_i into second level ciphertext C_j using re-encryption key $Rk_{i\to j}$ and returns C_j to adversary $\mathcal{A}_1$.

- Strong decrypt-1 oracle $\mathbb{O}_{dec1}$: Upon receiving inputs (ID_i, C_i), $\mathbb{C}$ runs decrypt algorithm using corresponding private key and returns the result to adversary $\mathcal{A}_1$ or an error symbol Υ if the result is invalid.

- Strong decrypt-2 oracle $\mathbb{O}_{dec2}$:Upon receiving inputs (ID_i, ID_j, C_j), $\mathbb{C}$ runs decrypt algorithm using corresponding private key and returns result to adversary $\mathcal{A}_1$ or an error symbol Υ if result is invalid.

- Challenge phase: Once $\mathcal{A}_1$ decides Phase-I of $Game_1$ is over, it outputs target identity ID' and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it desires to be confronted. $\mathcal{A}_1$ must not query extract partial key $\mathbb{O}_{p_key}$ for ID' and re-encryption key generate $\mathbb{O}_{Rk}$ oracle for $Rk_{ID'\to ID}$, where ID is in list $\mathcal{L}_{CI}$ in Phase-I. ID' must not belong to list $\mathcal{L}_{CI}$. $\mathbb{C}$ chooses a random bit $ß \in \{0,1\}$ and

computes C' by running encrypt algorithm, which takes $(ID', \mathcal{M}_ß, \widetilde{PK'})$ as an input. The challenger $\mathbb{C}$ transmits C' as a challenge to $\mathcal{A}_1$.

- Phase-II: $\mathcal{A}_1$ continues to access aforementioned oracles except $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for ID'if $Pk_i \neq \widetilde{Pk}_i$. A challenge derivative [158] can be challenge ciphertext (ID_i, C_i) or ciphertext (ID_j, C_j) obtained by re-encrypting challenge ciphertext C_i to C_j. Thus (ID_i, C_i) and (ID_j, C_j) are challenge derivative if $\mathcal{A}_I$ has issued query to oracle $\mathbb{O}_{re-enc}$ to obtain C_j. In addition, $\mathcal{A}_1$ is not allowed to issue following queries, which makes it to win $Game_1$ trivially.

  - $\mathbb{O}_{p_key}$ and $\mathbb{O}_{u_key}$ for ID' if (ID', C') is a challenge derivative.
  - $\mathbb{O}_{Rk}$ for (ID_i, ID_j) if ID_j is in list $\mathcal{L}_{CI}$.
  - $\mathbb{O}_{re-enc}$ for (ID_i, ID_j, C_i) if (ID_i, C_i) is a challenge derivative and ID_j is in list $\mathcal{L}_{CI}$.
  - $\mathbb{O}_{dec1}$ for (ID', C_i) unless the public key used for creating challenge has been replaced.
  - $\mathbb{O}_{dec2}$ for (ID_i, C_i) if (ID_i, C_i) is a challenge derivative.

- Guess phase: The adversary $\mathcal{A}_1$ has to guess whether $C'$is ciphertext of $\mathcal{M}_0$ or $\mathcal{M}_1$ so it generates the output $ß' \in \{0,1\}$ and succeeds in the $Game_1$ if $ß' = ß$.

Definition 2.5 *Original first level ciphertext security*: A single hop unidirectional CL-PRE scheme is said to be (t, v) secure against adversary $\mathcal{A}_1$ in the original first level ciphertext security $Game_1$, if for any t time, the adversary $\mathcal{A}_1$ who makes at most $\mathbb{q}_{p_key}$ queries to $\mathbb{O}_{p_key}$, $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries to $\mathbb{O}_{Pk}$, $\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$, $\mathbb{q}_{Pk_rep}$ queries to $\mathbb{O}_{Pk_rep}$, $\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$, $\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$, $\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$ and $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$ in this game, we have $Adv_{A_1}^{CL-PRE_IND-CCA_ORG} \leq v$.

Security of transformed second level ciphertext

This section deals with transformed second level ciphertext security game played between the adversary $\mathcal{A}_1$ and the challenger $\mathbb{C}$.

- Initialization and setup phase: The challenger $\mathbb{C}$ runs setup algorithm as before and returns Ψ to $\mathcal{A}_1$ and keeps s secret to it. $\mathcal{A}_1$ now discloses list of identities whose full secret key is required. Thereafter, $\mathbb{C}$ initializes $\mathcal{L}_{CI}$. For each identity ID_i, $\mathbb{C}$ runs generate private key and public key algorithm and returns public/private key pair (Pk_i, Sk_i) to $\mathcal{A}_1$. $\mathbb{C}$ also initializes $\mathcal{L}_{Pk}$ with queried identities.

- Phase-I: Adversary $\mathcal{A}_1$ can query any of the aforementioned oracles except $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for ID_i if $Pk_i \neq \widetilde{Pk}_i$.

- Challenge phase: Once $\mathcal{A}_1$ decides that Phase-I is over, it outputs $(ID^\star, ID', \mathcal{M}_0, \mathcal{M}_1)$ on which it desires to be confronted, where $ID^\star$ is delegator's identity, ID' is target's identity, and $(\mathcal{M}_0, \mathcal{M}_1 \in \mathcal{M})$ are two equal length messages. In this phase, ID' should not be in $\mathcal{L}_{CI}$ and $\mathcal{A}_1$ has not queried $\mathbb{O}_{p_key}$ for ID' before the challenge phase, $\mathbb{O}_{Rk}$ from $ID^\star \rightarrow ID'$ and $\mathbb{O}_{Rk}$ from $ID' \rightarrow ID_i$, where $ID_i \in \mathcal{L}_{CI}$. $\mathbb{C}$ chooses a random bit $ß \in \{0,1\}$ and computes C' by running proxy re-encrypt algorithm, which takes $(ID', ID^\star, \mathcal{M}_ß, \widetilde{Pk}')$ as input. The challenger $\mathbb{C}$ transmits C' as a challenge to adversary $\mathcal{A}_1$.

- Phase-II: After the challenge phase, $\mathcal{A}_1$ continues to query aforementioned oracles except $\mathbb{O}_{p_key}$, $\mathbb{O}_{u_key}$ for ID_i if $Pk_i \neq \widetilde{Pk}_i$, $\mathbb{O}_{p_key}$ for ID'. It also cannot query $\mathbb{O}_{Rk}$ from $ID^\star \rightarrow ID'$ if $ID^\star$ is in $\mathcal{L}_{CI}$ and $\mathbb{O}_{dec2}$ for (ID', C'), unless the public key used for creating challenge ciphertext has been replaced.

- Guess phase: The adversary $\mathcal{A}_1$ has to guess whether C' is ciphertext of $\mathcal{M}_0$ or $\mathcal{M}_1$ so it generates the output $ß' \in \{0,1\}$ and succeeds in the $Game_1$ if $ß' = ß$.

Definition 2.6 *Transformed second level ciphertext security*: A single hop unidirectional CL-PRE scheme is said to be (t, v) secure against adversary $\mathcal{A}_1$ in the transformed second level ciphertext security $Game_1$, if for any t time, adversary $\mathcal{A}_1$ who makes at most $\mathbb{q}_{p_key}$ queries to $\mathbb{O}_{p_key}$, $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries to $\mathbb{O}_{Pk}$, $\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$, $\mathbb{q}_{Pk_rep}$ queries to $\mathbb{O}_{Pk_rep}$, $\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$, $\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$, $\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$ and $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$ in this game, we have $Adv_{A_1}^{CL-PRE_IND-CCA_TRF} \leq v$.

2.4.2.2 Security against Type 2 adversary

Type 2 adversary $\mathcal{A}_2$ models curious KGC, which has access to system parameters Ψ and s. This game is played between $\mathcal{A}_2$ and challenger $\mathbb{C}$. Since $\mathcal{A}_2$ has access to s, it can compute partial key of identity on its own so there is no need to access $\mathbb{O}_{p_key}$ oracle explicitly.

Security of original first level ciphertext

In this section, original first level ciphertext security game played between the adversary $\mathcal{A}_2$ and the challenger $\mathbb{C}$ is described. Before starting the game, $\mathbb{C}$ maintains a list of public keys of entities $\mathcal{L}_{Pk}$ and a list of corrupted identities $\mathcal{L}_{CI}$. An entity is corrupted if $\mathcal{A}_2$ has already queried $\mathbb{O}_{u_key}$ oracle for an identity ID_i.

- Initialization and Setup phase: $\mathbb{C}$ initializes $\mathcal{L}_{Pk}$ with $Pk_i = \widetilde{Pk}_i$ for all identities and $\mathcal{L}_{CI} = \phi$. $\mathbb{C}$ runs system setup algorithm with security parameter k and computes s and a list of public parameters Ψ. It gives Ψ and s to $\mathcal{A}_2$.
- Phase-I: The adversary $\mathcal{A}_2$ can query oracles $\mathbb{O}_1, \mathbb{O}_2, \ldots, \mathbb{O}_m$ adaptively, where $\{\mathbb{O}_j | \, j \in 1,2.., m\}$ is one of the aforementioned oracles and $\mathbb{C}$ responds to all the oracles correctly except the one in which $\mathcal{A}_2$ queries $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for an identity whose public key has already been replaced by it.

- Challenge phase: Once $\mathcal{A}_2$ decides Phase-I of $Game_2$ is over, it outputs a target identity ID' and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it desires to be confronted. $\mathcal{A}_2$ must not replace public key corresponding to the target identity and

query $\mathbb{O}_{Rk}$ oracle for $Rk_{ID'\rightarrow ID,}$ where ID is in list $\mathcal{L}_{CI}$. ID' must not belong to list $\mathcal{L}_{CI}$. $\mathbb{C}$ chooses a random bit ß $\in \{0,1\}$ and computes C' by running encrypt algorithm, which takes $(ID', \mathcal{M}_{ß}, \widetilde{PK'})$ as inputs. The challenger $\mathbb{C}$ transmits C' as a challenge to the adversary $\mathcal{A}_2$.

- Phase-II: $\mathcal{A}_2$ continues to access aforementioned oracles except $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for ID' if $Pk_i \neq \widetilde{Pk}_i$. $\mathcal{A}_2$ is not allowed to issue following queries, which makes it to win $Game_2$ trivially.

    - $\mathbb{O}_{p_key}$ and $\mathbb{O}_{u_key}$ for ID' if (ID', C') is a challenge derivative.
    - $\mathbb{O}_{Rk}$ for (ID_i, ID_j) if ID_j is in list $\mathcal{L}_{CI}$.
    - $\mathbb{O}_{re-enc}$ for (ID_i, ID_j, C_i) if (ID_i, C_i) is a challenge derivative and ID_j is in list $\mathcal{L}_{CI}$.
    - $\mathbb{O}_{dec1}$ for (ID', C_i) unless the public key used for creating challenge has been replaced.
    - $\mathbb{O}_{dec2}$ for (ID_i, C_i) if (ID_i, C_i) is a challenge derivative.

- Guess phase: The adversary $\mathcal{A}_2$ has to guess whether $C'$is ciphertext of $\mathcal{M}_0$ or $\mathcal{M}_1$ so it generates the output ß$' \in \{0,1\}$ and succeeds in the $Game_2$ if ß$' =$ ß.

Definition 2.7 *Original first level ciphertext security*: A single hop unidirectional CL-PRE scheme is said to be (t, v) secure against adversary $\mathcal{A}_2$ in the original first level ciphertext security $Game_2$, if for any t time adversary $\mathcal{A}_2$ who makes at most $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries to $\mathbb{O}_{Pk}$, $\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$, $\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$, $\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$, $\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$ and $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$ in this game, we have $Adv_{A_2}^{CL-PRE_IND-CCA_ORG} \leq v$.

Security of transformed second level ciphertext

This section demonstrates transformed second level ciphertext security game played between the adversary $\mathcal{A}_2$ and the challenger $\mathbb{C}$.

- Initialization and setup phase: The challenger $\mathbb{C}$ runs setup algorithm as before and returns Ψ and s to $\mathcal{A}_2$. $\mathcal{A}_2$ now discloses list of identities whose full secret key is required. Thereafter, $\mathbb{C}$ initializes $\mathcal{L}_{CI}$. For each identity ID_i, $\mathbb{C}$ runs generate private key and public key algorithm and returns public/private key pair (Pk_i, Sk_i) to $\mathcal{A}_2$. $\mathbb{C}$ also initializes $\mathcal{L}_{Pk}$ with queried identities.

- Phase-I: The adversary $\mathcal{A}_2$ can query any of the aforementioned oracles except $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for ID_i if $Pk_i \neq \widetilde{Pk}_i$.

- Challenge phase: Once $\mathcal{A}_2$ decides that Phase-I is over, it outputs $(ID^\star, ID', \mathcal{M}_0, \mathcal{M}_1)$ on which it desires to be confronted where $ID^\star$ is delegator's identity, ID' is target's identity and $(\mathcal{M}_0, \mathcal{M}_1 \in \mathcal{M})$ are two equal length messages. In this phase, ID' should not be in $\mathcal{L}_{CI}$ and public key of target identity should not be replaced. $\mathcal{A}_2$ has not queried $\mathbb{O}_{Rk}$ from $ID^\star \rightarrow ID'$and $\mathbb{O}_{Rk}$ from $ID' \rightarrow ID_i$, where $ID_i \in \mathcal{L}_{CI}$. $\mathbb{C}$ chooses a random bit ß $\in \{0,1\}$ and computes C' by running proxy re-encrypt algorithm, which takes $(ID', ID^\star, \mathcal{M}_ß, \widetilde{PK}')$ as inputs. The challenger $\mathbb{C}$ transmits C' as a challenge to adversary $\mathcal{A}_2$.

- Phase-II: After the challenge phase, $\mathcal{A}_2$ continues to query aforementioned oracles except $\mathbb{O}_{p_key}$ or $\mathbb{O}_{u_key}$ for ID_i if $Pk_i \neq \widetilde{Pk}_i$ and $\mathbb{O}_{p_key}$ for ID'. It also cannot query $\mathbb{O}_{Rk}$ from $ID^\star \rightarrow ID'$ if $ID^\star$ is in $\mathcal{L}_{CI}$ and $\mathbb{O}_{dec2}$ for (ID', C') unless the public key used for creating challenge ciphertext has been replaced.

Definition 2.8 *Transformed second level ciphertext security*: A single hop unidirectional CL-PRE scheme is said to be (t, v) secure against adversary $\mathcal{A}_2$ in the transformed second level ciphertext security $Game_2$, if for any t time, the adversary $\mathcal{A}_2$ who makes at most $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries to $\mathbb{O}_{Pk}$, $\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$, $\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$,

$\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$, $\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$ and $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$ in this game, we have $Adv_{A_2}^{CL-PRE_IND-CCA_TRF} \leq v$.

Chapter 3

Certificateless Proxy Signcryption

3.1 Introduction

With the advent of latest technologies in the healthcare industry, new benchmarks have been attained in the quality of service to the patients. However, the issue of security still needs to be addressed. Since a long time, the industry has been entangled in non-trivial issues of communicating health-related information in the safest possible manner with full consideration to confidentiality and authentication [116]. The concept of e-prescription has revolutionized the healthcare industry and has allowed health practitioners (doctors, physicians, pharmacists or nurses) to electronically transmit prescriptions to pharmacies directly. Traditionally, patients were examined and the diagnosis and prescriptions were made available to them in writing which has now been substituted by e-prescriptions. Often a patient is not in the situation to collect prescribed medicines from a pharmacy or obtain other medical services. A proxy delegated by the patient proves to be serviceable under such circumstances. The facets of confidentiality and non-repudiation in this service are taken care by one of the cryptographic primitives known as proxy signcryption. In case of any drug inappropriateness, the refutations on the doctor's part are prevented by the presence of his digital signature on the prescription. Such automation has not only improved patient's safety and quality of care but has also strengthened drug surveillance.

The genesis of proxy signatures was pioneered in 1996 by Mambo et al. [117]. Proxy signature, a variant of the digital signature, allows an original signer to entrust his signing

---

The major findings of this chapter have been peer reviewed and published in:

- *Tarunpreet Bhatia and A.K. Verma, "Cryptanalysis and Improvement of Certificateless Proxy Signcryption Scheme for E-Prescription System in Mobile Cloud Computing", Annals of Telecommunications, vol. 72, no. 9, pp. 563-576, 2017.*

authority to a third party known as a proxy signer for signing messages on his behalf. The efficiency of proxy signatures is greatly affected by the certificate-based PKC due to an additional overhead involved in the administration of certificates. In 1984, Shamir [118] laid the foundations of IBE to overcome the inherent problems of key generation, storage and management in PKC. The sender directly utilizes the receiver's identity for encrypting messages instead of looking for public keys and certificates [119]. Following this, many publicly verifiable identity-based proxy signature schemes [120-122] came into existence, but they failed to meet the security goals of unforgeability and undeniability. IBC still faces the key-escrow problem in which KGC creates private keys of all the users and has full supremacy in deciphering any ciphertext or signing any message at the behest of any entity. Certificateless cryptography was conceptualized by Al-Ryami and Paterson [91] to solve the key-escrow issue in IBE in 2003. KGC in cooperation with user generates the private key of the user; hence it can withstand malicious KGC attack. Due to an inadequacy of public key authentication, adversaries in certificateless PKC can reconstitute public key of any entity as opposed to the traditional certificate based PKC. Thus, there is a need to ensure that certificateless schemes are unable to forge the digital signatures that masquerade the verification algorithm with fake public keys.

3.2 Related Work

In 1997, Zheng [123] introduced the concept of signcryption (simultaneous signature and encryption); which allows an intended recipient to verify the signatures on confidential messages. Unlike the traditional sign-then-encrypt approach, signcryption primitive saves computational and bandwidth load. Barbosa and Farshim [124] integrated certificateless primitive with a signcryption scheme in 2008 and gave a formal notation. However, its performance was not acceptable. Several certificateless signcryption schemes have been proposed in the literature [125-131], but a majority of them either involve expensive bilinear pairing operations or are prone to forgery or confidentiality attacks. The computational cost of a bilinear pairing operation is roughly 10 folds more than that of the scalar point multiplication over elliptic curves [127]. Jing [128] proposed a certificateless signcryption scheme without pairing and proved that his scheme outperformed Selvi et al's scheme [129] and Xie's scheme [130] in terms of modular exponential operations. Shi et

al. [131] proved that Jing's scheme [128] neither provides unforgeability nor confidentiality against a Type 1 adversary and devised a new certificateless signcryption scheme. Gamage et al. [132] introduced a proxy signcryption scheme based on discrete logarithm in 1999 but it lacked any formal security analysis. Jung et al. [133] later pointed out that Gamage et al's scheme [132] violated the forward secrecy. Since then, many proxy signature schemes and signcryption schemes have been proposed [134-139, 32] but a majority of them rely on the bilinear pairings except Yanfeng et al.'s scheme [32]. Elkamchouchi and Abouelseoud [134] proposed an identity-based proxy signcryption scheme based on the bilinear pairings over elliptic curves for partial delegation of signing rights along with a revocation protocol for dishonest agents and a renewal procedure for proxies of honest agents. Wang et al. [135] proposed a pairing based self-certified proxy signcryption scheme for designing secure mobile agent protocols for commercial mobile applications. It reduced storage and communication overheads significantly in contrast to the certificate-based public key systems. Lin et al. [136] proposed a proxy signcryption scheme which involved one pairing operation in signcryption phase and three pairing operations in unsigncryption phase. This scheme was proved to be vulnerable to a chosen warrant attack by Lo and Tsai [137] and they further proposed a secure proxy signcryption scheme relying on the bilinear pairing operations. They proved that their scheme was more efficient in comparison to Elkamchouchi's [134] and Lin et al.'s [136] schemes in terms of the number of pairing operations involved in various phases. Yanfeng et al. [32] were first to propose a pairing-free CLPSc scheme but without a formal security analysis. Ming et al. [138] proposed proxy signcryption scheme that outperformed Lin et al.'s scheme [136] while having no pairing operation in signcryption phase and four pairing operations in unsigncryption phase. Zhou [139] proposed a publicly verifiable proxy signcryption scheme resistant to insider and proxy key exposure attacks. Both of these schemes have been proved secure in the standard model. Zhou's scheme involved only 8 system public parameters in setup stage, in contrast to 2m+5 in [138] where m represents the bit-length of an identity, making it suitable for a limited storage application environment such as smart cards. Yeh [140] proposed a proxy credential forgery attack on Elkamchouchi's [134] and Lin et al.'s [136] schemes and a simple modification to these two schemes to prevent the forgery attack without increasing any computational overhead.

A large number of CLPSc schemes have been quoted in the literature, however, design of CLPSc schemes stands neglected. Such schemes have a dual benefit of saving storage space and communication bandwidth as a message needs not to be sent separately with its signature. This thesis proposes a forgery attack on existing pairing-free CLPSc proposed by Yanfeng et al. [32]. An improvement to their scheme is provided to prevent the forgery attack. The proposed CLPSc scheme meets the security goals of confidentiality, authentication, integrity and non-repudiation. The proposed scheme also avoids bilinear pairing operations, making it more efficient for resource-constrained mobile devices being deployed in the MCC paradigm. To the best of our knowledge, there has not yet been a provable secure pairing-free CLPSc scheme.

3.3 Review of Yanfeng et al's scheme

Yanfeng et al. [32] proposed a pairing-free proxy signcryption scheme which consists of 10 polynomial time algorithms as below:

- **Setup:** KGC chooses a security parameter k and an elliptic curve E/F_p over prime finite field F_p. Let G be a cyclic subgroup of the elliptic curve group with P as a generator of order q. This algorithm returns system parameters $\psi = \{E, F_P, G, P, \mathcal{H}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, P_{Pub}\}$ for publishing. It chooses a master secret key $s \in Z_q^*$ and computes a master public key $P_{Pub} = sP$. The message space is $\{0,1\}^l$. It also chooses 4 cryptographic collision-free hash functions as follows:

$$\mathcal{H}_1: \{0,1\}^* \times G \rightarrow Z_q^* \qquad (3.1)$$
$$\mathcal{H}_2: \{0,1\}^* \times G \times G \rightarrow Z_q^*$$
$$\mathcal{H}_3: \{0,1\}^* \times G \times G \rightarrow \{0,1\}^l$$
$$\mathcal{H}: \{0,1\}^* \rightarrow Z_q^*$$

- **Extract-Partial-Key:** KGC randomly chooses $r_A \epsilon Z_q^*$ and computes $R_A = r_A P$ and $h_A = \mathcal{H}_1(ID_A, R_A)$. It further computes $S_A = (r_A + h_A s) \bmod q$ as $A's$ partial private key and transmits (S_A, R_A) to an entity A.

- **Set-Secret:** Each entity A verifies whether $S_A P = R_A + h_A P_{Pub}$. After successful verification, each entity A randomly picks $k_A \in Z_q^*$, computes $P_A = k_A P$ and sets k_A as its secret value.

- **Set-Private-Key:** Given the security parameters ψ, the partial private key S_A and the secret value k_A of the entity A as inputs, this algorithm generates a private key pair $Sk_A = (S_A, k_A)$ as an output.

- **Set-Public-Key:** This algorithm generates a public key pair $Pk_A = (R_A, P_A)$ of the entity A as an output.

- **Gen-Delegation:** The algorithm accepts the original signcrypter $A's$ secret key pair (S_A, k_A), its public key pair (R_A, P_A) and the message warrant $\mathcal{M}_W$ as inputs and generates the delegation $D_{A\to C}$ on $\mathcal{M}_W$ as an output. The original signcrypter A randomly chooses $a \in Z_q^*$ and computes $K = aP$. It then calculates y as Eq. (3.2):

$$y = (a + h_1 S_A + h_2 k_A) \tag{3.2}$$

where $h_1 = \mathcal{H}_2(\mathcal{M}_W || ID_C, K, R_A)$ and $h_2 = \mathcal{H}_2(\mathcal{M}_W || ID_C, K, P_A)$

The original signcrypter sends the delegation $D_{A\to C} = (ID_A, ID_C, \mathcal{M}_W, R_A, P_A, K, y)$ to the proxy signcrypter C.

- **Verify-Delegation:** The proxy signcrypter C verifies the delegation by computing $h_1 = \mathcal{H}_2(\mathcal{M}_W || ID_C, K, R_A)$, $h_2 = \mathcal{H}_2(\mathcal{M}_W || ID_C, K, P_A)$ and $h_A = \mathcal{H}_1(ID_A, R_A)$. It further checks whether $yP = h_1(R_A + h_A P_{Pub}) + K + h_2 P_A$. If not, it rejects the delegation.

- **Gen-Proxy-Key:** The proxy signcrypter C after verifying the delegation successfully, computes a proxy signing key $D_C = (y + h_3 S_c + h_4 k_c)$ where $h_3 = \mathcal{H}_2(\mathcal{M}_W || ID_A, K, R_C)$ and $h_4 = \mathcal{H}_2(\mathcal{M}_W || ID_A, K, P_C)$.

- **Proxy-Signcryption:** The proxy signcrypter C takes the proxy key D_C, a message m and the public key of the receiver (R_B, P_B) as inputs and generates a signcrypted message for the intended recipient B as an output. The detailed process is as

follows: C randomly chooses two random numbers $u \epsilon Z_q^*$ and $v \epsilon Z_q^*$ and computes $U = uP,\ V = vP,\ R_1 = u(R_B + h_B P_{Pub})$ and $R_2 = vP_B$ where $h_B = \mathcal{H}_1(ID_B, R_B)$. C generates proxy signcryption of message m as follows:

$$z = m \oplus \mathcal{H}_3(ID_A||ID_C||ID_B, R_1, R_2) \quad (3.3)$$

$$S = (u + hD_C) \quad (3.4)$$

$$\text{where } h = \mathcal{H}(\mathcal{M}||ID_A||ID_C||ID_B)$$

C sends the signcrypted text $(ID_A, R_A, P_A, \mathcal{M}_W, ID_C, R_C, P_C, U, V, K, z, S)$ to the receiver B.

- **Proxy-Unsigncryption:** The receiver B receives the signcrypted ciphertext $(ID_A, R_A, P_A, \mathcal{M}_W, ID_C, R_C, P_C, U, V, K, z, S)$. It computes:

$$R_1 = S_B U,\ R_2 = k_B V \quad (3.5)$$

$$m = z \oplus H_3(ID_A||ID_C||ID_B, R_1, R_2) \quad (3.6)$$

The receiver B accepts the message m if Eq. (3.7) holds otherwise, it returns an error message.

$$SP = h[K + h_1(R_A + h_A P_{Pub}) + h_2 P_A + h_3(R_C + h_C P_{Pub}) + h_4 P_C] + U \quad (3.7)$$

where

$$h_A = \mathcal{H}_1(ID_A, R_A)$$

$$h_C = \mathcal{H}_1(ID_C, R_C)$$

$$h_1 = \mathcal{H}_2(\mathcal{M}_W||ID_C, K, R_A)$$

$$h_2 = \mathcal{H}_2(\mathcal{M}_W||ID_C, K, P_A)$$

$$h_3 = \mathcal{H}_2(\mathcal{M}_W||ID_A, K, R_C)$$

$$h_4 = \mathcal{H}_2(\mathcal{M}_W||ID_A, K, P_C)$$

$$h = \mathcal{H}(m||ID_A||ID_C||ID_B)$$

3.4 Cryptanalysis of Yanfeng et al's scheme

This section analyzes and proves that Yanfeng et al.'s scheme [32] cannot withstand a forgery attack by Type 2 adversary. The Type 2 adversary $\mathcal{A}_2$ forges a valid signature $(z',\ S')$ on its chosen message m' by performing the following steps.

1. The adversary $\mathcal{A}_2$ randomly chooses three random numbers $k', u', v' \in Z_q^*$ and computes $K' = k'P, U' = u'P, V' = v'P$.

2. There are two ways in which adversary $\mathcal{A}_2$ can obtain a message warrant $\mathcal{M}_W'$. $\mathcal{A}_2$ can either intercept the delegation message from an original signer to a proxy signer or it can call Proxy-Signcryption oracle to sign $\mathcal{M}_W'$ and publish the delegator's signature on it.

3. The adversary $\mathcal{A}_2$ further computes ${R_1}' = u'(R_B + h_B l)$ and ${R_2}' = v'P_B$ where $h_B = \mathcal{H}_1(ID_B, R_B)$ and $l = P_{Pub}$.

4. The adversary $\mathcal{A}_2$ computes P_{Pub}' and replaces P_{Pub} with P_{Pub}'.

$$P_{Pub}' = \frac{1}{h_1 h_A + h_3 h_C}\{-h_1 R_A - h_2 P_A - h_3 R_C - h_4 P_C - K' + D_C' P\} \tag{3.8}$$

5. The adversary $\mathcal{A}_2$ chooses a proxy key D_C' and generates a proxy signcryption $(z',\ S')$ for message m' as follows:

$$z' = m' \oplus \mathcal{H}_3(ID_A||ID_C||ID_B, {R_1}', {R_2}') \tag{3.9}$$

$$S' = (u' + hD_C') \tag{3.10}$$

$$\text{where } h = \mathcal{H}(m'||ID_A||ID_C||ID_B) \tag{3.11}$$

6. The adversary $\mathcal{A}_2$ sends the signcrypted ciphertext $\left(ID_A, R_A, P_A, \mathcal{M}_W^{'}, ID_C, R_C, P_C, U^{'}, V^{'}, K^{'}, z^{'}, S^{'}\right)$ to the receiver B.

The following steps demonstrate how the fake signature S' successfully passes verification at the receiver's end and the message m' is successfully retrieved by the receiver.

1. The receiver B verifies the received message through Eq. (3.12):

$$S'P = h[K' + h_1(R_A + h_A P'_{Pub}) + h_2 P_A + h_3(R_C + h_C P'_{Pub}) + h_4 P_C] + U' \quad (3.12)$$

$$\text{where } h_A = \mathcal{H}_1(ID_A, R_A)$$

$$h_C = \mathcal{H}_1(ID_C, R_C)$$

$$h_1 = \mathcal{H}_2(\mathcal{M}'_W || ID_C, K, R_A)$$

$$h_2 = \mathcal{H}_2(\mathcal{M}'_W || ID_C, K, P_A)$$

$$h_3 = \mathcal{H}_2(\mathcal{M}'_W || ID_A, K, R_C)$$

$$h_4 = \mathcal{H}_2(\mathcal{M}'_W || ID_A, K, P_C)$$

$$h = \mathcal{H}(m' || ID_A || ID_C || ID_B)$$

2. If above verification passes, the receiver B recovers the message m' using Eq. (3.13):

$$m' = z' \oplus \mathcal{H}_3(ID_A || ID_C || ID_B, R_1', R_2') \quad (3.13)$$

$$\text{where } R_1' = S_B U', R_2' = k_B V'$$

3.5 Proposed CLPSc scheme

In this section, a novel pairing-free CLPSc scheme using ECC has been proposed which is applicable to resource-constrained low-computing mobile devices. A sender delegates the responsibility of employing health-related information (e-prescriptions) to a proxy signer (agent). The proxy signer on its part would exert the message to its final destination (pharmacy). The original signcrypter A would request for services from the CSP (receiver B). B verifies requester through the digital signatures. A delegates the signcryption rights of e-prescriptions to proxy signcrypter C on its behalf.

The proposed CLPSc scheme (shown in Figure 3.1) consists of 10 polynomial time algorithms outlined as below:

- **Setup:** KGC chooses a security parameter k and an elliptic curve E/F_p over prime finite field F_p. Let G be a cyclic subgroup of the elliptic curve group with P as a generator of order q. It chooses a master secret key $s \in Z_q^*$ and computes a master

public key $P_{Pub} = sP$. The message space is $\mathcal{M} = \{0,1\}^l$. It also chooses 4 cryptographic collision resistant hash functions as follows:

$$\mathcal{H}_1: \{0,1\}^* \times G \rightarrow Z_q^* \tag{3.14}$$
$$\mathcal{H}_2: \{0,1\}^* \times G \times G \times G \times G \rightarrow Z_q^*$$
$$\mathcal{H}_3: G \rightarrow \{0,1\}^l$$
$$\mathcal{H}_4: \{0,1\}^* \rightarrow Z_q^*$$

This algorithm returns the system parameters $\psi = \{E, F_P, G, P, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, P_{Pub}\}$ for publishing.

- **Extract-Partial-Private-Key:** KGC randomly chooses $r_A \in Z_q^*$ and computes $R_A = r_A P$ and $h_A = \mathcal{H}_1(ID_A, R_A)$. It further computes $S_A = (R_A + h_A s)$ as $A's$ partial private key and transmits S_A to an entity A through a secure channel and R_A over an insecure channel.

- **Set-Secret:** Each entity A verifies whether $S_A P = R_A + h_A P_{Pub}$. After successful verification, each entity A picks randomly $k_A \in Z_q^*$, computes $P_A = k_A P$ and sets k_A as its secret value.

- **Set-Private-Key:** Given the security parameters ψ, the partial private key S_A and the secret value k_A of the entity A as inputs, this algorithm generates a private key pair $Sk_A = (S_A, k_A)$ as an output.

- **Set-Public-Key:** This algorithm generates a public key pair $Pk_A = (R_A, P_A)$ of the entity A as an output.

- **Gen-Delegation:** This algorithm accepts the original signcrypter $A's$ secret key pair (S_A, k_A), its public key pair (R_A, P_A) and the message warrant $\mathcal{M}_W$ as inputs and generates the delegation $D_{A \rightarrow C}$ on $\mathcal{M}_W$ as an output. The original signcrypter A randomly chooses $a \in Z_q^*$ and computes $K = aP$. It then calculates t as:

$$t = \left(a + h_1(S_A + k_A)\right) \tag{3.15}$$
where $h_1 = \mathcal{H}_2(\mathcal{M}_W || ID_C, K, R_A, P_A, P_{Pub})$

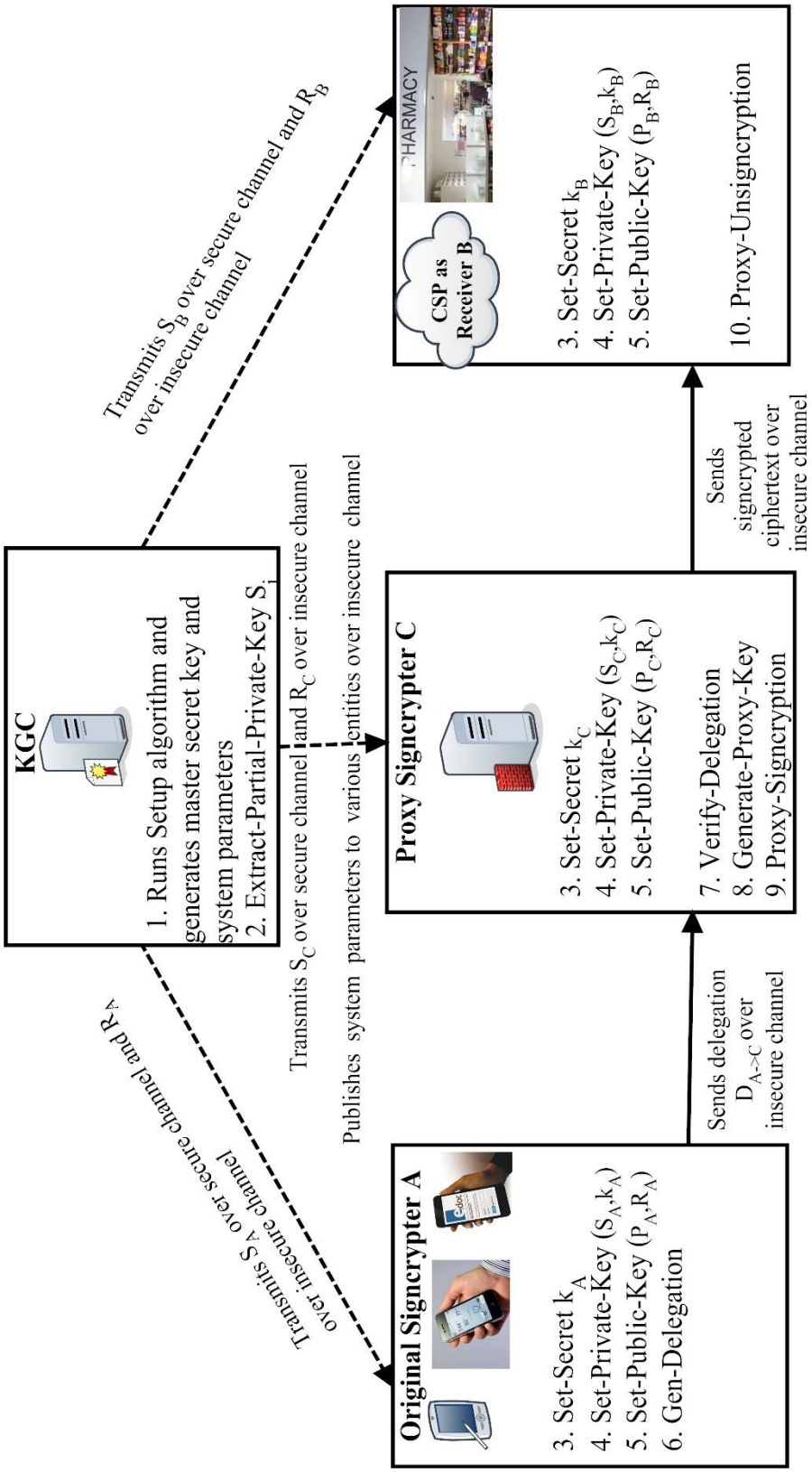

Figure 3.1: Overview of proposed CLPSc scheme for E-Prescription system in MCC

The original signcrypter A sends the delegation $D_{A\to C} = (ID_A, ID_C, \mathcal{M}_W,\ R_A,\ P_A, K, t)$ to the proxy signcrypter C.

- **Verify-Delegation:** The proxy signcrypter C verifies the delegation by computing $h_1 = \mathcal{H}_2(\mathcal{M}_W||ID_C, K, R_A, P_A, P_{Pub})$ and $h_A = \mathcal{H}_1(ID_A, R_A)$. It then checks whether $tP = h_1(R_A + h_A P_{Pub} + P_A) + K$. If not, it rejects the delegation.

- **Gen-Proxy-Key:** The proxy signcrypter C after verifying the delegation successfully, computes a proxy signing key $D_C = \big(t + h_2(S_C + k_C)\big)$ where $h_2 = \mathcal{H}_2(\mathcal{M}_W||ID_A, K, R_C, P_C, P_{Pub})$.

- **Proxy-Signcryption:** The proxy signcrypter C takes the proxy key D_C, a message $m \in \mathcal{M}$ and the public key of the receiver (R_B, P_B) as inputs and generates a signcrypted message for B as an output. The detailed process is as follows: C randomly chooses a random number $u \in Z_q^*$ and computes $U = uP,\ V = uP_B$ and $h = \mathcal{H}_3(V)$. C generates proxy signcryption of message m using Eq. (3.16) and Eq. (3.17):

$$z = m \oplus h \tag{3.16}$$

$$S = (u + h_3 D_C) \tag{3.17}$$

where $h_3 = \mathcal{H}_4(m||ID_A||ID_C||ID_B)$

C sends the signcrypted ciphertext $(ID_A, R_A, P_A, \mathcal{M}_W, ID_C, R_C, P_C, U, K, z, S)$ to the receiver B.

- **Proxy-Unsigncryption:** The receiver B receives the signcrypted ciphertext $(ID_A, R_A, P_A, \mathcal{M}_W, ID_C, R_C, P_C, U, K, z, S)$. It computes:

$$V = k_B U \tag{3.18}$$

$$m = z \oplus \mathcal{H}_3(V) \tag{3.19}$$

The receiver B accepts the message m if Eq. (3.20) holds, otherwise it returns an error message.

$$SP = h_3[K + h_1(R_A + P_A + h_A P_{Pub}) + h_2(R_C + P_C + h_C P_{Pub})] + U \tag{3.20}$$

where $h_A = \mathcal{H}_1(ID_A, R_A)$

$$h_C = \mathcal{H}_1(ID_C, R_C)$$
$$h = \mathcal{H}_3(V)$$
$$h_1 = \mathcal{H}_2(\mathcal{M}_W||ID_C, K, R_A, P_A, P_{Pub})$$
$$h_2 = \mathcal{H}_2(\mathcal{M}_W||ID_A, K, R_C, P_C, P_{Pub})$$
$$h_3 = \mathcal{H}_4(m||ID_A||ID_C||ID_B)$$

Correctness

The correctness of verification equation in the Proxy-Unsigncryption algorithm is proved as follows:

$$S = (u + h_3 D_C)$$
$$\begin{aligned} SP &= (u + h_3 D_C)P \\ &= uP + h_3 D_C P \\ &= U + h_3\big(t + h_2(S_C + k_C)\big)P \\ &= U + h_3\big(a + h_1(S_A + k_A) + h_2(S_C + k_C)\big)P \\ &= U + h_3(aP + h_1(S_A + k_A)P + h_2(S_C + k_C)P) \\ &= U + h_3(K + h_1 S_A P + h_1 k_A P + h_2 S_C P + h_2 k_C P) \\ &= U + h_3(K + h_1(r_A + h_A s)P + h_1 k_A P + h_2(r_C + h_C s)P + h_2 k_C P) \\ &= U + h_3(K + h_1(R_A + h_A P_{Pub}) + h_1 P_A + h_2(R_c + h_c P_{Pub}) + h_2 P_C) \\ &= U + h_3\big(K + h_1(R_A + P_A + h_A P_{Pub}) + h_2(R_C + P_C + h_C P_{Pub})\big) \end{aligned}$$

The correctness of message decryption in the Proxy-Unsigncryption algorithm is proved as follows:

$$\begin{aligned} V = uP_B &= u(k_B\, P) = k_B(uP\,) \\ &= \; k_B U \end{aligned}$$

Yanfeng et al's [32] scheme is vulnerable to a forgery attack under Type 2 adversary in which a fake signature gets verified at the receiver's end by modifying master public key P_{Pub} to P'_{Pub}. Since the hash values do not contain P_{Pub}, any modification in P_{Pub} is overlooked at the receiver's end and verification steps proceed normally. The proposed CLPSc scheme includes P_{Pub} in hash values h_1 and h_2 used in Gen-Delegation and Gen-

Proxy-Key algorithms respectively. Hence, an adversary is unable to launch a forgery attack even after replacing P_{Pub}, thereby making the proposed scheme resilient against Type 2 adversary.

3.6 Security Analysis

In this section, it has been shown that the proposed CLPSc scheme is secure in the random oracle model under the difficulty of solving Elliptic Curve Computational Diffie-Hellman (EC-CDH) problem. The following theorems are provided for security proof.

Theorem 1: (Proof of confidentiality) The proposed CLPSc scheme is said to be IND-CCA2 secure in the random oracle model if the underlying EC-CDH problem is hard to solve. It follows from Lemma 1 and Lemma 2.

Lemma 1: The proposed scheme is $\left(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Ppk}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \mathbb{q}_{Usc}, \upsilon\right)$-secure against indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2) in the random oracle model if there is no probabilistic polynomial time Type 1 adversary $\mathcal{A}_1$ making at most $\mathbb{q}_{\mathcal{H}_i}$ queries to random oracle $\mathcal{H}_i$, $\mathbb{q}_{Cu}$ queries to Create-User oracle, $\mathbb{q}_{Ppk}$ queries to Reveal-Partial-Private-Key oracle, $\mathbb{q}_{Sk}$ queries to Reveal-Secret-Key oracle, $\mathbb{q}_{R_Pk}$ queries to Replace-Public-Key oracle, $\mathbb{q}_{Gd}$ queries to Generate-Delegation oracle, $\mathbb{q}_{Prk}$ queries to Reveal-Proxy-Key oracle, $\mathbb{q}_{Sc}$ queries to Proxy-Signcrypt oracle and $\mathbb{q}_{Usc}$ queries to Proxy-Unsigncrypt oracle that can (t', υ')-solve the EC-CDH problem with advantage υ':

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3}}\left(2\upsilon - \mathbb{q}_{Usc}\left(\frac{1}{2}\right)^{l}\right)$$

$$t' \leq t + O(1)\left(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{Cu} + \mathbb{q}_{Ppk} + \mathbb{q}_{Sk} + \mathbb{q}_{R_{Pk}} + \mathbb{q}_{Gd} + \mathbb{q}_{Prk} + \mathbb{q}_{Sc} + \mathbb{q}_{Usc}\right) + t_{PM}\left(2\mathbb{q}_{Cu} + \mathbb{q}_{Ppk} + 2\mathbb{q}_{Sc} + 7\mathbb{q}_{Usc}\right)$$

where t_{PM} denotes time for one ECC based scalar point multiplication.

Proof: Suppose that a probabilistic polynomial time Type 1 adversary $\mathcal{A}_1$ can $(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Ppk}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \mathbb{q}_{Usc}, \upsilon)$-break the proposed scheme with advantage υ. Suppose a challenger $\mathbb{C}$ gives honest answers to the queries of the adversary $\mathcal{A}_1$. An algorithm needs to be designed that utilizes $\mathcal{A}_1$ to solve the EC-CDH problem. Given an EC-CDH instance $(P, xP, yP) \in G_q$ with unknown x and y, the problem is to compute xyP. The algorithm simulates the role of the challenger $\mathbb{C}$.

Initialization: To solve the EC-CDH problem, the challenger $\mathbb{C}$ selects a master secret key s and computes a master public key $P_{Pub} = sP$. The challenger $\mathbb{C}$ calculates system parameters $\psi = \{E, F_P, G, P, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, P_{Pub}\}$ by running Setup algorithm and outputs ψ to $\mathcal{A}_1$. Hash functions $\mathcal{H}_i$ $(1 \leq i \leq 4)$ are simulated as random oracles and are controlled by the challenger $\mathbb{C}$. The challenger $\mathbb{C}$ provides answers to the queries of $\mathcal{A}_1$ as follows:

Random oracle queries: In order to avoid inconsistency between responses to hash queries, the challenger $\mathbb{C}$ maintains 4 hash lists $\mathcal{L}_{\mathcal{H}_i}$ $(1 \leq i \leq 4)$ which are initially empty. It responds to queries from $\mathcal{A}_1$ as follows:

- $\mathcal{H}_1$ random queries: After receiving a query for (ID_A, R_A) from an adversary $\mathcal{A}_1$, $\mathbb{C}$ probes hash-list $\mathcal{L}_{\mathcal{H}_1}$ for the corresponding value. If an entry is found, $\mathbb{C}$ returns z_1 as an answer otherwise it chooses a random value $z_1 \in Z_q^*$, adds $<(ID, K), z_1>$ to $\mathcal{L}_{\mathcal{H}_1}$ and returns z_1 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_2$ random queries: After receiving a query for $(\mathcal{M}_W||ID_C, K, R_A, P_A, P_{Pub})$ from an adversary $\mathcal{A}_1$, $\mathbb{C}$ probes hash-list $\mathcal{L}_{\mathcal{H}_2}$ for the corresponding value. If an entry is found, $\mathbb{C}$ returns z_2 as an answer otherwise it chooses a random value $z_2 \in Z_q^*$, adds $<(\mathcal{M}_W||ID_C, K, R_A, P_A, P_{Pub}), z_2>$ to $\mathcal{L}_{\mathcal{H}_2}$ and returns z_2 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_3$ random queries: After receiving a query for (V) from an adversary $\mathcal{A}_1$, $\mathbb{C}$ probes hash-list $\mathcal{L}_{\mathcal{H}_3}$ for the corresponding value. If an entry is found, $\mathbb{C}$ returns z_3

as an answer otherwise it chooses a random value $z_3 \in \{0,1\}^l$, adds $<V, z_3>$ to $\mathcal{L}_{\mathcal{H}_3}$ and returns z_3 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_4$ random queries: After receiving a query for $(m||ID_A||ID_C||ID_B)$ from an adversary $\mathcal{A}_1$, $\mathbb{C}$ probes hash-list $\mathcal{L}_{\mathcal{H}_4}$ for the corresponding value. If an entry is found, $\mathbb{C}$ returns z_4 as an answer otherwise it chooses a random value $z_4 \in Z_q^*$, adds $<(m||ID_A||ID_C||ID_B), z_4>$ to $\mathcal{L}_{\mathcal{H}_4}$ and returns z_4 as an answer to $\mathcal{A}_1$ in response to its query.

Phase-I: The challenger $\mathbb{C}$ maintains the lists for issued public, private, partial, proxy keys and responds to $\mathcal{A}_1's$ queries as follows:

- Create-User query: The challenger $\mathbb{C}$ maintains a list $\mathcal{L}_I = \{ID_A, R_A, P_A, S_A, k_A\}$ of queried identities. When adversary $\mathcal{A}_1$ queries on the identity ID_A, $\mathbb{C}$ probes $\mathcal{L}_I$ to check if tuple corresponding to ID_A exists. If it exists, it returns the corresponding public key $Pk_A = (R_A, P_A)$. The challenger $\mathbb{C}$ calculates $P_P = xP$ and $P_R = yP$ for the challenging identities proxy signer and receiver respectively. Otherwise, $\mathbb{C}$ chooses 3 random numbers $k_A, r_A, z_1 \in Z_q^*$ for the entity A and computes $P_A = k_A P$, $R_A = r_A P$ and $S_A = r_A + z_1 s$. It adds tuple $< ID_A, R_A, P_A, S_A, k_A >$ to list $\mathcal{L}_I$ and $< (ID_A, R_A), z_1 >$ to $\mathcal{L}_{\mathcal{H}_1}$. It is assumed that an adversary $\mathcal{A}_1$ always calls Create-User query before calling other oracles.

- Reveal-Partial-Private-Key query: When adversary $\mathcal{A}_1$ queries on ID_A, $\mathbb{C}$ first checks if ID_A corresponds to the identity of an original signer, a proxy signer or a receiver and aborts the simulation. Otherwise, it probes list $\mathcal{L}_I$ to find the tuple corresponding to ID_A and returns its partial private key S_A to the adversary $\mathcal{A}_1$.

- Reveal-Secret-Key query: When adversary $\mathcal{A}_1$ queries on ID_A, $\mathbb{C}$ first checks if ID_A corresponds to the identity of an original signer, a proxy signer or a receiver and aborts the simulation. Otherwise, it probes list $\mathcal{L}_I$ to find the tuple corresponding to ID_A and returns its secret key k_A to the adversary $\mathcal{A}_1$.

- Replace-Public-Key query: When adversary $\mathcal{A}_1$ queries on ID_A, $\mathbb{C}$ probes $\mathcal{L}_I$ to find the tuple corresponding to ID_A and replaces the public key $Pk_A = (R_A, P_A)$ with an updated value $Pk_A{}' = (R_A{}', P_A{}')$.

- Generate-Delegation query: When adversary $\mathcal{A}_1$ queries on $(ID_A, ID_C, \mathcal{M}_W)$, $\mathbb{C}$ runs the Gen-Delegation algorithm and returns the corresponding delegation $D_{A\to C} = (ID_A, ID_C, \mathcal{M}_W, R_A, P_A, a, K, t)$ as a result to the adversary $\mathcal{A}_1$.

- Reveal-Proxy-Key query: The challenger $\mathbb{C}$ maintains a list of issued proxy keys $\mathcal{L}_{P_k} = \{ID_A, ID_C, \mathcal{M}_W, a, K, t, D_C\}$. When adversary $\mathcal{A}_1$ queries on $(ID_A, ID_C, \mathcal{M}_W)$, $\mathbb{C}$ probes $\mathcal{L}_{P_k}$. If the tuple corresponding to $(ID_A, ID_C, \mathcal{M}_W)$ exists, it returns a proxy key D_C as a result to the adversary $\mathcal{A}_1$. Otherwise, $\mathbb{C}$ probes $\mathcal{L}_{\mathcal{H}_2}$ for $(\mathcal{M}_W||ID_C, K, R_C, P_C, P_{Pub})$ which returns z_2 as a solution. It queries the aforementioned Generate-Delegation query on $(ID_A, ID_C, \mathcal{M}_W)$ to obtain $D_{A\to C} = (ID_A, ID_C, \mathcal{M}_W, R_A, P_A, a, K, t)$ as a result. It probes list $\mathcal{L}_I$ for ID_C to obtain the corresponding secret key (S_C, k_C) and computes $D_C = t + z_2(S_C + k_C)$. It adds tuple $< ID_A, ID_C, \mathcal{M}_W, a, K, t, D_C >$ to list $\mathcal{L}_{p_k}$ and returns D_C as a result.

- Proxy-Signcrypt query: When the adversary $\mathcal{A}_1$ queries for signcrypting a message m with 3 input identities (ID_A, ID_B, ID_C) and the message warrant $\mathcal{M}_W$, $\mathbb{C}$ probes list $\mathcal{L}_{p_k}$ for an entry $(ID_A, ID_C, \mathcal{M}_W)$. If it exists, $\mathbb{C}$ utilizes the corresponding proxy key D_C and K to create a signcrypted message δ. Otherwise, it issues the aforementioned Reveal-Proxy-Key query which returns the proxy key D_C as a result. In order to signcrypt a plaintext message m on behalf of original signer, $\mathbb{C}$ first checks if ID_A and ID_B are challenging identities or not. If not, $\mathbb{C}$ answers the query by running Proxy-Signcryption algorithm since it knows the secret key (S_A, k_A) corresponding to ID_A. The challenger adds a tuple $< ID_A, Pk_A, \mathcal{M}_W, ID_B, Pk_B, \mathcal{M}, \delta >$ to $\mathcal{L}_{Sc}$. Otherwise, $\mathbb{C}$ chooses $x, z_4 \in Z_q^*$ randomly to compute $U = xP, V = xyP$. $\mathbb{C}$ probes $\mathcal{L}_{\mathcal{H}_3}$ for (V) which returns z_3 as a solution and then computes $z = m \oplus z_3$ and $S = x + z_4 D_C$. It adds a tuple $< (m||ID_A||ID_C||ID_B), z_4 >$ to list $\mathcal{L}_{\mathcal{H}_4}$ such that $\mathcal{H}_4(m||ID_A||ID_C||ID_B) = z_4$.

- Proxy-Unsigncrypt query: When the adversary $\mathcal{A}_1$ submits the signcrypted message $\delta = (ID_A, R_A, P_A, \mathcal{M}_W, ID_C, R_C, P_C, U, K, z, S)$ for the receiver B, $\mathbb{C}$ checks if ID_B is a challenging identity or not. If it is, $\mathbb{C}$ checks $\mathcal{L}_{Sc}$ for an entry $< ID_A, Pk_A, \mathcal{M}_W, ID_B, Pk_B, *, \delta >$ and returns the corresponding plaintext message m. Else, $\mathbb{C}$ calculates $V = k_B U$ since it knows the secret key of the receiver. $\mathbb{C}$ searches the hash-list $\mathcal{L}_{\mathcal{H}_3}$ with V as key value which returns z_3 as a solution and calculates $m = z \oplus z_3$. $\mathbb{C}$ also searches the hash-list $\mathcal{L}_{\mathcal{H}_1}$ for (ID_A, R_A) and (ID_C, R_C), $\mathcal{L}_{\mathcal{H}_2}$ for $(\mathcal{M}_W || ID_C, K, R_A, P_A, P_{Pub})$ and $(\mathcal{M}_W || ID_C, K, R_C, P_C, P_{Pub})$ and $\mathcal{L}_{\mathcal{H}_4}$ for $(m || ID_A || ID_C || ID_B)$ to obtain z_1, z_1', z_2, z_2' and z_4 as solutions respectively. If equation $SP = z_4[K + z_2(R_A + P_A + z_1 P_{Pub}) + z_2'(R_C + P_C + z_1' P_{Pub})] + U$ is satisfied, then m is returned. Otherwise, $\mathbb{C}$ returns an error message.

Challenge: The adversary $\mathcal{A}_1$ chooses 3 distinct identities (ID_O^*, ID_P^*, ID_R^*) and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$. The challenger flips a coin ß $\in \{0,1\}$ and generates a signcrypted message δ^* corresponding to $\mathcal{M}_ß$. The challenger $\mathbb{C}$ probes $\mathcal{L}_{p_k}$ for an entry $(ID_O^*, ID_P^*, \mathcal{M}_W{}^*)$ to obtain $(ID_O^*, ID_P^*, \mathcal{M}_W{}^*, a, K^*, t^*, D_C{}^*)$. $\mathbb{C}$ chooses $x, z_4{}^* \in Z_q^*$ randomly to compute $U^* = xP$, $V^* = xP_R$. $\mathbb{C}$ probes $\mathcal{L}_{\mathcal{H}_3}$ for (V^*) which returns $z_3{}^*$ as a solution and computes $z^* = \mathcal{M}_ß \oplus z_3{}^*$ and $S^* = x + z_4{}^* D_C{}^*$. It adds a tuple $< (\mathcal{M}_ß || ID_O^* || ID_P^* || ID_R^*), z_4{}^* >$ to list $\mathcal{L}_{\mathcal{H}_4}$ such that $\mathcal{H}_4(\mathcal{M}_ß || ID_O^* || ID_P^* || ID_R^*) = z_4{}^*$. The challenger $\mathbb{C}$ returns a signcrypted message $\delta^* = (ID_O{}^*, R_O{}^*, P_O{}^*, \mathcal{M}_W{}^*, ID_P{}^*, R_P{}^*, P_P{}^*, U^*, K^*, z^*, S^*)$ for $\mathcal{M}_ß$ to the adversary $\mathcal{A}_1$.

Phase-II: After obtaining δ^*, the adversary $\mathcal{A}_1$ continues to probe the challenger with queries similar to those in Phase-I except for Proxy-Unsigncrypt query on δ^* to find a message $\mathcal{M}_0$ or $\mathcal{M}_1$ trivially, Reveal-Partial-Private-Key and Reveal-Secret-Key oracle for the receiver at any point of the game.

Output: Finally $\mathcal{A}_1$ outputs a bit ß′ $\in \{0,1\}$ and wins the game if ß′ = ß.

Analysis: Here, it will be proved that if an adversary can break the proposed scheme, then there exists an algorithm, simulated by the challenger which can solve the EC-CDH problem. The challenger $\mathbb{C}$ always outputs a valid proxy delegation for $\mathcal{A}_1$'s $\mathbb{q}_{\boldsymbol{Gd}}$ and $\mathbb{q}_{\boldsymbol{Prk}}$

queries. For each $\mathbb{q}_{Sc}$, $\mathbb{C}$ computes an indistinguishable signcrypted message by manipulating $\mathcal{H}_3$ random oracle. Therefore, simulations of $\mathbb{q}_{Gd}$, $\mathbb{q}_{Prk}$ and $\mathbb{q}_{Sc}$ queries are perfect. Next, Proxy-Unsigncrypt query will be evaluated. Let E_1 be an event such that Proxy-Unsigncrypt query returns an error for a valid signcrypted message $\delta^* = (ID_O{}^*, R_O{}^*, P_O{}^*, \mathcal{M}_W{}^*, ID_P{}^*, R_P{}^*, P_P{}^*, U^*, K^*, z^*, S^*)$. Let E_2 be an event such that a signcrypted message submitted by the adversary $\mathcal{A}_1$ is valid. Let $Ask\mathcal{H}_3{}^*$ be the event that $\mathcal{H}_3$ random oracle has been queried for V^* by an adversary in Phase-II. The error probability of $\mathbb{q}_{Usc}$ is given by Eq. (3.21):

$$P[E_2|\neg Ask\mathcal{H}_3{}^*] \leq \left(\frac{1}{2}\right)^l \tag{3.21}$$

Since an adversary is allowed to make at most $\mathbb{q}_{Usc}$ Proxy-Unsigncrypt queries, the probability of E_1 is given by Eq. (3.22)

$$P[E_1] \leq \mathbb{q}_{Usc}\left(\frac{1}{2}\right)^l \tag{3.22}$$

Let E_3 be an event such that the entire simulation game is perfect. This is possible if the adversary $\mathcal{A}_1$ does not query $\mathcal{H}_3$ random oracle for V^* in Phase-II. The adversary $\mathcal{A}_1$ gains no advantage in guessing ß due to random outputs obtained from the random oracles.

$$P[ß' = ß|E_3] = \frac{1}{2} \tag{3.23}$$

By the theorem of total probability, we have

$$P[ß' = ß] = P[ß' = ß|E_3]P[E_3] + P[ß' = ß|\neg E_3]P[\neg E_3]$$

$$P[ß' = ß] \leq \frac{1}{2}P[E_3] + P[\neg E_3]$$

$$P[ß' = ß] \leq \frac{1}{2}(1 - P[\neg E_3]) + P[\neg E_3] \tag{3.24}$$

Similarly,

$$P[ß' = ß] \geq P[ß' = ß|E_3]P[E_3]$$

$$P[ß' = ß] \geq \frac{1}{2}(1 - P[\neg E_3])$$

$$P[ß' = ß] \geq \frac{1}{2} - \left(\frac{1}{2} P[\neg E_3]\right) \tag{3.25}$$

From Eq. (3.24) and (3.25) we obtain Eq. (3.26),

$$\left|P[ß' = ß] - \frac{1}{2}\right| \leq \left(\frac{1}{2} P[\neg E_3]\right) \tag{3.26}$$

Since $\mathcal{A}_1$ has non-negligible advantage υ to break the proposed scheme, we therefore have

$$\upsilon = \left|P[ß' = ß] - \frac{1}{2}\right|$$

$$\upsilon \leq \left(\frac{1}{2} P[\neg E_3]\right)$$

$$\upsilon \leq \frac{1}{2}(P[E_1] + P[Ask\mathcal{H}_3{}^*])$$

$$P[Ask\mathcal{H}_3{}^*] \leq 2\upsilon - \mathbb{q}_{\boldsymbol{Usc}} \left(\frac{1}{2}\right)^l$$

If $Ask\mathcal{H}_3{}^*$ happens, then the challenger $\mathbb{C}$ solves the EC-CDH problem with probability $\frac{1}{\mathbb{q}_{\mathcal{H}_3}}$ and advantage υ' is given by Eq. (3.27):

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3}} P[Ask\mathcal{H}_3{}^*]$$

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3}} \left(2\upsilon - \mathbb{q}_{\boldsymbol{Usc}} \left(\frac{1}{2}\right)^l\right) \tag{3.27}$$

Lemma 2: The proposed scheme is $\left(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \mathbb{q}_{Usc}, \upsilon\right)$-secure against indistinguishability under adaptive chosen-ciphertext attack (IND-CCA2) in the random oracle model if there is no probabilistic polynomial time Type 2 adversary $\mathcal{A}_2$ making at most $\mathbb{q}_{\mathcal{H}_i}$ queries to random oracle $\mathcal{H}_i$, $\mathbb{q}_{Cu}$ queries to Create-User oracle, $\mathbb{q}_{Sk}$ queries to Reveal-Secret-Key oracle, $\mathbb{q}_{Gd}$ queries to Generate-Delegation oracle, $\mathbb{q}_{Prk}$ queries to Reveal-Proxy-Key oracle, $\mathbb{q}_{Sc}$ queries to Proxy-Signcrypt oracle and $\mathbb{q}_{Usc}$ queries to Proxy-Unsigncrypt oracle that can (t', υ')-solve the EC-CDH problem with advantage υ':

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3}} \left(2\upsilon - \mathbb{q}_{\boldsymbol{Usc}} \left(\frac{1}{2}\right)^l\right)$$

$$t' \leq t + O(1)\left(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{Cu} + \mathbb{q}_{Sk} + \mathbb{q}_{Gd} + \mathbb{q}_{Prk} + \mathbb{q}_{Sc} + \mathbb{q}_{Usc}\right) + t_{PM}(2\mathbb{q}_{Cu} + 2\mathbb{q}_{Sc} + 7\mathbb{q}_{Usc})$$

where t_{PM} denotes time for one ECC based scalar point multiplication.

Proof: The proof is similar to Lemma 1.

Theorem 2: (Proof of Unforgeability) The proposed CLPSc scheme is said to be EUF-CMA secure in the random oracle model if the underlying EC-CDH problem is hard to solve. It follows from Lemma 3 and Lemma 4.

Lemma 3: The proposed scheme is $(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Ppk}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \upsilon)$-secure against existential forgery under adaptive chosen-message attack (EUF-CMA) in the random oracle model if there is no probabilistic polynomial time Type 1 adversary $\mathcal{A}_1$ making at most $\mathbb{q}_{\mathcal{H}_i}$ queries to random oracle $\mathcal{H}_i$, $\mathbb{q}_{Cu}$ queries to Create-User oracle, $\mathbb{q}_{Ppk}$ queries to Reveal-Partial-Private-Key oracle, $\mathbb{q}_{Sk}$ queries to Reveal-Secret-Key oracle, $\mathbb{q}_{R_Pk}$ queries to Replace-Public-Key oracle, $\mathbb{q}_{Gd}$ queries to Generate-Delegation oracle, $\mathbb{q}_{Prk}$ queries to Reveal-Proxy-Key oracle and $\mathbb{q}_{Sc}$ queries to Proxy-Signcrypt oracle that can (t', υ')-solve the EC-CDH problem with advantage υ':

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3} * \mathbb{q}_{\mathcal{H}_4}} \left(\upsilon - \left(1 + \mathbb{q}_{\mathcal{H}_4}\right) \left(\frac{1}{2}\right)^{l} \right)$$

$$t' \leq t + O(1)\left(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{Cu} + \mathbb{q}_{Ppk} + \mathbb{q}_{Sk} + \mathbb{q}_{Gd} + \mathbb{q}_{Prk} + \mathbb{q}_{Sc}\right) + t_{PM}\left(2\mathbb{q}_{Cu} + \mathbb{q}_{Ppk} + 2\mathbb{q}_{Sc}\right)$$

where t_{PM} denotes time for one ECC based scalar point multiplication.

Proof: Suppose that a probabilistic polynomial time Type 1 adversary $\mathcal{A}_1$ can $(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Ppk}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \upsilon)$-break the proposed scheme with advantage υ. Suppose a challenger $\mathbb{C}$ gives honest answers to the queries of the adversary $\mathcal{A}_1$. An algorithm should be developed that utilizes $\mathcal{A}_1$ to solve the EC-CDH

problem. Given an EC-CDH instance $(P, xP, yP) \in G_q$ with unknown x and y, the problem is to compute xyP. The algorithm simulates the role of the challenger $\mathbb{C}$.

Initialization: To solve the EC-CDH problem, the challenger $\mathbb{C}$ selects a master secret key s and computes a master public key $P_{Pub} = sP$. The challenger $\mathbb{C}$ calculates the system parameters $\psi = \{E, F_P, G, P, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, P_{Pub}\}$ by running Setup algorithm and outputs ψ to $\mathcal{A}_1$. Hash functions $\mathcal{H}_i$ $(1 \leq i \leq 4)$ are simulated as random oracles and controlled by the challenger $\mathbb{C}$.

Phase-I: As defined in Phase-I of the proof of Theorem 1, the adversary $\mathcal{A}_1$ is allowed to make $\mathbb{q}_{Cu}, \mathbb{q}_{Ppk}, \mathbb{q}_{Sk}, \mathbb{q}_{R_Pk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}$ queries and $\mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}$ queries to the random oracles.

Forgery: Finally, the adversary $\mathcal{A}_1$ outputs a new signcryption $\delta^* = (ID_A{}^*, R_A{}^*, P_A{}^*, \mathcal{M}_W{}^*, ID_C{}^*, R_C{}^*, P_C{}^*, U^*, K^*, z^*, S^*)$ on message m for ID_A^* as the original signer and ID_B^* as the receiver under warrant $\mathcal{M}_W{}^*$, which is not composed by Proxy-Signcrypt oracle, unlike the normal case. ID_C^* is not designated as a proxy signer by ID_A^*. The adversary $\mathcal{A}_1$ succeeds in the game if the outcome of the Proxy-Unsigncrypt oracle is not an error message and $\mathcal{A}_1$ has not queried both Reveal-Partial-Private-Key oracle and Replace-Public-Key oracle or Reveal-Secret-Key oracle of the original signer at any instant of the game.

Analysis: Here, it has to be proved that if an adversary can break the proposed scheme, then there exists an algorithm, simulated by the challenger which can solve the EC-CDH problem. According to the analysis of Theorem 1, simulations of $\mathbb{q}_{Gd}$ and $\mathbb{q}_{Prk}$ queries are perfect. Since the challenger answers all the queries of random oracles with an indistinguishable value, its simulations are also perfect until $\mathcal{A}_1$ queries $\mathcal{H}_3$ and $\mathcal{H}_4$. Let E be an event such that forged signcrypted message $\delta^* = (ID_A{}^*, R_A{}^*, P_A{}^*, \mathcal{M}_W{}^*, ID_C{}^*, R_C{}^*, P_C{}^*, U^*, K^*, z^*, S^*)$ is valid. Let $Ask\mathcal{H}_3{}^*$ and $Ask\mathcal{H}_4{}^*$ be the event such that $\mathcal{H}_3$ and $\mathcal{H}_4$ random oracles have been queried by an adversary previously. The probability that $\mathcal{A}_1$ can guess the correct values without querying $\mathcal{H}_3$ and

$\mathcal{H}_4$ random oracles is not greater than $\left(\frac{1}{2}\right)^l$. The non-negligible advantage υ of the adversary $\mathcal{A}_1$ to break the proposed scheme can be calculated as:

$$\upsilon = P[E] = P[E|Ask\mathcal{H}_4{}^*] + P[E|\neg\, Ask\mathcal{H}_4{}^*]$$

$$\upsilon \leq P[E|Ask\mathcal{H}_4{}^*] + \left(\frac{1}{2}\right)^l$$

$$\upsilon \leq P[E|Ask\mathcal{H}_4{}^* \cap Ask\mathcal{H}_3{}^*] + P[E|Ask\mathcal{H}_4{}^* \cap \neg\, Ask\mathcal{H}_3{}^*] + \left(\frac{1}{2}\right)^l$$

$$\upsilon \leq P[E|Ask\mathcal{H}_4{}^* \cap Ask\mathcal{H}_3{}^*] + \mathbb{q}_{\mathcal{H}_4}\left(\frac{1}{2}\right)^l + \left(\frac{1}{2}\right)^l$$

$$P[E|Ask\mathcal{H}_4{}^* \cap Ask\mathcal{H}_3{}^*] \geq \upsilon - \left(1 + \mathbb{q}_{\mathcal{H}_4}\right)\left(\frac{1}{2}\right)^l$$

When an event E occurs under the condition such that both $\mathcal{H}_3$ and $\mathcal{H}_4$ random oracles have been queried, the probability that the challenger solves the EC-CDH problem is $\frac{1}{\mathbb{q}_{\mathcal{H}_3{}^*}\mathbb{q}_{\mathcal{H}_4}}$ and an advantage υ' is given by Eq. (3.28)

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3{}^*}\mathbb{q}_{\mathcal{H}_4}} P[E|Ask\mathcal{H}_4{}^* \cap Ask\mathcal{H}_3{}^*]$$

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3{}^*}\mathbb{q}_{\mathcal{H}_4}}\left(\upsilon - \left(1 + \mathbb{q}_{\mathcal{H}_4}\right)\left(\frac{1}{2}\right)^l\right) \tag{3.28}$$

Lemma 4: The proposed scheme is $(t, \mathbb{q}_{\mathcal{H}_1}, \mathbb{q}_{\mathcal{H}_2}, \mathbb{q}_{\mathcal{H}_3}, \mathbb{q}_{\mathcal{H}_4}, \mathbb{q}_{Cu}, \mathbb{q}_{Sk}, \mathbb{q}_{Gd}, \mathbb{q}_{Prk}, \mathbb{q}_{Sc}, \upsilon)$-secure gainst existential forgery under adaptive chosen-message attack (EUF-CMA) in the random oracle model if there is no probabilistic polynomial time Type 2 adversary $\mathcal{A}_2$ making at most $\mathbb{q}_{\mathcal{H}_i}$ queries to random oracle $\mathcal{H}_i$, $\mathbb{q}_{Cu}$ queries to Create-User oracle, $\mathbb{q}_{Sk}$ queries to Reveal-Secret-Key oracle, $\mathbb{q}_{Gd}$ queries to Generate-Delegation oracle, $\mathbb{q}_{Prk}$ queries to Reveal-Proxy-Key oracle, $\mathbb{q}_{Sc}$ queries to Proxy-Signcrypt oracle that can (t', υ')-solve the EC-CDH problem with advantage υ':

$$\upsilon' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_3{}^*}\mathbb{q}_{\mathcal{H}_4}}\left(\upsilon - \left(1 + \mathbb{q}_{\mathcal{H}_4}\right)\left(\frac{1}{2}\right)^l\right)$$

$$t' \leq t + O(1)\left(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{Cu} + \mathbb{q}_{Sk} + \mathbb{q}_{Gd} + \mathbb{q}_{Prk} + \mathbb{q}_{Sc} + \mathbb{q}_{Usc}\right) + t_{PM}(2\mathbb{q}_{Cu} + 2\mathbb{q}_{Sc})$$

where t_{PM} denotes time for one ECC based scalar point multiplication.

Proof: The proof is similar to Lemma 3.

3.7 Computational and Performance Analysis

This section deals with the comparison of the proposed CLPSc scheme with various other proxy signcryption schemes in the literature [134-139, 32]. Table 3.1 provides a detailed comparison of schemes in terms of computational costs where t_{bp} denotes the time taken for a bilinear pairing operation, t_e denotes exponentiation operation time, t_{enc} denotes symmetric encryption time, t_{dec} denotes symmetric decryption time and t_{pm} denotes an ECC-based point multiplication time. Since point addition and hash functions have very little cost, they are ignored while calculating the overall running time. The running time of proxy credential generation, proxy-signcryption and proxy-unsigncryption phases is computed by using the standard cryptographic library MIRACL [34]. The hardware platform consists of a PIV 3GHz processor and a 512 MB RAM and the method followed in [127] is used to evaluate the total computation time. A Koblitz curve sect163k1 over $F_{2^{163}}$ is selected from the list of elliptic curves mentioned by NIST [141]. The curve $E: y^2 + xy = (x^3 + ax^2 + b)$ over 163 bit random prime is defined by $a = 1, b = 1$.

According to Table 4.1 and Figure 4.2 , the total computation time of the the proposed scheme is 13.63% of Elkamchouchi et al's scheme [134], 15.68% of Wang et al.'s scheme [135], 17.60% of Lin et al.'s scheme [136], 17.86% of Lo et al.'s scheme [137], 60% of Yanfeng et al.'s scheme [32], 13.95% of Ming et al.'s scheme [138] and 7.87% of Zhou et al.'s scheme [139].

Table 3.1: Comparative analysis of different schemes

Schemes → Parameters ↓	Elkamchouchi et al. [134]	Wang et al. [135]	Lin et al. [136]	Lo et al. [137]	Yanfeng et al. [32]	Ming et al. [138]	Zhou et al. [139]	Proposed Scheme
Proxy key generation	$2t_{pm}$	0	t_{pm}	t_{pm}	0	$3t_e$	$2t_e$	0
Proxy-Signcryption	$2t_e + 2t_{bp} + t_{pm}$	$t_{bp} + 2t_{pm}$	$t_e + t_{bp} + t_{enc} + 4t_{pm}$	$t_e + t_{bp} + t_{enc} + 3t_{pm}$	$5t_{pm}$	$5t_e + t_{bp}$	$4t_e + t_{bp}$	$2t_{pm}$
Proxy-Unsigncryption	$2t_e + 4t_{bp}$	$5t_{bp} + t_{pm}$	$3t_{bp} + t_{dec} + 4t_{pm}$	$3t_{bp} + t_{dec} + 4t_{pm}$	$10t_{pm}$	$4t_{bp}$	$7t_e + 8t_{bp}$	$7t_{pm}$
Total operations	$4t_e + 6t_{bp} + 2t_{pm}$	$6t_{bp} + 3t_{pm}$	$t_e + 4t_{bp} + t_{enc} + t_{dec} + 9t_{pm}$	$t_e + 4t_{bp} + t_{enc} + t_{dec} + 8t_{pm}$	$15t_{pm}$	$8t_e + 5t_{bp}$	$13t_e + 9t_{bp}$	$9t_{pm}$
Running time (ms)	145.9	126.87	113.06	111.39	33.15	142.68	252.63	19.89
Security proof on confidentiality	No	No	Yes	Yes	No	Yes	Yes	Yes
Unforgeability proof on proxy signcryption	No	No	Yes	Yes	No	Yes	Yes	Yes
Resistance against forgery attack	Yes	Yes	No	Yes	No	Yes	Yes	Yes

It is evident from Table 3.1 and Figure 3.2 that the proposed CLPSc scheme is more efficient than the existing proxy signcryption schemes [134-139, 32] in terms of the computational cost.

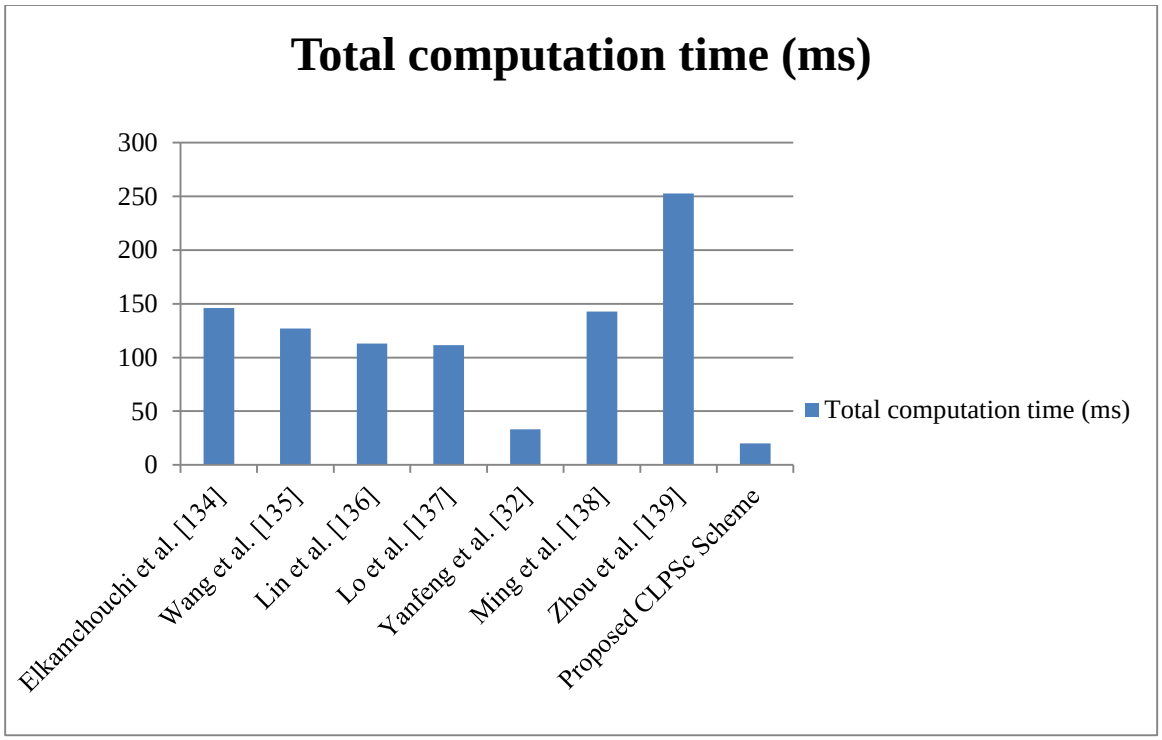

Figure 3.2: Comparison of existing schemes with proposed CLPSc scheme in terms of computation time

3.8 Summary

To the best of our knowledge, the scheme proposed by Yanfeng et al. is the only secure pairing-free CLPSc scheme that exists in the literature. However, this scheme is susceptible to a forgery attack. An exhaustive cryptanalysis proves that Yanfeng et al's scheme is forgeable under Type 2 adversary. A novel pairing-free CLPSc scheme using ECC for e-prescription system in MCC paradigm has been proposed. This scheme was analyzed and found to be more efficient than existing schemes. So, it can be employed with mobile devices operating in resource-constrained environments with intermittent bandwidth, connectivity, battery power, etc. The formal security analysis proved that the proposed CLPSc scheme is resilient against forgery attack by Type 1 and Type 2 adversaries.

Chapter 4

Certificateless Proxy Re-encryption

4.1 Introduction

A major cause for security nightmares for healthcare organizations is the constant threat associated with the MCDs of them being misplaced, stolen or breached due to their increased portability [142]. Serious security and privacy concerns must be brought in the foray before PHRs are transferred onto the cloud. The m-health devices with their heaps of accomplishments pose complex security issues with their HIPAA non-compliance. Worldwide doctors are considering data clouds as an alternative to storing medical records electronically. However, the storage scheme comes with a readymade baggage of security issues engulfing really sensitive information [143]. The major security and privacy challenges posed here are privacy of the patient's data, access control management, authenticity and integrity of data. A technology strategy securing data, taking into account its widespread accessibility with MCDs, is an area that could provide a fruitful solution to this dilemma. Encryption would be hence an indispensable part of the solution.

A conventional way of incorporating data confidentiality involves a significant effort by the data owner in encrypting the data with key (symmetric or public key) before outsourcing to the cloud [144]. The traditional asymmetric cryptography is feasible only under the circumstance of the data possessor requiring an access to the uploaded data as it can be decrypted by his/her private key. This approach hampers flexible sharing of data with multiple users as it necessitates sharing of keys by the genuine data possessor for

---

The major findings of this chapter have been peer reviewed and published in:

- *Tarunpreet Bhatia, A.K. Verma, Gaurav Sharma, "Secure Sharing of Mobile Personal Healthcare Records using Certificateless Proxy Re-Encryption in Cloud", Transactions on Emerging Telecommunications Technologies, doi: 10.1002/ett.3309.*

decryption with authorized target users, henceforth violating least privilege principle [145]. Another alternative is to download the data, re-encrypt with intended recipient's public key and re-upload on cloud by the data owner. This strategy incurs heavy processing overhead, extensive computation and communication cost to the data owner, defeating the very purpose of cloud computing. Broadcast encryption and group key management provide an alternative for sharing data within a group, but managing a group in today's cloud-based networks, where group size is large and frequent membership changes, is very tedious.

Proxy re-encryption (PRE) delivers a propitious solution to the aforementioned problems. In PRE [92], data owner allows the delegation of decryption rights to authorized users by sending re-encryption key to cloud-resident proxy server which further transforms the ciphertext under the public key of data owner into ciphertext applicable under intended recipient's public key without peeping into the message contents. The authorized users can decrypt the message. Proxy re-encryption makes itself useful in various fields such as encrypted email forwarding [146], cloud storage [147], digital rights management [148], distributed file systems [149] and public transportation privacy [150]. PRE is classified as unidirectional in which the proxy can metamorphose ciphertext from user A to user B and bidirectional, which works in both directions. PRE can also be categorized into a single-hop and a multi-hop based on the number of transformations. In the single-hop, transformation occurs one time from user A to user B while in the multi-hop, it can occur multiple times from user A to B to C and so on.

4.2 Related Work

The notion of PRE was presented by Blaze et al. [105] in 1998. However, it was bidirectional and colluding unsafe. Following Blaze et al's PRE scheme, Ateniese et al. put forward the unidirectional PRE scheme based on paillier encryption. Later, they proposed two more IND-CPA (Indistinguishability under Chosen-Plaintext Attack) secure schemes [149] based on bulletin board system with pairing and two level encryption schemes, respectively. Canneti and Hohenberger [151] advanced first IND-CCA (Indistinguishability under adaptive Chosen Ciphertext Attack) secure bidirectional multi-hop PRE scheme by applying Canetti, Halevi, and Katz transformation. Libert and

Vergnaud extended [152] to make it IND-CCA2 secure and re-encrypted ciphertext is publicly verifiable. Deng et al. [153] proposed first IND-CCA2 secure pairing-free bidirectional PRE scheme by integrating El-Gamal encryption and Schnorr's signature, making it efficient than previous paradigms and leaving the scope for the construction of IND-CCA2 secure PRE schemes in standard model which was solved by Wang et al. in 2015 [154] using Cramer-Shoup encryption and compared their efficiency with the work of Canneti and Hohenberger [151].

In order to combat against certification management problem in PRE, Green and Ateniese [106] employed conventional PRE in Identity-based cryptographic primitive for the first time and proposed two identity-based PRE (IB-PRE) schemes, one multi-hop CCA secure and other single-hop CCA secure. Following this along the years, many unidirectional IB-PRE schemes have been proposed [155-162]. However, schemes [155, 156] are vulnerable to collusion attack in which proxy in collusion with delegatee can extract the private key of the delegator. Later, Wang et al. [157] constructed IB-PRE scheme in the random oracle model, and Mizuno and Doi [158] designed IB-PRE algorithm based on CPA security in the standard model. Shao and Cao [159] proposed CCA-secure IB-PRE scheme in the standard model. Liang et al. [160] put forth the first CCA-secure single hop IBE-PRE in the standard model upholding conditional re-encryption. In 2014, Liang et al. [161] proposed cloud-based revocable IB-PRE scheme in which ciphertext is re-encrypted by proxy under an identity and time period. However, Wang et al. [162] proved that Liang's scheme [160] is prone to collusion and re-encryption key forgery attack which allowed the revoked users to decipher ciphertext even after a timer expires. They further proposed an improvement based on expensive pairing operations under the standard model.

IBE exposes another loophole in the form of key escrow problem. This is where CL-PRE rises up to the occasion. Sur et al. [163] have presented CL-PRE primitive with pairing for the first time, and since then, this primitive has attracted more attention from academia and research community. They claimed their scheme to be CCA-secure, but, while its cryptanalysis, Zheng et al. [164] performed concrete attack on their scheme. In 2012, Xu et al. [92] designed CL-PRE scheme for data apportioning with public cloud. It leveraged cloud for encryption-based access control and key management and claimed its security

against CPA. They further proposed multi-proxy and randomized CL-PRE to augment the security and efficiency. In 2013, Guo et al. [165] proved that Xu et al.'s scheme [92] is vulnerable against Type 1 adversary and proposed replayable CCA (RCCA) secure PRE scheme in the random oracle model. They compared the performance with IB-PRE scheme in the work of Green et al. [106]. All the aforementioned schemes [92, 163-165] rely on expensive bilinear pairing operations. To the extent of our knowledge, only four pairing-free CL-PRE schemes are in existence in the literature. Yang et al. [166] proposed first pairing-free CL-PRE scheme in 2014. They proved that Xu et al.'s [92] CL-PRE scheme lacks confidentiality and compared their schemes with the works of Xu et al. [92] and Sur et al. [163] in terms of computation time. In 2014, Wang et al. [167] proposed CCA-secure bidirectional CL-PRE scheme under Yang et al.'s security models [166]. However, proxy needs secret keys of both sender and receiver for re-encryption. In 2015, Qin et al. [33] proposed CL-PRE scheme for data apportioning in cloud and claim its security against IND-CCA under strong security model in which Type 1 adversary is permitted to substitute public key of challenge identity before challenge phase. However, they did not provide any formal security analysis. The outcomes from simulation proved their scheme outperformed Xu et al.'s scheme [92], Sur et al's scheme [163] and Yang et al.'s scheme [166] in terms of communication and storage overhead. Srinivasan and Rangan [168] broke the confidentiality of Yang et al's scheme [166], making it CCA-insecure and proposed CCA-secure unidirectional and single-hop CL-PRE scheme. They compared their scheme with Guo et al's scheme [165] in terms of efficiency. The scheme in the work of Srinivasan and Rangan [168] required several pre-computations during the key generation which are stored locally, thereby increasing memory overhead and making it unsuitable for mobile devices. Selvi et al. [169] demonstrated a major flaw in the CCA-security proof of Srinivasan's et al. scheme [168] in which adversary can easily distinguish between the simulation of the challenger and the real system.

4.3 Review of Qin's et al scheme

Qin's CL-PRE scheme consists of following 9 algorithms as below.

- **Setup:** KGC chooses a security parameter k and generates a prime number q and group G of order q. It picks a random generator $g \in G$ and $y \in Z_q^*$ to compute $x = g^y$. KGC also chooses master secret key $s \in Z_q^*$. It also chooses 5 cryptographic collision-free hash functions as follows:

$$\mathcal{H}_1: \{0,1\}^* \times G \rightarrow Z_q^* \quad (4.1)$$
$$\mathcal{H}_2: \{0,1\}^* \times G \times G \rightarrow Z_q^*$$
$$\mathcal{H}_3: \{0,1\}^* \rightarrow Z_q^*$$
$$\mathcal{H}_4: G \rightarrow \{0,1\}^{m+n}$$
$$\mathcal{H}_5: G \rightarrow Z_q^*$$

  where m is bit length of plaintext message and n is random bit string. The plaintext message space is $\mathcal{M} = \{0,1\}^m$ and ciphertext space is $\mathcal{C} = \{0,1\}^{m+n}$. This algorithm returns public parameters $\Psi = \{G, g, q, x, m, n, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5\}$ for publishing.

- **Extract partial key:** It accepts public parameters Ψ, identity ID_A, master secret key s and picks random numbers $i_A, j_A \epsilon Z_q^*$ and computes S_A using Eq. (4.3) and P_A using Eq. (4.5):

$$\alpha_1 = g^{i_A}, \alpha_2 = g^{j_A} \quad (4.2)$$
$$S_A = x_A^1 = i_A + s\mathcal{H}_1(ID_A, \alpha_1) \quad (4.3)$$
$$x_A^2 = j_A + s\mathcal{H}_2(ID_A, \alpha_1, \alpha_2) \quad (4.4)$$
$$P_A = (\alpha_1, \alpha_2, x_A^2) \quad (4.5)$$

- **Set secret value:** Each identity A randomly generates $k_A \in Z_q^*$ as a secret value for ID_A.

- **Generate private key:** This algorithm accepts public parameters Ψ, partial private key S_A and secret key k_A as inputs and generates full private key $Sk_A = (S_A, k_A) = (x_A^1, k_A)$ for an identity ID_A.

- **Generate public key:** This algorithm accepts public parameters Ψ, partial public key P_A, user's secret key k_A as inputs and computes $p_A = g^{k_A}$ and generates a full public key $Pk_A = (\alpha_1, \alpha_2, x_A^2, p_A)$ for an identity ID_A.

- **Generate re-encryption key:** This algorithm is executed by the data owner, which takes as inputs public parameters Ψ, identity ID_A, public/private key pair (Pk_A, Sk_A) of data owner/sender, identity ID_B and data public key Pk_B of receiver. It generates a re-encryption key $Rk_{A\to B}$ as shown in Eq. (4.8) using Eq. (4.6) and (4.7):

$$d_B = \beta_1 h^{H_1(ID_B,\beta_1)} \tag{4.6}$$

$$d_{AB} = \mathcal{H}_3\left({d_B}^{k_A} || \ {p_B}^{x_A^1} || ID_A || Pk_A || ID_B || Pk_B\right) \tag{4.7}$$

$$Rk_{A\to B} = (x_A^1 \mathcal{H}_5(p_A) + k_A) d_{AB} \tag{4.8}$$

- **Encrypt:** It takes public parameters Ψ, message $m \in \mathcal{M}$ and A's public key Pk_A as inputs and performs following steps to generate first level ciphertext $c_A \in \mathcal{C}_1$:

  i. It verifies $g^{x_A^2} = \alpha_2 h^{H_2(ID_A,\alpha_1,\alpha_2)}$

  ii. It chooses $\sigma \in \{0,1\}^n$

  iii. It calculates $d_A = \alpha_1 h^{\mathcal{H}_1(ID_A,\alpha_1)}$ and $l = \mathcal{H}_3(m||\sigma||ID_A||p_A)$

  iv. It computes $c_1 = g^l$ and $c_2 = (m||\sigma) \oplus \mathcal{H}_4((d_A{}^{\mathcal{H}_5(p_A)}.p_A)^l)$ and returns $c_A = (c_1, c_2)$ for data owner A.

- **Re-encrypt:** It takes public parameters Ψ, first level ciphertext c_A and re-encryption key $Rk_{A\to B}$ as inputs, computes ${c_1}' = {c_1}^{Rk_{A\to B}}$ and ${c_2}' = c_2$ and returns re-encrypted ciphertext $c_B = ({c_1}', {c_2}')$ for receiver B or distinguished symbol Υ if c_A is invalid.

- **Decrypt:** It takes public parameters Ψ, private key Sk_u for cloud user $u \in \{A, B\}$ and ciphertext $c_u \in \mathcal{C}_{level}$, where $level \in \{A, B\}$ corresponding to user u as inputs and generates a plain text message $m \in \mathcal{M}$ or an error symbol Υ if c_u is invalid.

  i. Decrypt$_1$: To decrypt first level ciphertext $c_A = (c_1, c_2)$ with $Sk_A = (x_A^1, k_A)$, it computes $(m||\sigma) = c_2 \oplus \mathcal{H}_4\left({c_1}^{x_A^1 \mathcal{H}_5(p_A)+k_A}\right)$ and returns m if $g^{l'} = c_1$ holds, where $l' = \mathcal{H}_3(m||\sigma||ID_A||p_A)$, otherwise, returns an error symbol Υ if c_A is invalid.

ii. Decrypt$_2$: To decrypt second level ciphertext $c_B = (c_1', c_2')$ with $Sk_B = (x_B^1, k_B)$, it computes $(m||\sigma)$ in Eq. (4.11) using Eq. (4.9) and (4.10)

$$d_A = \alpha_1 h^{H_1(ID_A, \alpha_1)} \tag{4.9}$$

$$d_{BA} = \mathcal{H}_3\left({p_A}^{x_B^1} || {d_A}^{k_B} || ID_A || Pk_A || ID_B || Pk_B \right) \tag{4.10}$$

$$(m||\sigma) = c_2' \oplus \mathcal{H}_4\left((c_1')^{\frac{1}{d_{BA}}} \right) \tag{4.11}$$

and returns plaintext message m if $({d_A}^{\mathcal{H}_5(p_A).p_A})^{l' d_{BA}} = c_1'$ holds where $l' = \mathcal{H}_3(m||\sigma||ID_A||p_A)$ otherwise returns an error symbol Υ if c_B is invalid.

4.4 Cryptanalysis of Qin's et al scheme

The adversary performs following steps to launch attack against Qin's et al scheme.

1. Adversary chooses ID_A as a challenge identity and two more identities, i.e., ID_B and ID_C.
2. Adversary obtains challenge ciphertext ${c_A}^*$ from the challenger.
3. Adversary issues following queries to oracles to break confidentiality and win the game trivially:
   i. Adversary queries for user key extract oracle $\mathbb{O}_{u_key}$ for ID_B and obtains public/private key pair (p_B, k_B).
   ii. Adversary queries for user key extract oracle $\mathbb{O}_{u_key}$ for ID_C and obtains public/private key pair (p_C, k_C). It also queries for partial key extract oracle $\mathbb{O}_{p_key}$ for ID_C and obtains partial public/private key pair $(P_C, S_C) = (\alpha_1, \alpha_2, x_A^2, x_A^1)$. The challenger thus adds ID_C to the corrupted users list $\mathcal{L}_{CI}$ as it obtains full secret key $Sk_C = (S_C, k_C) = (x_C^1, k_C)$ of ID_C.
   iii. Adversary queries re-encryption key generate oracle $\mathbb{O}_{Rk}$ to obtain $Rk_{B\to C}$.
   iv. Adversary queries re-encryption key generate oracle $\mathbb{O}_{Rk}$ to obtain $Rk_{A\to B}$ and then re-encryption oracle $\mathbb{O}_{re-enc}$ to re-encrypt challenge ciphertext ${c_A}^* = ({c_1}^*, {c_2}^*)$ from ID_A to ${c_B}^* = ({c_1'}^*, {c_2'}^*)$ for ID_B.
4. Adversary computes partial secret key $S_B = x_B^1$ for ID_B using Eq. (4.14):

$$d_{CB} = \mathcal{H}_3\left(p_B{}^{x_C^1} || d_C{}^{k_C} || ID_B || Pk_B || ID_C || Pk_C\right) \quad (4.12)$$

$$\frac{Rk_{B\to C}}{d_{CB}} = (x_B^1 \mathcal{H}_5(p_B) + k_B) \quad (4.13)$$

$$x_B^1 = \left(\frac{Rk_{B\to C}}{d_{CB}} - k_B\right)\frac{1}{\mathcal{H}_5(p_B)} \quad (4.14)$$

5. Now adversary has access to the full secret key $Sk_B = (S_B, k_B) = (x_B^1, k_B)$ of ID_B. It can now decrypt $c_B{}^* = (c_1{}'^*, c_2{}'^*)$ easily to obtain the plaintext message m.

4.5 Proposed CLPRE scheme

The existing schemes in [92, 149, 151-154] are based on traditional PKI based PRE. Due to the wide count of patients and doctors associated with a telemedical system, management of public keys become cumbersome. IB-PRE schemes [149, 106, 155-161] and attribute-based PRE [170, 171] schemes are also inefficient as third party can compute private keys of all entities and is able to read messages violating HIPAA data privacy and security rules. To overcome expensive certificate management in PKI and key escrow problem in IB-PRE and attribute-based PRE, the proposed pairing free CL-PKC scheme adheres to HIPAA compliance.

The overall system architecture of cloud-based mPHRs sharing using proposed CL-PRE scheme is shown in Figure 4.1. It comprises four entities, i.e., data owner, CSP, KGC and data recipients. The data owner comprises patient's MCD, which detects patient health data continuously from the sensors in the patient's body. It enables one to collect one's own health data with the employment of applications, home gadgets such as smart scales coupled with wearable devices such as blood pressure wrist monitors and fitness trackers. An accelerating rate of provision of devices and applications inherently implies the patient's empowerment and ability to monitor all facets of one's health, ranging from body weight and personal fitness to the far end of the spectrum involving a look over the chronic/life-threatening conditions. KGC generates partial public/private key pairs based on users' identities and delivers private keys over a secure channel and public keys over an insecure channel to cloud users.

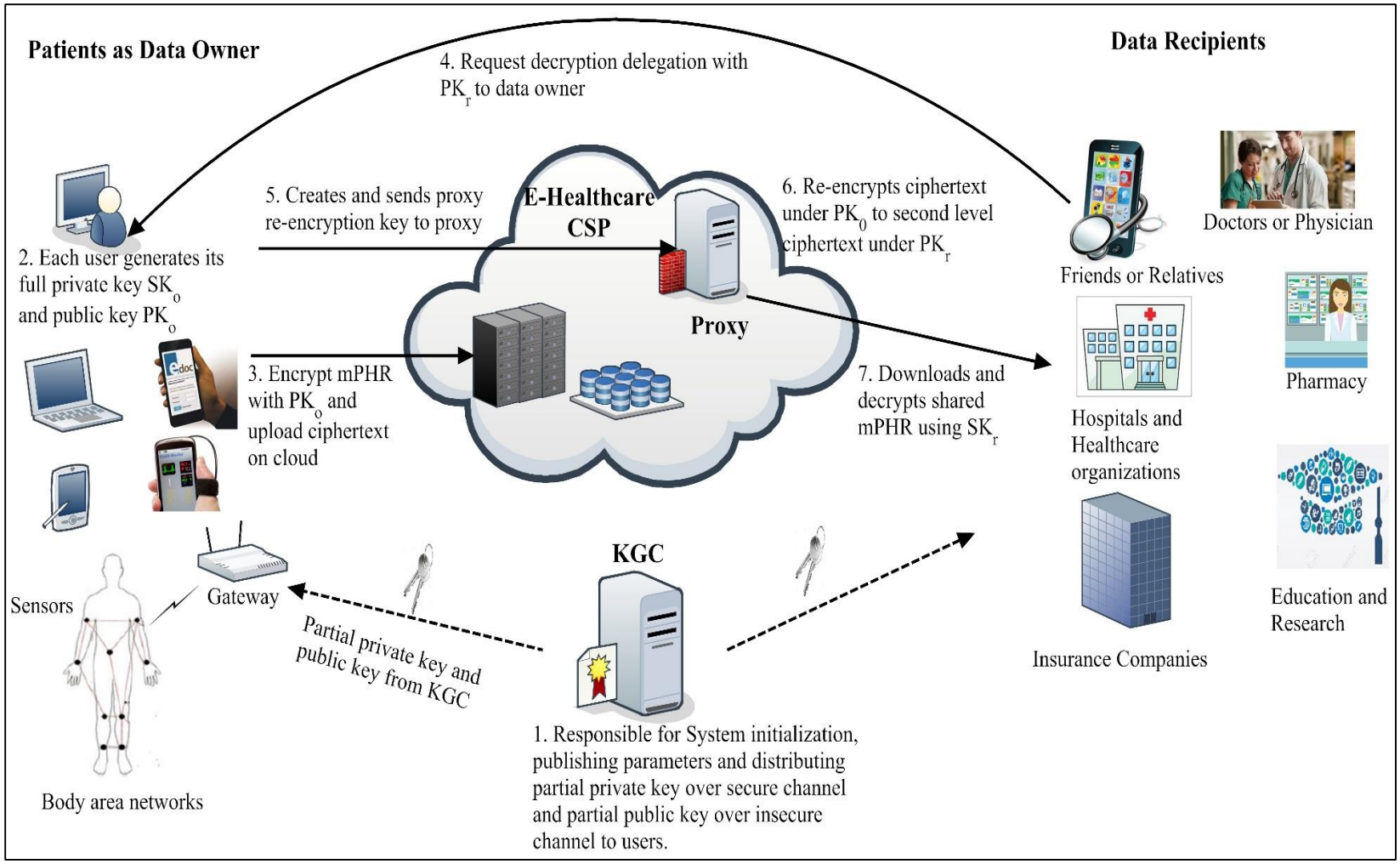

Figure 4.1: Architecture of cloud-based mPHRs sharing using proposed CL-PRE scheme

The data owner encrypts its mPHR using these keys before outsourcing to semi-trusted CSP. The CSP follows underlying protocol, algorithm and policies fairly but cannot guarantee confidentiality. The CSP, in collusion with data recipients, holds the possibility of launching attacks against data possessor. The data recipients include doctors, relatives, friends, healthcare organizations, insurance companies, etc. The CSP stores the encrypted data and makes it available for the genuine data owner and authorized data recipients only. A patient's general practitioner would be able to access the patient's records only with the patient's consent and would be able to use the data collected in getting a bigger picture of the patient's overall health. This enables the patient to play an active role in their own healthcare and well-being. Along with the access to patient's data for the clinical purpose, it can also be extended for using the data for researches, which is known as the secondary use of information. The data recipients request for decryption delegation to the data owner which, in turn, generates re-encryption key for each recipient and forwards re-encryption key to CSP through a secure channel. Cloud-resident proxy transforms ciphertext under data owner's public key into ciphertext under each data recipient's public key without any leakage of information. Each data recipient can then download the re-encrypted data from the cloud and decipher the data using his private key.

The proposed CL-PRE scheme consists of following 9 algorithms as below.

- **Setup:** KGC chooses a security parameter k, k-bit prime p and an elliptic curve E/F_p over prime finite field F_p. Let G be the cyclic subgroup of an elliptic curve points on E with P as generator of order q. KGC also chooses a master secret key $s \in Z_q^*$ and computes master public key $P_{Pub} = sP$. It also chooses 7 cryptographic collision-free hash functions as follows:

$$\mathcal{H}_1: \{0,1\}^* \times G \rightarrow Z_q^* \tag{4.15}$$
$$\mathcal{H}_2: G\ X\ G\ X\ \{0,1\}^* X\ G\ X\ G \rightarrow Z_q^*$$
$$\mathcal{H}_3: G^4 \rightarrow Z_q^*$$
$$\mathcal{H}_4: \{0,1\}^l\ X\ \{0,1\}^n\ X\ \{0,1\}^*\ X\ G\ X\ G \rightarrow Z_q^*$$
$$\mathcal{H}_5: G \rightarrow \{0,1\}^{l+n}$$

$$\mathcal{H}_6: \{0,1\}^* X\ G^3 \rightarrow Z_q^*$$

$$\mathcal{H}_7: G\ X\ \{0,1\}^{l+n}\ X\ G \rightarrow Z_q^*$$

where l is bit length of plaintext message and n is random bit string. The plaintext message space is $\mathcal{M} = \{0,1\}^l$ and ciphertext space is $\mathcal{C} = \{0,1\}^{l+n}$. This algorithm returns public parameters $\Psi = \{E, F_p, G, P,\ \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5, \mathcal{H}_6, \mathcal{H}_7, P_{Pub}\}$ for publishing.

- **Extract partial key:** KGC randomly chooses $r_{A_1}, r_{A_2}, v_A \in Z_q^*$ and computes $R_{A_1} = r_{A_1}P,\ R_{A_2} = r_{A_2}P,\ V_A = v_A P$. It computes $S_{A_1} = \left(r_{A_1} + s\mathcal{H}_1\left(ID_A, R_{A_1}\right)\right),\ S_{A_2} = \left(r_{A_2} + s\mathcal{H}_1\left(ID_A, R_{A_2}\right)\right)$ and $\beta_A = v_A + s\mathcal{H}_6\left(ID_A, V_A, R_{A_1}, R_{A_2}\right)$. KGC transmits $S_A = \left(S_{A_1}, S_{A_2}\right)$ as partial private key over secure channel and $R_A = \left(R_{A_1}, R_{A_2}, V_A, \beta_A\right)$ as partial public key over an insecure channel to entity A.

- **Set secret value:** Each identity A randomly generates $k_{A_1}, k_{A_2} \in Z_q^*$ as secret values for ID_A.

- **Generate private key:** This algorithm accepts public parameters Ψ, partial private key S_A and user secret key (k_{A_1}, k_{A_2}) as inputs and generates full private key $Sk_A = \left(S_{A_1}, S_{A_2}, k_{A_1}, k_{A_2}\right)$ for identity ID_A.

- **Generate public key:** This algorithm accepts public parameters Ψ, partial public key R_A, user's secret key (k_{A_1}, k_{A_2}) as inputs, computes $u_{A_1} = k_{A_1}P, u_{A_2} = k_{A_2}P$ and generates full public key $Pk_A = (R_{A_1}, R_{A_2}, V_A, \beta_A, u_{A_1}, u_{A_2})$ for identity ID_A.

- **Generate re-encryption key:** This algorithm takes public parameters Ψ, identity ID_A, public/private key pair (Pk_A, Sk_A) of data owner/sender, identity ID_B and public key Pk_B of receiver as inputs. It generates re-encryption key $Rk_{A \rightarrow B}$ in Eq. (4.18) using Eq. (4.17) and Eq. (4.16):

$$d_B = R_{B_1} + \mathcal{H}_1(ID_B, R_{B_1})P_{Pub} \tag{4.16}$$

$$d_{AB} = \mathcal{H}_2\left(k_{A_1} d_B, S_{A_1} u_{B_2}, ID_A, ID_B, Pk_A, Pk_B\right) \tag{4.17}$$

$$Rk_{A\to B} = \left(\left(S_{A_1} + k_{A_1}\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2} + k_{A_2}\right) d_{AB} \qquad (4.18)$$

- **Encrypt:** It takes public parameters Ψ, message $m \in \mathcal{M}$ and A's public key Pk_A as inputs and performs following steps to generate first level ciphertext $c_A \in \mathcal{C}_1$:

  i. It verifies the partial private key as: $S_{A_1}P = R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub}$ and $S_{A_2}P = R_{A_2} + \mathcal{H}_1(ID_A, R_{A_2})P_{Pub}$. If invalid, return Υ.

  ii. It checks the validity of public keys of any identity ID_i as: $\beta_i P = V_i + \mathcal{H}_6\left(ID_i, V_i, R_{i_1}, R_{i_2}\right)P_{Pub}$. If invalid, return Υ.

  iii. It chooses $\sigma \in \{0,1\}^n$, $\mu \in Z_q^*$.

  iv. It calculates $t = \mathcal{H}_4\left(m, \sigma, ID_A, u_{A_1}, u_{A_2}\right)$.

  v. It computes $c_1 = tP$, $c_3 = \mu P$, $c_2 = (m||\sigma) \oplus \mathcal{H}_5\left(t\left(\left(R_{A_1} + \mathcal{H}_1\left(ID_A, R_{A_1}\right)P_{Pub} + u_{A_1}\right)\mathcal{H}_3\left(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}\right) + R_{A_2} + \mathcal{H}_1\left(ID_A, R_{A_2}\right)P_{Pub} + u_{A_2}\right)\right)$ and $c_4 = \mu + t\mathcal{H}_7(c_1, c_2, c_3)$. It returns $c_A = (c_1, c_2, c_3, c_4)$ for proxy server.

- **Proxy Re-encrypt:** The cloud-resident proxy server takes public parameters Ψ, first level ciphertext c_A and re-encryption key $Rk_{A\to B}$ as inputs. It first validates the received ciphertext as $c_4 P = c_3 + \mathcal{H}_7(c_1, c_2, c_3)\, c_1$. If verification is successful, it computes $c_1' = c_1 Rk_{A\to B}$ and $c_2' = c_2$ and returns re-encrypted ciphertext $c_B = (c_1', c_2')$ for receiver B. Otherwise, it returns a distinguished symbol Υ.

- **Decrypt:** It takes public parameters Ψ, private key Sk_u for cloud user $u \in \{A, B\}$ and ciphertext $c_u \in \mathcal{C}_{level}$, where $level \in \{A, B\}$ corresponding to user u and generates corresponding plain text message $m \in \mathcal{M}$ or an error symbol Υ if c_u is invalid.

  i. Decrypt$_1$: To decrypt first level ciphertext $c_A = (c_1, c_2, c_3, c_4)$ with $Sk_A = \left(S_{A_1}, S_{A_2}, k_{A_1}, k_{A_2}\right)$, the data owner A first validates the received ciphertext as $c_4 P = c_3 + \mathcal{H}_7(c_1, c_2, c_3)\, c_1$. If verification is successful,

it computes $(m||\sigma) = c_2 \oplus \mathcal{H}_5\Big(\Big((S_{A_1} + k_{A_1})\,\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2} + k_{A_2}\Big)\,c_1\Big)$ and returns m if $c_1 = t'P$ holds, where $t' = \mathcal{H}_4(m, \sigma, ID_A, u_{A_1}, u_{A_2})$, otherwise, returns an error symbol Υ.

ii. Decrypt$_2$: To decrypt second level ciphertext $c_B = (c_1', c_2')$ with $Sk_B = (S_B, k_{B_1}, k_{B_2})$, data receiver B computes $(m||\sigma)$ in Eq. (4.21) using Eq. (4.19) and Eq. (4.20) as follows:

$$d_A = R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})\,P_{Pub} \tag{4.19}$$

$$d_{BA} = \mathcal{H}_2(u_{A_1}S_{B_1}, d_A k_{B_2}, ID_A, ID_B, Pk_A, Pk_B) \tag{4.20}$$

$$(m||\sigma) = c_2' \oplus \mathcal{H}_5((c_1')/d_{BA}) \tag{4.21}$$

and returns a plaintext message m if $\Big(\Big(\big(R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub} + u_{A_1}\big)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + R_{A_2} + \mathcal{H}_1(ID_A, R_{A_2})P_{Pub} + u_{A_2}\Big)\,d_{BA}\Big) = c_1'$ holds; otherwise, it returns an error symbol Υ.

The correctness of decryption of the proposed CL-PRE scheme is proved as follows:

First d_{BA} and d_{AB} are proved equal.

$$d_{BA} = \mathcal{H}_2(u_{A_1}S_{B_1}, d_A k_{B_2}, ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}PS_{B_1}, (R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub})k_{B_2}, ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}(S_{B_1}P), (R_{A_1}k_{B_2} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub}k_{B_2}), ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}(R_{B_1} + \mathcal{H}_1(ID_B, R_{B_1})P_{Pub}), (R_{A_1}k_{B_2} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub}k_{B_2}), ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}d_B, (r_{A_1}P.k_{B_2} + \mathcal{H}_1(ID_A, R_{A_1})sPk_{B_2}), ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}d_B, (r_A + \mathcal{H}_1(ID_A, R_{A_1})s)k_{B_2}P), ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = \mathcal{H}_2(k_{A_1}d_B, S_{A_1}u_{B_2}, ID_A, ID_B, Pk_A, Pk_B)$$
$$d_{BA} = d_{AB}$$

The plaintext message m is verified if Eq. (4.22) holds.

$$\left(\left(\left(R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub} + u_{A_1}\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + R_{A_2} + \mathcal{H}_1(ID_A, R_{A_2})P_{Pub} + u_{A_2}\right) d_{BA}\right) = c_1' \quad (4.22)$$

$$c_1' = Rk_{A\to B}\,.P$$

$$c_1' = \left(\left(S_{A_1} + k_{A_1}\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2} + k_{A_2}\right) d_{AB}.P$$

$$c_1' = \left(\left(S_{A_1}P + k_{A_1}P\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + S_{A_2}P + k_{A_2}P\right) d_{AB}$$

$$c_1' = \left(\left(\left(r_{A_1} + s\mathcal{H}_1(ID_A, R_{A_1})\right)P + k_{A_1}P\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + \left(r_{A_2} + s\mathcal{H}_1(ID_A, R_{A_2})\right)P + k_{A_2}P\right) d_{BA}$$

$$c_1' = \left(\left(r_{A_1}P + sP\,\mathcal{H}_1(ID_A, R_{A_1}) + k_{A_1}P\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + r_{A_2}P + sP\,\mathcal{H}_1(ID_A, R_{A_2}) + k_{A_2}P\right) d_{BA}$$

$$c_1' = \left(\left(R_{A_1} + \mathcal{H}_1(ID_A, R_{A_1})P_{Pub} + u_{A_1}\right)\mathcal{H}_3(R_{A_1}, R_{A_2}, u_{A_1}, u_{A_2}) + R_{A_2} + \mathcal{H}_1(ID_A, R_{A_2})P_{Pub} + u_{A_2}\right) d_{BA}$$

4.6 Security Analysis

In this section, the security of the proposed CL-PRE scheme has been analyzed with the following two lemmas.

Lemma 1: The proposed CL-PRE scheme is CL-PRE_IND-CCA_ORG secure against adversary $\mathcal{A}_1$ in random oracle model assuming underlying Elliptic Curve-Computational Diffie Hellman (EC-CDH) assumption is hard to solve.

Suppose $\mathcal{H}_i$ $(1 \le i \le 7)$ are random oracles and there exists a $\left(t, v, \mathbb{q}_{p_key}, \mathbb{q}_{u_key}, \mathbb{q}_{Pk}, \mathbb{q}_{Sk}, \mathbb{q}_{Pk_rep}, \mathbb{q}_{Rk}, \mathbb{q}_{re-enc}, \mathbb{q}_{dec1}, \mathbb{q}_{dec2}\right)$ $CL-PRE_IND-CCA_ORG$ adversary $\mathcal{A}_1$ with advantage v against the original first level ciphertext security $Game_1$, making at most $\mathbb{q}_{p_key}$ queries to $\mathbb{O}_{p_key}$, $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries

to $\mathbb{O}_{Pk}$,$\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$, $\mathbb{q}_{Pk_rep}$ queries to $\mathbb{O}_{Pk_rep}$, $\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$, $\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$, $\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$, $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$ and $\mathbb{q}_{\mathcal{H}_i}$ queries to random oracles $\mathcal{H}_i$ $(1 \leq i \leq 7)$, then there exists an algorithm $\mathfrak{B}$, which solves EC-CDH problem with advantage v':

$$v' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_5}}\left(2v\frac{1}{e(1+\mathbb{q}_{p_key}+\mathbb{q}_{Sk}+\mathbb{q}_{Rk})} - \frac{\mathbb{q}_{\mathcal{H}_4}}{2^{l+n}} - \tau - \frac{\mathbb{q}_{re-enc}}{\mathbb{q}} - \mathbb{q}_{Dec}\left(\frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1-{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1-{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{2}/_{\mathbb{q}}\right)\right)$$

within time

$$t' \leq t + \big(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{\mathcal{H}_5} + \mathbb{q}_{\mathcal{H}_6} + \mathbb{q}_{\mathcal{H}_7} + \mathbb{q}_{p_key} + \mathbb{q}_{u_key} + \mathbb{q}_{Pk} + \mathbb{q}_{Sk} + \mathbb{q}_{PK_rep} + \mathbb{q}_{Rk} + \mathbb{q}_{re-enc} + \mathbb{q}_{dec1} + \mathbb{q}_{dec2}\big)O(1) + (3\mathbb{q}_{p_key} + 2\mathbb{q}_{u_key} + 2\mathbb{q}_{Pk} + 4\mathbb{q}_{Rk} + \mathbb{q}_{re-enc} + 2\mathbb{q}_{dec1} + 5\mathbb{q}_{dec2})t_{PM}$$

where t_{PM} denotes the point multiplication time in group G, τ denotes the probability $\mathcal{A}_1$ can distinguish between correctly formed re-encryption keys and an incorrectly formed re-encryption keys, and v denotes the advantage adversary may have over EUF-CMA security of Schnorr signature within time t.

Proof: Consider an adversary $\mathcal{A}_1$ against IND-CCA security of the proposed scheme. An algorithm $\mathfrak{B}$ is designed against EC-CDH problem. Algorithm $\mathfrak{B}$ is challenged with an instance of EC-CDH problem $(P, xP, yP) \in G_q$ with unknown x and y, the problem is to compute xyP.

Initialization: To solve EC-CDH problem, the algorithm $\mathfrak{B}$ sets master secret key $s = x$ and master public key $P_{Pub} = xP$. Hash functions $\mathcal{H}_i$ $(1 \leq i \leq 7)$ are simulated as random oracles controlled by $\mathfrak{B}$. $\mathfrak{B}$ sends system parameters $\Psi = \{E, F_p, G, P, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{H}_5, \mathcal{H}_6, \mathcal{H}_7, P_{Pub}\}$ to $\mathcal{A}_1$. Algorithm $\mathfrak{B}$ simulates the role of

challenger in various phases of formal security game and plays $Game - I_{A_1}^{CL-PRE_IND-CCA_ORG}$ with $\mathcal{A}_1$.

Random oracle queries: Algorithm $\mathfrak{B}$ maintains 7 hash lists $\mathcal{L}_{\mathcal{H}_i}$ $(1 \leq i \leq 7)$ to avoid inconsistency between responses of hash queries. These hash lists are initially empty and $\mathfrak{B}$ responds to queries from $\mathcal{A}_1$ as follows:

- $\mathcal{H}_1$ random queries: After receiving a query for (ID, R) from an adversary $\mathcal{A}_1$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_1}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_1 as an answer; otherwise, it chooses a random value $z_1 \in Z_q^*$, adds $<(ID, R), z_1>$ to $\mathcal{L}_{\mathcal{H}_1}$, and returns z_1 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_2$ random queries: After receiving a query for $(k_{i_1} d_j, S_{i_1} u_{j_2}, ID_i, ID_j, Pk_i, Pk_j)$ from an adversary $\mathcal{A}_1$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_2}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_2 as an answer; otherwise, it chooses a random value $z_2 \in Z_q^*$, adds $<(k_{i_1} d_j, S_{i_1} u_{j_2}, ID_i, ID_j, Pk_i, Pk_j), z_2>$ to $\mathcal{L}_{\mathcal{H}_2}$, and returns z_2 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_3$ random queries: After receiving a query for $(R_1, R_2. u_1, u_2)$ from an adversary $\mathcal{A}_1$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_3}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_3 as an answer; otherwise, it chooses a random value $z_3 \in Z_q^*$, adds $<(R_1, R_2, u_1, u_2), z_3>$ to $\mathcal{L}_{\mathcal{H}_3}$, and returns z_3 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_4$ random queries: After receiving a query for $(m, \sigma, ID, u_1, u_2)$ from an adversary $\mathcal{A}_1$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_4}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_4 as an answer; otherwise, it chooses a random value $z_4 \in Z_q^*$, adds $<(m, \sigma, ID, u_1, u_2), z_4>$ to $\mathcal{L}_{\mathcal{H}_4}$, and returns z_4 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_5$ random queries: After receiving a query for (K) from adversary $\mathcal{A}_1$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_5}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_5 as an

answer; otherwise, it chooses a random value $z_5 \in Z_q^*$, adds $<(K), z_5>$ to $\mathcal{L}_{\mathcal{H}_5}$, and returns z_5 as an answer to $\mathcal{A}_1$ in response to its query.

- $\mathcal{H}_6$ random queries: After receiving a query for (ID, V, R_1, R_2) from an adversary $\mathcal{A}_I$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_6}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_6 as an answer; otherwise, it chooses a random value $z_6 \in Z_q^*$, adds $<(ID, V, R_1, R_2), z_6>$ to $\mathcal{L}_{\mathcal{H}_6}$, and returns z_6 as an answer to $\mathcal{A}_I$ in response to its query.
- $\mathcal{H}_7$ random queries: After receiving a query for (c_1, c_2, c_3) from an adversary $\mathcal{A}_I$, $\mathfrak{B}$ probes hash-list $\mathcal{L}_{\mathcal{H}_7}$ for corresponding value. If an entry is found, $\mathfrak{B}$ returns z_7 as an answer; otherwise, it chooses a random value $z_7 \in Z_q^*$, adds $<(c_1, c_2, c_3), z_7>$ to $\mathcal{L}_{\mathcal{H}_7}$, and returns z_7 as an answer to $\mathcal{A}_I$ in response to its query.

Phase-I: Algorithm $\mathfrak{B}$ maintains the lists for issued public, private, partial, re-encryption keys and responds to $\mathcal{A}_1$'s queries as follows:

- Partial key extract queries: On receiving an input query ID (whose public key has not yet been replaced) from $\mathcal{A}_1$, $\mathfrak{B}$ first checks $\mathcal{L}_{par_key}$ containing tuples of the form $< ID, (S_1, S_2, R_1, R_2, V, \beta), (r_1, r_2), coin >$ and responds as below:

  i. If ID is found and $coin = 0$, $\mathfrak{B}$ returns corresponding partial key $(S_1, S_2, R_1, R_2, V, \beta)$as answer to $\mathcal{A}_1$.

  ii. If $coin = 1$, $\mathfrak{B}$ aborts the simulation.

  iii. Otherwise, $\mathfrak{B}$ picks a random number $v, r_1, r_2 \in Z_q^*$ and computes $R_1 = r_1 P$, $R_2 = r_2 P$ and $V = vP$. The random oracle queries (ID, R_1), (ID, R_2) and (ID, R_1, R_2, V) are issued to oracles $\mathcal{H}_1$ and $\mathcal{H}_6$ and result $z_1, z_2 \in \mathcal{L}_{\mathcal{H}_1}$ and $z_6 \in \mathcal{L}_{\mathcal{H}_6}$ is obtained. $\mathfrak{B}$ further computes $S_1 = (r_1 + az_1), S_2 = (r_2 + az_2), \beta = v + az_6$ and $< ID, (S_1, S_2, R_1, R_2, V, \beta), (r_1, r_2), coin = 0 >$ to list $\mathcal{L}_{Par_key}$ and returns partial key $(S_1, S_2, R_1, R_2, V, \beta)$ as an answer to $\mathcal{A}_I$.

- Public key request queries: On receiving an input query ID from $\mathcal{A}_1$, $\mathfrak{B}$ first checks public-key list $\mathcal{L}_{pub_key}$ containing tuples of the form $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ and responds as below:

    i. If the corresponding entry is found in $\mathcal{L}_{pub_key}$, $\mathfrak{B}$ returns corresponding public key $Pk_{ID} = (R_1, R_2, V, \beta, u_1, u_2)$ as an answer to $\mathcal{A}_1$.

    ii. Else, $\mathfrak{B}$ randomly picks a $coin \in \{0,1\}$ such that probability $P[coin = 0] = \lambda$ and runs aforementioned partial key extract oracle with ID as an input to obtain partial key $(S_1, S_2, R_1, R_2, V, \beta)$ as an answer.

    iii. If $coin = 0$, $\mathfrak{B}$ picks $k_1, k_2 \in Z_q^*$ and computes $u_1 = k_1 P, u_2 = k_2 P$. It adds tuple $<ID, (S_1, S_2, k_1, k_2) >$ to $\mathcal{L}_{pri_key}$, tuple $<ID, (k_1, k_2), (u_1, u_2) >$ to $\mathcal{L}_{usr_key}$ and tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ to $\mathcal{L}_{pub_key}$. It returns corresponding public key $Pk_{ID} = (R_1, R_2, V, \beta, u_1, u_2)$ as an answer to adversary $\mathcal{A}_1$.

    iv. If $coin = 1$, $\mathfrak{B}$ picks $k_1, k_2 \in Z_q^*$ and computes $u_1 = xk_1 P$, $u_2 = k_2 P$. It adds tuple $<ID, (S_1, S_2, k_1, k_2), coin >$ to $\mathcal{L}_{pri_key}$, tuple $<ID, (k_1, k_2), (u_1, u_2) >$ to $\mathcal{L}_{usr_key}$ and tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ to $\mathcal{L}_{pub_key}$. It returns corresponding public key $Pk_{ID} = (R_1, R_2, V, \beta, u_1, u_2)$ as an answer to adversary $\mathcal{A}_1$.

- Private key extract queries: On receiving an input query ID from $\mathcal{A}_1$, $\mathfrak{B}$ issues aforementioned public key extract query with an input ID to get a tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ and responds as follows:

i. If $coin = 0$, $\mathfrak{B}$ searches the $\mathcal{L}_{pri_key}$ with ID as a key value. If an entry is found, it returns the corresponding private key $Sk_{ID} = (S_1, S_2, k_1, k_2)$ as an answer to adversary $\mathcal{A}_1$.

ii. If $coin = \perp$, $\mathfrak{B}$ rejects and returns failure since public key corresponding to an identity ID has already been replaced.

iii. If $coin = 1$, $\mathfrak{B}$ aborts the simulation.

- User key extract queries: On receiving an input query ID whose public key has not yet been replaced. $\mathfrak{B}$ responds as follows:

  i. $\mathfrak{B}$ probes $\mathcal{L}_{usr_key}$ containing tuples of the form $< ID, (k_1, k_2), (u_1, u_2) >$ with ID as a key value. If an entry is found, it returns the corresponding user key $(k_1, k_2), (u_1, u_2)$ as an answer to adversary $\mathcal{A}_1$.

  ii. If entry is not found, $\mathfrak{B}$ queries $\mathcal{L}_{pub_key}$ containing tuples of the form $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ with ID as a key value. If $coin = \perp$, $\mathfrak{B}$ rejects and returns failure since public key corresponding to identity ID has already been replaced. Otherwise, $\mathfrak{B}$ again searches $\mathcal{L}_{usr_key}$ with ID as a key value and returns a user key $(k_1, k_2), (u_1, u_2)$ to adversary $\mathcal{A}_1$. Such a tuple is guaranteed to exist in $\mathcal{L}_{usr_key}$ now.

- Public key replace queries: The adversary $\mathcal{A}_1$ can replace public key $Pk_{ID} = (R_1, R_2, V, \beta, u_1, u_2)$ of any identity ID with any value of its choice $Pk_{ID}{}' = (R_1{}', R_2{}', V', \beta', u_1{}', u_2{}')$. $\mathfrak{B}$ checks the validity of public key. If check fails, it rejects and returns failure. Otherwise, it searches a tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin >$ in $\mathcal{L}_{pub_key}$ and replaces $(R_1, R_2, V, \beta, u_1, u_2)$ with $(R_1{}', R_2{}', V', \beta', u_1{}', u_2{}')$ and $coin = \perp$ if a tuple is found. In case tuple is not found, it adds $< ID, (R_1{}', R_2{}', V', \beta', u_1{}', u_2{}'), coin = \perp >$ in $\mathcal{L}_{pub_key}$.

- Re-encryption key generate queries: On receiving input identities (ID_i, ID_j) from adversary $\mathcal{A}_1$, $\mathfrak{B}$ searches the $\mathcal{L}_{re-enc_key}$ containing tuples of the form $< (ID_i, ID_j), Rk_{i\to j}, d_j, \delta >$ and responds as follows:

    i. If an entry is found for input (ID_i, ID_j), $\mathfrak{B}$ returns the corresponding value of re-encryption key $Rk_{i\to j}$ to adversary $\mathcal{A}_1$.

    ii. Otherwise, $\mathfrak{B}$ issues aforementioned public key extract query with an input ID_i to get a tuple $< ID_i, (R_{i_1}, R_{i_2}, V_i, \beta_i, u_{i_1}, u_{i_2}), coin_i > \in \mathcal{L}_{pub_key}$. If $coin_i = 0$ and public key of ID_i has not been replaced, $\mathfrak{B}$ issues private key extract query with an input ID_i to obtain the private key $Sk_{ID_i} = (S_{i_1}, S_{i_2}, k_{i_1}, k_{i_2})$ and public key extract query with an input ID_j to obtain the public key $Pk_{ID_j} = (R_{j_1}, R_{j_2}, V_i, \beta_j, u_{j_1}, u_{j_2})$. $\mathfrak{B}$ computes $d_j = R_{j_1} + \mathcal{H}_1(ID_j, R_{j_1})\, P_{Pub}$ and $d_{ij} = \mathcal{H}_2(k_{i_1} d_j, S_{i_1} u_{j_2}, ID_i, ID_j, Pk_i, Pk_j)$. It further calculates re-encryption key $Rk_{i\to j} = \left((S_{i_1} + k_{i_1})\mathcal{H}_3(R_{i_1}, R_{i_2}, u_{i_1}, u_{i_2}) + S_{i_2} + k_{i_2}\right) d_{ij}$, adds $< (ID_i, ID_j), Rk_{i\to j}, d_j, \delta = 1 >$ to $\mathcal{L}_{re-enc_key}$ and returns $Rk_{i\to j}$ to adversary $\mathcal{A}_1$. If $coin = \perp$, $\mathfrak{B}$ rejects and returns failure. If $coin = 1$, $\mathfrak{B}$ aborts the simulation.

- Re-encryption queries: On receiving an input query from adversary $\mathcal{A}_1$ for re-encryption of first level ciphertext $c_A = (c_1, c_2, c_3, c_4)$ for identities ID_i to ID_j, $\mathfrak{B}$ first checks validity of c_A using $c_4 P = c_3 + \mathcal{H}_7(c_1, c_2, c_3)\, c_1$. (c_3, c_4) is used only for validation of ciphertext and (c_1, c_2) is used for further re-encryption. If check fails, it rejects and returns failure. Otherwise, $\mathfrak{B}$ searches $\mathcal{L}_{pub_key}$ for the tuple $< ID_i, (R_{i_1}, R_{i_2}, V_i, \beta_i, u_{i_1}, u_{i_2}), coin_i > \in \mathcal{L}_{pub_key}$ and responds as follows:

    i. If $coin_i = 0$ and public key of ID_i has not been replaced, $\mathfrak{B}$ then queries aforementioned re-encryption key oracle to obtain corresponding re-encryption key $Rk_{i\to j}$. It computes second level

ciphertext $c_B = (c_1', c_2')$ as $c_1' = c_1 Rk_{i\to j}$ and $c_2' = c_2$. $\mathfrak{B}$ returns $c_B = (c_1', c_2')$ to adversary $\mathcal{A}_1$.

ii. If $coin_i = 1$, $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$. It computes $c_1 = yP$ and calculates $Rk_{i\to j} \in G$ and adds it to the $< (ID_i, ID_j), Rk_{i\to j}, d_j = 0, \delta = 0 >$ to list $\mathcal{L}_{re-enc_key}$. It further calculates $c_1' = c_1 Rk_{i\to j}$ and $c_2' = (m||\sigma) \oplus \mathcal{H}_5\Big(y\Big(\big(R_{i_1} + \mathcal{H}_1(ID_i, R_{i_1})P_{Pub} + u_{i_1}\big)\mathcal{H}_3\big(R_{i_1}, R_{i_2}, u_{i_1}, u_{i_2}\big) + R_{i_2} + \mathcal{H}_1(ID_i, R_{i_2})P_{Pub} + u_{i_2}\Big)\Big)$. $\mathfrak{B}$ returns $c_B = (c_1', c_2')$ to adversary $\mathcal{A}_1$.

- Strong decrypt-1 queries: On receiving an input (ID, c_A) where $c_A = (c_1, c_2, c_3, c_4)$ is first level ciphertext from $\mathcal{A}_1$, $\mathfrak{B}$ first searches tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin > \in \mathcal{L}_{pub_key}$ and responds as follows:

  i. If $coin = 0$ and public key of ID_i has not been replaced, $\mathfrak{B}$ issues aforementioned private key extract query with ID as an input to obtain private key $Sk_{ID} = (S_1, S_2, k_1, k_2) \in \mathcal{L}_{pri_key}$. $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$ and computes $(m||\sigma) = c_2 \oplus \mathcal{H}_5\Big(\big((S_1 + k_1)\,\mathcal{H}_3(R_1, R_2, u_1, u_2) + S_2 + k_2\big)\,c_1\Big)$ and gives result m to $\mathcal{A}_1$.

  ii. Otherwise, $\mathfrak{B}$ checks validity of ciphertext c_A. If check fails, $\mathfrak{B}$ rejects and returns failure.

  iii. If $coin = 1$, $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$, computes $c_1 = yP$ and $c_2 = (m||\sigma) \oplus \mathcal{H}_5\Big(y\big((R_1 + \mathcal{H}_1(ID, R_1)P_{Pub} + u_1)\mathcal{H}_3(R_1, R_2, u_1, u_2) + R_2 + \mathcal{H}_1(ID, R_2)P_{Pub} + u_2\big)\Big)$. It returns the result m on this ciphertext to adversary $\mathcal{A}_1$.

- Strong decrypt-2 queries: On receiving an input $(ID,\ c_B)$ where $c_B = (\ c_1',\ c_2')$ is a second level ciphertext from $\mathcal{A}_1$, $\mathfrak{B}$ first searches tuple $< ID, (R_1, R_2, V, \beta, u_1, u_2), coin > \in \mathcal{L}_{pub_key}$ and responds as follows:

    i. If $coin = 0$ and public key of ID_i has not been replaced, $\mathfrak{B}$ issues aforementioned private key extract query with ID as input to obtain private key $Sk_{ID} = (S_1, S_2, k_1, k_2) \in \mathcal{L}_{pri_key}$. It queries re-encryption key generate algorithm to obtain d_{ji}. $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$ and computes $(m||\sigma) = c_2' \oplus \mathcal{H}_5\big((c_1')/d_{ji}\big)$ and gives result m to $\mathcal{A}_1$.

    ii. Otherwise, $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$, computes $c_1 = yP$ and $c_2 = (m||\sigma) \oplus \mathcal{H}_5\Big(y\big((R_1 + \mathcal{H}_1(ID, R_1)P_{Pub} + u_1)\mathcal{H}_3(R_1, R_2, u_1, u_2) + R_2 + \mathcal{H}_1(ID, R_2)P_{Pub} + u_2\big)\Big)$. It re-encrypts the ciphertext (c_1, c_2) using re-encryption algorithm simulated above. It returns the result m on this re-encrypted ciphertext to adversary $\mathcal{A}_1$.

Challenge Phase: Once Phase-I is over, adversary $\mathcal{A}_1$ outputs a target identity ID^* and two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ on which it desires to be confronted. It must be ensured that the adversary has not queried partial key of target identity ID^*. $\mathfrak{B}$ responds to the query as follows:

i. $\mathfrak{B}$ issues public key extract query with ID^* as an input to get a tuple $< ID^*, ({R_1}^*, {R_2}^*, V^*, \beta^*, {u_1}^*, {u_2}^*), coin^* > \in \mathcal{L}_{pub_key}$.

ii. If $coin^* = 0$, $\mathfrak{B}$ aborts the simulation.

iii. If $coin^* = 1$, $\mathfrak{B}$ picks randomly $\sigma \in \{0,1\}^n$, ß $\in \{0,1\}$ and computes first level ciphertext ${c_1}^* = yP$ and second level ciphertext ${c_2}^*$ as follows:

$$c_2{}^* = (m_ß||\sigma) \oplus \mathcal{H}_5\Big(y\big((R_1{}^* + \mathcal{H}_1(ID^*, R_1{}^*)P_{Pub} + u_1{}^*)\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + R_2{}^* + \mathcal{H}_1(ID^*, R_2{}^*)P_{Pub} + u_2{}^*\big)\Big)$$

$$c_2{}^* = (m_ß||\sigma) \oplus \mathcal{H}_5\big((yR_1{}^* + y\mathcal{H}_1(ID^*, R_1{}^*)P_{Pub} + yu_1{}^*)\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + yR_2{}^* + y\mathcal{H}_1(ID^*, R_2{}^*)P_{Pub} + yu_2{}^*\big)$$

$$c_2{}^* = (m_ß||\sigma) \oplus \mathcal{H}_5\big((yR_1{}^* + xy\mathcal{H}_1(ID^*, R_1{}^*)P + yxk_1{}^*P)\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + yR_2{}^* + xy\mathcal{H}_1(ID^*, R_2{}^*)P + yu_2{}^*\big)$$

$$c_2{}^* = (m_ß||\sigma) \oplus \mathcal{H}_5(yR_1{}^*\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + xyP\mathcal{H}_1(ID^*, R_1{}^*)\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + k_1{}^*xyP\,\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + yR_2{}^* + xy\mathcal{H}_1(ID^*, R_2{}^*)P + yu_2{}^*)$$

$\mathfrak{B}$ returns $C^* = (c_1{}^*, c_2{}^*)$ as challenge ciphertext to $\mathcal{A}_1$

Phase-II: Adversary $\mathcal{A}_1$ continues to issue queries as in Phase-I but with the restrictions as mentioned in $Game_1$ for the security of original first level ciphertext in Section 2.4.2.1.

Guess: Finally, the adversary $\mathcal{A}_1$ outputs a guess $ß^* \in \{0,1\}$. Algorithm $\mathfrak{B}$ randomly picks a tuple $<K, z_5>$ from hash-list $\mathcal{L}_{\mathcal{H}_5}$ where $z_5 = \mathcal{H}_5(K)$. Since $\mathcal{A}_1$ is restricted from replacing public key of target identity ID^*, $\mathfrak{B}$ knows user's secret values $(k_1{}^*, k_2{}^*, r_1{}^*, r_2{}^*)$ so it computes K' and outputs K' as solution to EC-CDH instance.

$$K' = \left(\frac{K - yPr_1{}^*\,\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) - yPr_2{}^* - k_2{}^*yP}{\mathcal{H}_1(ID^*, R_1{}^*)\mathcal{H}_3\big(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*\big) + k_1{}^*\mathcal{H}_3\big(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*\big) + \mathcal{H}_1(ID^*, R_2{}^*)}\right)$$

where $K = y\big((R_1{}^* + \mathcal{H}_1(ID^*, R_1{}^*)P_{Pub} + u_1{}^*)\mathcal{H}_3(R_1{}^*, R_2{}^*, u_1{}^*, u_2{}^*) + R_2{}^* + \mathcal{H}_1(ID^*, R_2{}^*)P_{Pub} + u_2{}^*\big)$

Analysis: It can be clearly stated that simulations of the random oracles $\mathcal{H}_i$ $(1 \leq i \leq 7)$ are perfect until $\mathcal{A}_1$ queries $\mathcal{H}_5$ with K. Let $Ask\mathcal{H}_5{}^*$ be the event that K is queried to $\mathcal{H}_5$. Let $Abort$ be the event that algorithm $\mathfrak{B}$ aborts the simulation before the guess phase. $\mathfrak{B}$ can abort the game in partial key extract, private key extract, re-encryption key extract or challenge phase. $\mathfrak{B}$ aborts the simulation for every partial key query $\mathbb{q}_{\boldsymbol{p_key}}$, private key

query $\mathbb{q}_{Sk}$ on input $ID \neq ID^*$ if $coin = 1$, re-encryption key query $\mathbb{q}_{Rk}$ on input (ID_i, ID_j) if $ID_i \neq ID^*$ and $coin = 1$ corresponding to ID_i. In the challenge phase, $\mathfrak{B}$ aborts if $coin = 0$ corresponding to the challenge identity ID^*. The probability for event $Abort$ is:

$$P[\neg Abort] \geq (1-\lambda)\lambda^{\mathbb{q}_{p_key}+\mathbb{q}_{Sk}+\mathbb{q}_{Rk}} \tag{4.23}$$

which is maximized at $\lambda^* = \frac{\mathbb{q}_{p_key}+\mathbb{q}_{Sk}+\mathbb{q}_{Rk}}{1+\mathbb{q}_{p_key}+\mathbb{q}_{Sk}+\mathbb{q}_{Rk}}$

Using λ^*, we get

$$P[\neg Abort] \geq \frac{1}{e(1+\mathbb{q}_{p_key}+\mathbb{q}_{Sk}+\mathbb{q}_{Rk})} \tag{4.24}$$

The simulation of re-encryption oracle is perfect as long as $\mathcal{A}_1$ can submit valid original ciphertext without querying $\mathcal{H}_4$ due to randomness of its output. Let $RErr$ denotes this event. Since $\mathcal{A}_1$ issues at most $\mathbb{q}_{re-enc}$ re-encryption queries, we have Eq. (4.25)

$$P[RErr] \leq \frac{\mathbb{q}_{re-enc}}{\mathbb{q}} \tag{4.25}$$

Similarly, strong decrypt-1 and strong decrypt-2 queries are perfect as long as $\mathcal{A}_1$ queries with valid ciphertext. Let $Valid$ denote the event that ciphertext is valid. Let $Ask\mathcal{H}_4{}^*$ be the event that $(m, \sigma, ID, u_1, u_2)$ has been queried to $\mathcal{H}_4$ as shown in Eq. (4.26)

$$P[Ask\mathcal{H}_4{}^* | \neg Ask\mathcal{H}_5{}^*] \leq \frac{\mathbb{q}_{\mathcal{H}_4}}{2^{l+n}} \tag{4.26}$$

$$P[Valid|\neg Ask\mathcal{H}_5{}^*]$$

$$= P[Valid \wedge Ask\mathcal{H}_4{}^*|\neg Ask\mathcal{H}_5{}^*] + P[Valid \wedge \neg Ask\mathcal{H}_4{}^*|\neg Ask\mathcal{H}_5{}^*]$$

$$P[Valid|\neg Ask\mathcal{H}_5{}^*] \leq P[Valid \wedge Ask\mathcal{H}_4{}^*|\neg Ask\mathcal{H}_5{}^*] + P[Valid \mid \neg Ask\mathcal{H}_4{}^* \wedge \neg Ask\mathcal{H}_5{}^*]$$

$$P[Valid|\neg Ask\mathcal{H}_5{}^*] \leq \frac{P[Valid \wedge Ask\mathcal{H}_4{}^* \wedge \neg Ask\mathcal{H}_5{}^*]}{\mathrm{P}[\neg Ask\mathcal{H}_5{}^*]} + P[Valid \mid \neg Ask\mathcal{H}_4{}^* \wedge \neg Ask\mathcal{H}_5{}^*]$$

$$P[Valid|\neg Ask\mathcal{H}_5{}^*] \leq \frac{P[Ask\mathcal{H}_4{}^*]}{\mathrm{P}[\neg Ask\mathcal{H}_5{}^*]} + P[Valid \mid \neg Ask\mathcal{H}_4{}^* \wedge \neg Ask\mathcal{H}_5{}^*]$$

$$P[Valid|\neg\, Ask\mathcal{H}_5{}^*] \leq \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1-{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{1}/_{\mathbb{q}}$$

Similarly,

$$P[Valid|\neg\, Ask\mathcal{H}_4{}^*] \leq \frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1-{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + {}^{1}/_{\mathbb{q}}$$

Hence,

$$P[Valid|\neg\, Ask\mathcal{H}_4{}^* \vee \neg\, Ask\mathcal{H}_5{}^*] \leq P[Valid|\neg\, Ask\mathcal{H}_4{}^*] + P[Valid|\neg\, Ask\mathcal{H}_5{}^*]$$

$$P[Valid|\neg\, Ask\mathcal{H}_4{}^* \vee \neg\, Ask\mathcal{H}_5{}^*] \leq \frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1-{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1-{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{2}/_{\mathbb{q}}$$

The event $DErr$ occurs if $(Valid|\neg\,(\, Ask\mathcal{H}_4{}^* \wedge Ask\mathcal{H}_5{}^*))$ happens atleast once during the simulation. $P[DErr]$ is calculated in Eq. (4.27)

$$P[DErr] \leq \mathbb{q}_{Dec}\left(\frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1-{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1-{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{2}/_{\mathbb{q}}\right) \tag{4.27}$$

Assume a polynomial time algorithm, which distinguishes between correctly formed re-encryption keys and incorrectly formed re-encryption keys. Let $RkErr$ be the event such that aforementioned algorithm exists and $P[RkErr] = \tau$. Consider an event $D = (Ask\mathcal{H}_5{}^* \vee (Ask\mathcal{H}_4{}^*|\neg\, Ask\mathcal{H}_5{}^*) \vee \text{RkErr} \vee RErr \vee DErr|\neg\, Abort)$. $\mathcal{A}_1$ does not gain any advantage in guessing greater than $\frac{1}{2}$ in guessing ß if event D does not happen due to randomness of output from random oracles. We have $P[ß^* = ß|\neg\, D] = \frac{1}{2}$. Hence by splitting $P[ß^* = ß]$, we have

$$P[ß^* = ß] = P[ß^* = ß|\neg\, D]P[\neg\, D] + P[ß^* = ß|D]P[D]$$

$$P[ß^* = ß] \leq \frac{1}{2}P[\neg\, D] + P[D] = \frac{1}{2}(1 + P[D])$$

$$\text{and } P[ß^* = ß] \geq P[ß^* = ß|\neg\, D]P[\neg\, D] = \frac{1}{2}(1 - P[D])$$

Let v be the advantage for adversary $\mathcal{A}_1$ given by Eq. (4.28)

$$v = \left| P[ß^* = ß] - \frac{1}{2} \right| \leq \frac{1}{2} P[D] \tag{4.28}$$

$$v = |2P[ß^* = ß] - 1|$$

$$v \leq \frac{1}{2} P[Ask\mathcal{H}_5{}^* \vee (Ask\mathcal{H}_4{}^* | \neg Ask\mathcal{H}_5{}^*) \vee RkErr \vee RErr \vee DErr | \neg Abort]$$

$$v \leq \frac{1}{2P[\neg Abort]} (P[Ask\mathcal{H}_5{}^*] + P[Ask\mathcal{H}_4{}^* | \neg Ask\mathcal{H}_5{}^*] + P[RkErr] + P[RErr] + P[DErr])$$

$$2vP[\neg Abort] - P[Ask\mathcal{H}_4{}^* | \neg Ask\mathcal{H}_5{}^*] - P[RkErr] - P[RErr] - P[DErr] \leq P[Ask\mathcal{H}_5{}^*]$$

If $Ask\mathcal{H}_5{}^*$ happens, then $\mathfrak{B}$ solves EC-CDH problem with probability $\frac{1}{\mathbb{q}_{\mathcal{H}_5}}$ and advantage v' is given by Eq. (4.29):

$$v' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_5}} P[Ask\mathcal{H}_5{}^*]$$

$$v' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_5}} \left(2v \frac{1}{e(1 + \mathbb{q}_{p_key} + \mathbb{q}_{Sk} + \mathbb{q}_{Rk})} - \frac{\mathbb{q}_{\mathcal{H}_4}}{2^{l+n}} - \tau - \frac{\mathbb{q}_{re-enc}}{\mathbb{q}} - \mathbb{q}_{Dec} \left(\frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1 - {}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1 - {}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{2}/_{\mathbb{q}} \right) \right) \tag{4.29}$$

Lemma 2: The proposed CL-PRE scheme is CL-PRE_IND-CCA_ORG secure against adversary $\mathcal{A}_2$ in random oracle model assuming underlying Elliptic Curve-Computational Diffie Hellman (EC-CDH) assumption is hard to solve.

Suppose $\mathcal{H}_i$ $(1 \leq i \leq 7)$ are random oracles and there exists a $\left(t, v, \mathbb{q}_{u_key}, \mathbb{q}_{Pk}, \mathbb{q}_{Sk}, \mathbb{q}_{Rk}, \mathbb{q}_{re-enc}, \mathbb{q}_{dec1}, \mathbb{q}_{dec1}\right) CL - PRE_IND - CCA_ORG$ adversary $\mathcal{A}_2$ with advantage v against the original first level ciphertext security $Game_2$ making at most $\mathbb{q}_{u_key}$ queries to $\mathbb{O}_{u_key}$, $\mathbb{q}_{Pk}$ queries to $\mathbb{O}_{Pk}$,$\mathbb{q}_{Sk}$ queries to $\mathbb{O}_{Sk}$,$\mathbb{q}_{Rk}$ queries to $\mathbb{O}_{Rk}$,$\mathbb{q}_{re-enc}$ queries to $\mathbb{O}_{re-enc}$,$\mathbb{q}_{dec1}$ queries to $\mathbb{O}_{dec1}$, $\mathbb{q}_{dec2}$ queries to $\mathbb{O}_{dec2}$

and $\mathbb{q}_{\mathcal{H}_i}$queries to random oracles $\mathcal{H}_i$ $(1 \leq i \leq 7)$, then there exists an algorithm $\mathfrak{B}$ which solves EC-CDH problem with advantage v':

$$v' \geq \frac{1}{\mathbb{q}_{\mathcal{H}_5}}\left(2v\frac{1}{e(1+\mathbb{q}_{Sk}+\mathbb{q}_{Rk})} - \frac{\mathbb{q}_{\mathcal{H}_4}}{2^{l+n}} - \tau - \frac{\mathbb{q}_{re-enc}}{\mathbb{q}} - \mathbb{q}_{Dec}\left(\frac{{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}}{1-{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}} + \frac{{}^{\mathbb{q}_{\mathcal{H}_4}}/_{2^{l+n}}}{1-{}^{\mathbb{q}_{\mathcal{H}_5}}/_{\mathbb{q}}} + {}^{2}/_{\mathbb{q}}\right)\right)$$

within time

$$t' \leq t + \left(\mathbb{q}_{\mathcal{H}_1} + \mathbb{q}_{\mathcal{H}_2} + \mathbb{q}_{\mathcal{H}_3} + \mathbb{q}_{\mathcal{H}_4} + \mathbb{q}_{\mathcal{H}_5} + \mathbb{q}_{\mathcal{H}_6} + \mathbb{q}_{\mathcal{H}_7} + \mathbb{q}_{u_key} + \mathbb{q}_{Pk} + \mathbb{q}_{Sk} + \mathbb{q}_{Rk} + \mathbb{q}_{re-enc} + \mathbb{q}_{dec1} + \mathbb{q}_{dec2}\right)O(1) + (2\mathbb{q}_{u_key} + 2\mathbb{q}_{Pk} + 4\mathbb{q}_{Rk} + \mathbb{q}_{re-enc} + 2\mathbb{q}_{dec1} + 5\mathbb{q}_{dec2})t_{PM}$$

where t_{PM} denotes the point multiplication time in group G, τ denotes the probability $\mathcal{A}_2$ can distinguish between correctly formed re-encryption keys and incorrectly formed re-encryption keys and v denotes the advantage adversary may have over EUF-CMA security of Schnorr signature within time t.

The security proof against adversary $\mathcal{A}_2$ is analogous to the aforementioned proof. Adversary $\mathcal{A}_2$ knows master secret key and cannot replace public keys of identities. It can easily compute partial key of identity on its own so there is no need to query partial key extract oracle explicitly.

4.7 Computational and Performance Analysis

In this section, the computational efficiency of proposed CL-PRE scheme has been compared with Sur et al.'s scheme [163], Xu et al.'s scheme [92], Guo et al.'s scheme [165], Yang et al.'s scheme [166], Qin et al.'s scheme [33] and Srinivasan et al.'s scheme [168]. A standard cryptographic library, MIRACL [34], has been used to compute running time of various CL-PRE schemes. The hardware platform involves i5 2.3GHz processor with 4 GB RAM. In order to achieve 1,024-bit RSA security level for bilinear pairing based

schemes, Tate pairing defined over the elliptic curve $E/F_p: y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime and p is a 512-bit prime has been used. To achieve an equivalent security in the ECC-based scheme, the verifiable random elliptic curve secp160r1 [141] over F_p where $p = 2^{160} - 2^{31} - 1$ has been employed. However, vendors can choose 2048-bit or 3072-bit security level, corresponding curves and other parameters while implementing on the real world. The schemes [163, 92, 165] are pairing based and the cost of pairing operation is approximately 10 folds more than scalar point multiplications over elliptic curves [127]. The schemes [163, 92, 165, 166] are based on the weak security model, but the proposed scheme is provable secure under the strong security model.

Table 4.1 compares the computational efficiency of the proposed CL-PRE scheme with existing unidirectional CL-PRE schemes in the literature [163, 92, 165, 166, 33, 168]. Here, t_{bp} denotes bilinear pairing operation, t_e denotes exponentiation operation and t_{pm} denotes ECC-based point multiplications. The computation cost of map-to-map point hash operations, point additions and general hash operations have been omitted because of their negligible computation time in comparison with other operations. It has been assumed that public key verification and ciphertext verification are done as a pre-processing step in all the schemes. To evaluate the total computation time, the method followed in [127] has been used. To the best of our knowledge, there exist 1 attack-free CL-PRE scheme [165] in the domain of CL-PRE but it is based on expensive bilinear pairing operation. According to Table 4.1 and Figure 4.2 , the total computation time of the proposed scheme is 22.87% of Sur et al.'s scheme [163], 36.68% of Xu et al.'s scheme [92], 19.28% of Guo et al.'s scheme [165], 59.83% of Yang et al.'s CL-PRE1 and 38.29% of CL-PRE2 schemes [166], 50.38% of Qin et al.'s scheme [33] and 34.19% of Srinivasan et al.'s scheme [168].

It is clear from Table 4.1 that the proposed scheme is computationally more efficient than all the existing CL-PRE schemes. The communication overhead of the proposed CL-PRE scheme is also less in comparison to existing schemes in terms of re-encryption key size [163, 92, 165, 168], first level ciphertext size [92, 166, 168] and second level ciphertext size [163, 92, 165, 168].

Table 4.1: Comparison of proposed scheme with other schemes

Schemes → Algorithm ↓	Sur et al. [163]	Xu et al. [92]	Guo et al. [165]	Yang et al. [166] CLPRE1	Yang et al. [166] CLPRE2	Qin et al. [33]	Srinivasan et al. [168]	Proposed CL-PRE
Setup	$t_e + 2t_{bp}$	t_e	$3t_e + t_{bp}$	t_e	t_e	t_e	t_e	t_{pm}
Extract partial key	t_e	t_e	$2t_e$	t_e	t_e	$2t_e$	$3t_e$	$3t_{pm}$
Generate public key	$2t_e$	t_e	t_e	$2t_e$	$2t_e$	t_e	$4t_e$	$2t_{pm}$
Generate Re-encryption key	$5t_e$	$2t_e$	$5t_e$	$3t_e$	$3t_e$	$3t_e$	$5t_e$	$4t_{pm}$
Encrypt	$6t_e$	$3t_e + t_{bp}$	$3t_e + 2t_{bp}$	$4t_e$	$5t_e$	$4t_e$	$8t_e$	$5t_{pm}$
Proxy re-encrypt	$2t_{bp}$	$3t_e + t_{bp}$	$5t_e + t_{bp}$	t_e	$3t_e$	t_e	t_e	t_{pm}
Decrypt1	$4t_e + 2t_{bp}$	t_{bp}	$3t_e + t_{bp}$	$3t_e$	$6t_e$	$2t_e$	$2t_e$	$2t_{pm}$
Decrypt2	$4t_e + t_{bp}$	t_{bp}	$5t_e + t_{bp}$	t_e	$4t_e$	$5t_e$	$4t_e$	$5t_{pm}$
First level ciphertext $\lvert c_\sigma \rvert$	$\lvert G_1 \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$2\lvert G \rvert + \lvert m \rvert + \lvert G_T \rvert$	$\lvert G_T \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$\lvert G \rvert + \lvert m \rvert$	$2\lvert G \rvert + \lvert Z_q \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$\lvert G \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$2\lvert G \rvert + \lvert Z_q \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$\lvert G \rvert + \lvert m \rvert + \lvert \sigma \rvert$
Second level ciphertext $\lvert c_r \rvert$	$\lvert G_1 \rvert + 2\lvert G_2 \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$2\lvert G \rvert + 2\lvert m \rvert + 2\lvert G_T \rvert$	$3\lvert G \rvert + +\lvert m \rvert + \lvert \sigma \rvert$	$3\lvert G \rvert + \lvert m \rvert$	$\lvert G \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$\lvert G \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$2\lvert G \rvert + 2\lvert m \rvert + 2\lvert \sigma \rvert$	$\lvert G \rvert + \lvert m \rvert + \lvert \sigma \rvert$
$\lvert Rk_{\sigma \to r} \rvert$	$3\lvert G_1 \rvert$	$2\lvert G_1 \rvert + 2\lvert G_2 \rvert$	$3\lvert G \rvert$	$\lvert Z_q \rvert$	$\lvert Z_q \rvert$	$\lvert Z_q \rvert$	$\lvert G \rvert + \lvert Z_q \rvert + \lvert m \rvert + \lvert \sigma \rvert$	$\lvert Z_q \rvert$
Total computation time (ms)	222.3	138.57	263.61	84.96	132.75	100.89	148.68	50.83
Pairing free	X	X	X	√	√	√	√	√
Attacked	√	√	X	√	√	√	√	X
Security	CCA	CPA	RCCA	CPA	CCA	CCA	CCA	CCA
Assumption	p-BDHI	DBDH	p-DBDH	CDH	CDH	CDH	CDH	EC-CDH
Security Model	Weak	Weak	Strong	Weak	Weak	Strong	Strong	Strong

$\lvert G \rvert, \lvert G_1 \rvert, \lvert G_2 \rvert, \lvert G_T \rvert, \lvert Z_q \rvert$ denote the bit-length of an element in $G, G_1, G_2 G_T$, Z_q $\lvert m \rvert, \lvert \sigma \rvert$ denote length of plaintext and random-bit string. p-BDHI denotes p-Bilinear Diffie-Hellman Inversion, DBDH denotes Decisional Bilinear Diffie- Hellman, CDH denotes Computational Diffie–Hellman, EC-CDH denotes Elliptic Curve Computational Diffie-Hellman

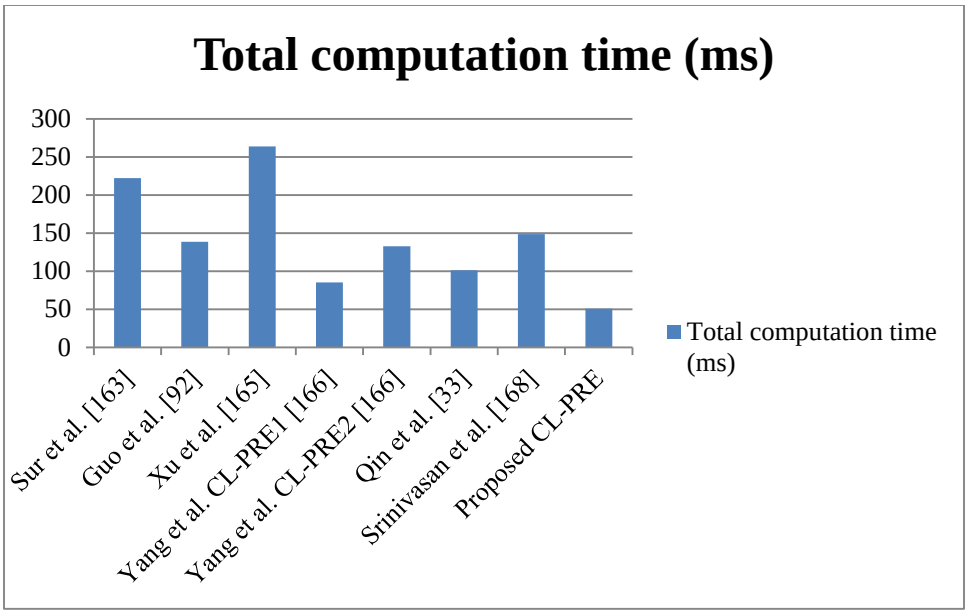

Figure 4.2: Comparison of existing schemes with proposed CL-PRE scheme in terms of computation time

4.8 Summary

Secure mPHRs storage and sharing is a critical issue while leveraging cloud resources for storing encrypted data without revealing underlying message contents to cloud resident proxy server. The most severe threat that looms over cryptography is the key escrow problem. The CL-PRE primitive holds great promise of overcoming these issues and providing secure data sharing in the cloud. An efficient and lightweight pairing free single-hop unidirectional CL-PRE scheme based on ECC for secure sharing of mPHRs in public cloud has been proposed. The proposed CL-PRE scheme is provably secure in the random oracle model under the EC-CDH assumption. The recently proposed CL-PRE scheme by Qin et al. [33] has been proved insecure under our security model. The proposed scheme is computationally efficient than existing schemes, which can be in tune with modern resource-constrained mobile devices to be used in the age of IoT in e-healthcare industry.

Chapter 5

Incremental Certificateless Proxy Re-encryption

5.1 Introduction

Due to the sensitive nature of EHRs constituents, mobile client should encrypt the e-healthcare data before outsourcing to cloud to prevent its exposure to intruders and CSPs. Any healthcare confidentiality and integrity protocol should target the EHR security, regulatory, and operational requirements to stipulate the migration of medical records to the cloud. The digital universe is expanding just like the physical universe and may embrace nearly as many digital bits as there are stars in the universe by 2020 and will reach 44 zettabytes [172]. Internet of Things (IoT) Tsunami has created a mark for itself in the digital universe. The digital universe is expected to grow from a mere 3.8% in 2016 to 10% by 2020 with the inception of IoT and related technologies. The engulfing waves of IoT comprise tens of billions of sensors, billions of intelligent systems and millions of applications flooding our memory system due to round-the-clock digital data acquisition and generation. According to Gartner, Inc., there will be nearly 26 billion IoT devices by 2020, which will demand smart industry solutions generating revenue in excess of $300 billion to IT vendors and service providers. According to the Cisco recent forecast, annual global IP traffic passed the zettabyte threshold by the end of 2016 and will reach 2.3 zettabytes per year by 2020 [173]. Global mobile data traffic will increase sevenfold between 2016 and 2021. Mobile data traffic will grow at a compound annual growth rate (CAGR) of 47% from 2016 to 2021, reaching 49 exabytes per month by 2021 [174]. Thus, the scale of data being processed every day by various cloud services, sensor networks, distributed storage systems and digital media service demands for novel secure and

---

The major findings of this chapter have been peer reviewed and communicated for publication in:

- *Tarunpreet Bhatia, A.K. Verma, "Towards a Secure Incremental Proxy Re-encryption for e-Healthcare Data Sharing in Mobile Cloud Computing" (Communicated)*

efficient solutions based on the incremental cryptography paradigm. The advent of IoT has given place for better monitoring and surveillance of patients' health via the acquisition of their body parameters through small wearable and implantable sensors. The data collected are communicated over the short range wireless communication devices. The data captured by the sensor nodes need to be stored for further analysis. With the wider adoption of Wireless Body Area Networks (WBANs) and reduction in the cost of the sensors due to improvement in sensor technology, the data being generated and henceforth stored are expected to increase exponentially. Thus, WBAN finds itself marred in the security and privacy concerns regarding the storage of EHRs either within the domain of WBAN or outsourcing to third-party CSPs.

Proxy re-encyption (PRE) is a cryptographic scheme that allows Bob to delegate his decryption rights to another user Alice. In a PRE scheme, a semi-honest proxy agent of Bob re-encrypts the ciphertext encrypted under Bob's public key on behalf of Bob into ciphertext encrypted under Alice's public key without learning anything about the message and private key of Bob. The PRE schemes are useful in the scenarios where data is to be shared with multiple authorized users over the cloud. Nevertheless, the integrity of EHRs should also be preserved, and usually, such EHRs should be signed by a trusted party. In any case, data hashing is inevitable, and as EHR is being updated, so the hash value needs to be re-computed to reflect the corresponding updations. The very context of medical data confines the modifications of EHR's to appendments of new data from sensors and minimal updations. An entire recalculation of the hash value on the updated EHR's not only sounds practically irrational but also is ludicrous if the size of the EHRs' is taken into perspective. Such modifications provide for an opportune implementation of the incremental approach of data hashing. Incremental cryptography can minimize the overhead associated with re-calculation of the hash value from the scratch by considering only modifications (insertion, updation and deletion) to a dataset. PRE is a cryptographic scheme for delegation of decryption rights. In a PRE scheme, a semi-honest proxy agent of Bob re-encrypts the ciphertext, on the message intended for Alice, on behalf of Bob, without learning anything about the message. The PRE schemes are useful in the scenarios where data are desired to be shared with the authorized users over the cloud. Nevertheless, the integrity of EHRs should also be preserved, and usually, such EHRs should be signed by a trusted party. In

any case, data hashing is inevitable, and as EHR is being updated, so the hash value needs to be re-computed to reflect the corresponding updations. The very context of medical data confines the modifications of EHR's to appendments of new data from sensors and minimal updations. An entire recalculation of the hash value of the updated EHR's not only sounds practically irrational, but also is ludicrous if the size of the EHRs' is taken into perspective. Such modifications provide for an opportune implementation of the incremental approach of data hashing. Incremental cryptography can minimize the overhead associated with re-calculation of the hash value from the scratch by considering only modifications (insertion, updation and deletion) to a dataset.

5.2 Related Work

Relying on the traditional integrity mechanisms such as MACs and digital signatures in their classical forms would not suit the constraints of healthcare services in the evolving cloud computing infrastructure. The existing security solutions for cloud environment [42, 175, 43] neglected resource-constrained nature of the mobile devices. Jia et al. [107] and Zhou et al. [83] proposed data security schemes based on the notion of proxy re-encryption for mobile cloud environment by offloading the data-intensive operations from a mobile device to the cloud without exposing the data content and security keys to anyone. Yang et al. [100] proposed modified public verifiable data possession (PDP) scheme which involved expensive bilinear mapping and Merkle Hash Trees for ensuring integrity.

Bellare et al. [110, 111] designed cryptographic algorithms founded on the notion of incremental cryptography, which updates output efficiently as per the underlying input modifications. A file visualized as a sequence of blocks would undergo re-computations for generating the ciphertext, only for those blocks that underwent any alterations excluding the ones that remain unaltered [176, 177]. If we apply the incremental cryptographic algorithm on a medical record and afterwards this record gets modified, it is feasible to update the result of the algorithm by merely using the modified record blocks rather than re-computing the algorithm on the entire record with incremental cryptography. This property contributes to major performance and energy efficiencies in e-healthcare setting due to the large size of medical documents and the frequent update mechanisms that

are performed on them. Alongside the confidentiality aspect of EHRs, it is imperative to ensure their integrity which would prevent their accidental or intentional modification.

The incremental hash functions have failed to make a mark in the industry due to following loopholes. Firstly, the known incremental hash functions [110, 111] need to perform expensive modular operations over large prime integers to attain a certain level of security (for example 2^{128} or 2^{256}) which makes their performance lag behind those of the simple cryptographic hash functions. Secondly, the security level of the incremental hash functions is disproportional to the size of the calculated hash value by several thousand bits, unlike ordinary cryptographic hash functions such as SHA-1 [178], SHA-2 [179], SHA-3 [180], where claimed bit-security level of the hash function corresponds to the size of the calculated hash value. Incremental hashing can be achieved using merkle trees [181] as discussed in [182] which necessitate the storage of all the intermediate nodes' hash values.

In 2005, the attacks on SHA-1 performed by cryptanalysts proved the inappropriateness of the algorithm for future use [183], and since 2010 many organizations have replaced it by SHA-2 or SHA-3. This was followed by subsequent announcements by Apple, Microsoft, Google, and Mozilla declaring SHA-1 SSL certificates unacceptable from 2017 onwards. CWI and Google successfully performed a collision attack against SHA-1 on February 23, 2017 [184], publishing two unrelated PDF files which give the same SHA-1 hash value as an output.

Mihajloska et al. [185] proposed two incremental hash functions iSHAKE128 and iSHAKE256 for zettabyte era based on extendable-output functions (XOF's). iSHAKE128 and iSHAKE256 have the flexibility of generating an output of any desired length. Technically, XOF can be employed as a hash function by choosing fixed output length but the two outputs of a common message are closely related for two different outputs lengths, i.e. the hash output of longer length is an extension of shorter length, which is an undesirable property of hash functions.

5.3 Proposed CL-iPRE Scheme

A lightweight CL-iPRE has been proposed for secure sharing of EHRs in public cloud. The proposed scheme relies on the proxy resident cloud for re-encryption. It has been assumed that the public cloud is fully untrusted and cloud-resident proxy server is semi-trusted, i.e. it follows the protocol but can gather information to deduce private information. The certificateless cryptography solves key escrow problem in traditional PKC. Any incremental cryptographic algorithm must run faster than re-computing the transformation from scratch. For example, if b denotes the block size then a single modification to a block should take $poly(b)$ time regardless of the number of blocks or the total size of entire EHR. It has been assumed that proxy does not collude with sender or receiver to obtain their secret keys. The architecture of proposed CL-iPRE scheme is shown in Figure 5.1. The proposed CL-iPRE scheme is as follows:

- **System setup:** This algorithm is run by KGC. It chooses a security parameter k and an elliptic curve E/F_p over a prime finite field F_p. Let G be the cyclic subgroup of the elliptic curve group with P as a generator of order q. It randomly selects master secret key s $\in Z_q^*$. It further computes master public key $P_{Pub} = sP$ and publishes a list of public parameters $\delta = \{E, F_p, G, P, P_{Pub}\}$ for encryption, decryption and proxy re-encryption.

- **Partial key generation:** A deterministic algorithm executed by KGC, which takes public parameters δ, cloud user identity ID_U, master secret key s, master public key P_{Pub} as inputs. It randomly chooses $r_U \in Z_q^*$ and computes $W_U = r_U P$ and $h_U = \mathcal{H}_1(ID_U, W_U)$. It generates $S_U = (r_U + h_U s)$ as $U's$ partial private key corresponding to user's identity ID_U and transmits S_U as partial private key to user U through a secure channel and W_U as partial public key over an insecure channel.

- **Set secret value:** An algorithm run by every cloud user takes public parameters δ, user's identity ID_U as inputs and outputs a secret value $k_U \in Z_q^*$.

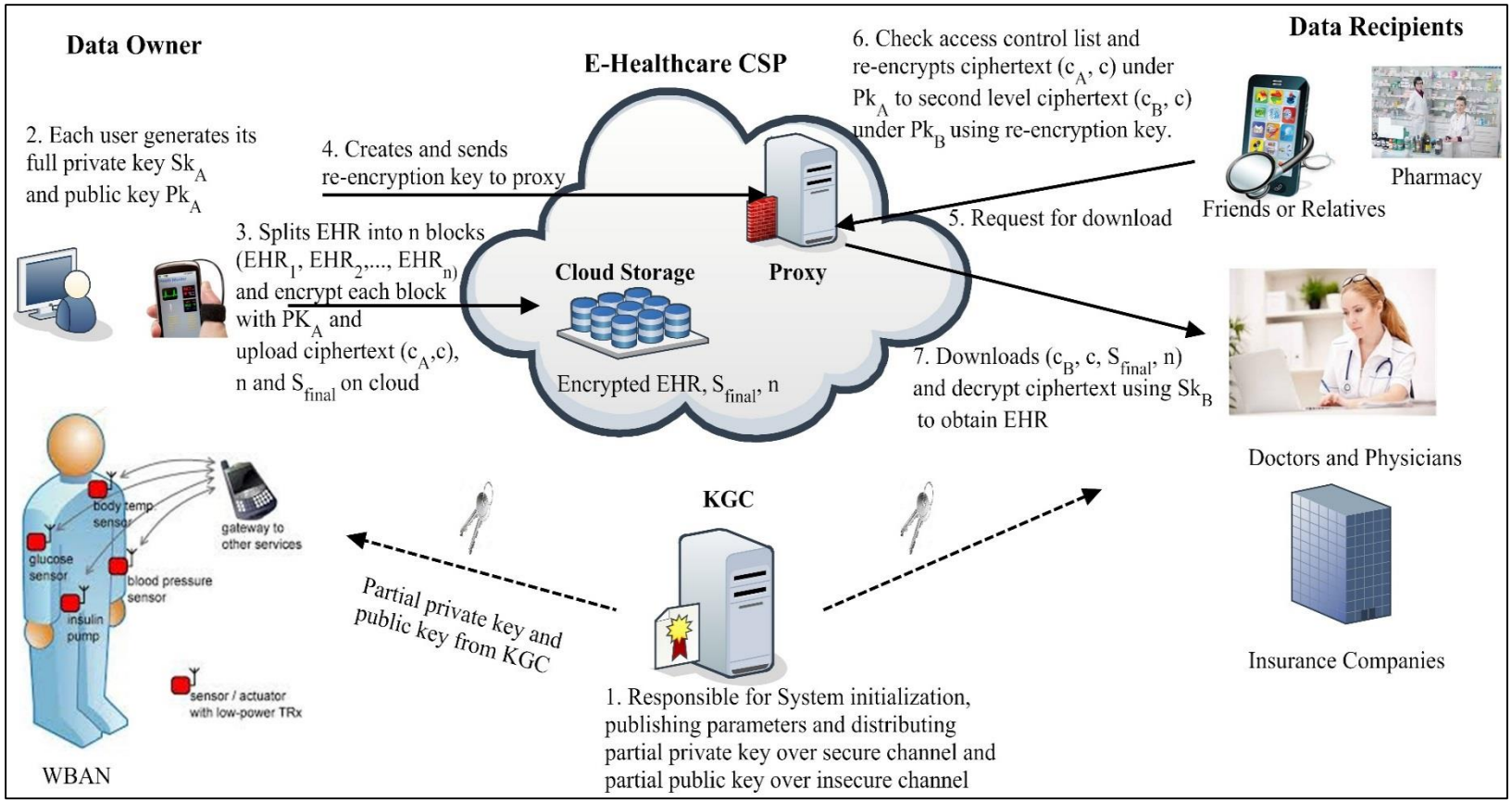

Figure 5.1: Architecture of proposed CL-iPRE scheme for secure sharing of EHRs in MCC

- **Private key generation**: This algorithm is executed by cloud user U with identity ID_U, which accepts public parameters δ, partial private key S_U and secret value $k_U \in Z_q^*$ as inputs and generates a full private key $Sk_U = (S_U - k_U)$ for user U.

- **Public key generation**: This algorithm is executed by cloud user U with identity ID_U. It computes $V_U = k_U P$ and a public key $Pk_U = Sk_U P$ for user U. Any user who wishes to use this public key must first validate the received public key using Eq. (5.1):

$$Pk_U = W_U + h_U P_{Pub} - V_U \tag{5.1}$$

where $h_U = \mathcal{H}_1(ID_U, W_U)$

- **Re-encryption key generation**: This algorithm is executed by data owner A to generate re-encryption key $Rk_{A\to B} = {Sk_B}/{Sk_A}$ using secure two-party integer division algorithm [187] which is forwarded to cloud resident proxy server $\mathcal{P}$ without leaking any information about Sk_B or Sk_A to data owner, data recipient or proxy server. This algorithm [187] allows two or more mutually mistrusting parties to evaluate a function on private data without revealing additional information to the parties involved.

- **Encryption:** The mobile data owner A wants to upload EHR on the cloud. The owner A splits EHR into n blocks $EHR_1, EHR_2, \ldots\ldots, EHR_n$ as shown in Eq. (5.2) where each block has fixed size of b bits except possibly the last.

$$EHR = ||_{i=1}^{n}(EHR_i) \tag{5.2}$$

$$b_j = \left\lfloor \frac{x}{n} \right\rfloor \text{ where } 1 \le j \le n-1 \tag{5.3}$$

$$b_n = x - \left(\left\lfloor \frac{x}{n} \right\rfloor * (n-1)\right) \tag{5.4}$$

Where EHR_i denotes i^{th} block of EHR, x denotes total size of $EHRs$, b_j denotes size of j^{th} block of EHR and $\left\lfloor \frac{x}{n} \right\rfloor$ denotes the mathematical floor function to remove fractional part. In order to provide confidentiality, data owner A encrypts each block EHR_i before storing on the cloud. The data owner A generates a random

number $\alpha \in Z_q^*$ and encrypts the message using its public key Pk_A using Eq. (5.5) and (5.6).

$$c_A = \alpha . Pk_A, c_i = EHR_i + \alpha . P \quad (5.5)$$

$$c = ||_{i=1}^{n}(c_i) \quad (5.6)$$

where $1 \leq i \leq n$

Similarly, hash for each block of EHR is generated using recently proposed SHA-3 algorithm. The hash function has to map b bits of each block to l bits i.e. $h: \{0,1\}^b \rightarrow \{0,1\}^l$ where l is a multiple of 64 bits. The individual hash values are then combined using word-wise addition in the commutative group $\left((Z_{2^{64}})^{l/64}, \boxplus_{64}\right)$ to generate final hash value for verifying integrity of stored $EHRs$. The group combining operator $\boxplus_{64}$ represents 64-bit word-wise addition of $l/64$ words and $\boxminus_{64}$ represents the inverse of addition operation i.e. word-wise subtraction of $l/64$ words. The final hash value H_{final} (Eq. (5.8) is digitally signed to obtain S_{final} which provides authentication and non-repudiation.

$$H_i = \mathcal{H}_{SHA-3}(EHR_i) \text{ where } 1 \leq i \leq n \quad (5.7)$$

$$H_{final} = H_1 \boxplus_{64} H_2 \boxplus_{64} \ldots . \boxplus_{64} H_n \quad (5.8)$$

The encrypted EHRs (c_A, c), the number of blocks n and S_{final} are uploaded on the cloud for storage. The data owner needs to save filename of $EHRs$ and number of blocks n for each EHR file in its local memory of mobile device.

- **Proxy re-encryption:** The data recipient B who wants to download data uploaded by user A requests semi-trusted proxy which transforms first level ciphertext $c_A \in \mathcal{C}_1$ received from data owner A into second level ciphertext $c_B \in \mathcal{C}_2$ for data recipient B. Proxy sends a re-encrypted message (c_B, c) to user B for decryption and integrity verification using Eq. (5.9), (5.10) and (5.11).

$$c_B = c_A Rk_{A\to B} = \alpha . Sk_A . P\left(\frac{Sk_B}{Sk_A}\right) = \alpha Sk_B P \tag{5.9}$$

$$c_i = EHR_i + \alpha . P \tag{5.10}$$

$$c = ||_{i=1}^{n}(c_i) \tag{5.11}$$

where $1 \leq i \leq n$

- **Decrypt:** This algorithm is executed by cloud user $U \in \{A, B\}$ that takes ciphertext c_U corresponding to user U as an input and generates corresponding plaintext message or an error symbol Υ if c_U is invalid. The data owner A decrypts the message using c_A and its secret key Sk_A using Eq. (5.12):

$$EHR_i = c_i - \left(\frac{1}{Sk_A}\right) c_A \text{ where } 1 \leq i \leq n \tag{5.12}$$

The data recipient B decrypts the message using c_B and its secret key Sk_B using Eq. (5.13):

$$EHR_i = c_i - \left(\frac{1}{Sk_B}\right) c_B \text{ where } 1 \leq i \leq n \tag{5.13}$$

The mobile user concatenates individual blocks of the file to get original EHR using Eq. (5.14) and verifies integrity of downloaded file using Eq. (5.16):

$$EHR = ||_{i=1}^{n}(EHR_i) \text{ where } 1 \leq i \leq n \tag{5.14}$$

$$H_i' = \mathcal{H}_{SHA-3}(EHR_i) \text{ where } 1 \leq i \leq n \tag{5.15}$$

$$H_{final}' = H_1 \boxplus_{64} H_2 \boxplus_{64} \ldots\ldots \boxplus_{64} H_n \tag{5.16}$$

The data recipient B uses $A's$ public key to decrypt (de-sign) the digital signature S_{final} and obtains H_{final}. It verifies integrity of EHR received by comparing H_{final} with calculated H_{final}'. If both values match, then integrity is confirmed.

5.4 Block Modification Operations

The block modification operations include block(s) updation, block(s) insertion and block(s) deletion. The overhead involved in dividing $EHRs$ and calculating hash values of individual blocks increase resource utilization on mobile device as compared to proxy re-

encryption schemes, however incremental concept improves resource utilization on mobile device while performing block modification operations on uploaded EHRs as hash values need not to be re-calculated from scratch for each block modification query.

5.4.1 Incremental block updation operations

Once the hash function is applied on individual blocks and S_{final} is stored on the cloud, there is no need to repeat entire procedure on a mobile device while updating the block(s). An incremental update operation can be applied that performs wordwise-subtraction $\boxminus_{64}$ of old hash values of updated blocks and wordwise-addition $\boxplus_{64}$ of updated hash values of modified blocks. Thus, only two hash operations are required for each modified block. The storage and communication overhead is also reduced because CSP needs to store signature value on cloud and mobile device needs to download l-bit hash value for re-computing hash value.

Let data owner A wants to modify q block(s) represented by EHR_{update} in the uploaded record starting from location index idx. The data owner A needs to download encrypted EHR i.e. (c_A, c) and S_{final} corresponding to record EHR from cloud storage on its device. c_A is used for encrypting updated blocks as Eq. (5.17):

$$c_{update} = ||_{j=1}^{q} \left(EHR_{update_j} + \left(\frac{1}{Sk_A} \right) c_A \right) \text{ where } 1 \leq j \leq q \qquad (5.17)$$

The data owner A then calculates updated hash value H_{new} using Eq. (5.19) and sends an update request to CSP along with c_{update}, location index idx, number of modified blocks q and H_{new}. The updated hash value H_{new} is digitally signed to obtain S_{new} using $A's$ private key.

$$H_{update} = H_{idx} \boxplus_{64} H_{idx+1} \boxplus_{64} \ldots\ldots \boxplus_{64} H_{idx+q} \qquad (5.18)$$

$$H_{new} = H_{final} \boxminus_{64} H_{idx} \boxminus_{64} H_{idx+1} \boxminus_{64} \ldots\ldots \boxminus_{64} H_{idx+q} \boxplus_{64} H_{update} \qquad (5.19)$$

CSP updates encrypted EHR using Eq. (5.20) and stores S_{new} obtained by signing H_{new} corresponding to stored EHR.

$$c = ||_{i=1}^{idx}(c_i)||c_{update}||_{k=q+1}^{n}(c_k) \tag{5.20}$$

5.4.2 Incremental block insertion operations

The mobile data owner A who wishes to insert q new blocks represented by EHR_{insert} after location index idx in previously uploaded record EHR, sends a download request to the cloud server. The data owner A downloads (c_A, c) and S_{final} corresponding to record EHR. c_A is used for encrypting new blocks using Eq. (5.21):

$$c_{insert} = ||_{j=1}^{q}\left(EHR_{insert_i} + \left(\frac{1}{Sk_A}\right)c_A\right) \text{ where } 1 \leq j \leq q \tag{5.21}$$

The data owner A downloads S_{final} from cloud corresponding to stored EHR and apply $\boxplus_{64}$ to obtain the hash values of newly inserted blocks. Thereafter, it combines it to H_{insert} to obtain the new hash value denoted by H_{new} using Eq. (5.23). The data owner then sends an insert request to CSP including S_{new} obtained by signing H_{new}, location index idx after which blocks are to be inserted and newly encrypted blocks c_{insert}.

$$H_{insert} = H_{insert_1} \boxplus_{64} H_{insert_2} \boxplus_{64} \ldots\ldots \boxplus_{64} H_{insert_q} \tag{5.22}$$

$$H_{new} = H_{final} \boxplus_{64} H_{insert} \tag{5.23}$$

CSP insert(s) the block(s) into corresponding encrypted EHR and updates ciphertext using Eq. (5.24):

$$c = ||_{i=1}^{idx}(c_i)||c_{insert}||_{k=idx+1}^{n}(c_k) \tag{5.24}$$

S_{final} value is updated to S_{new} and number of blocks $b = b + q$ for corresponding EHR. The block insertion can be done at multiple locations also.

5.4.3 Incremental block deletion operations

In order to delete q blocks at some random location index idx from uploaded EHR on cloud, data owner A downloads S_{final} of corresponding EHR from the cloud storage. The data owner re-calculates hash using Eq. (5.25):

$$H_{new} = H_{final} \boxminus_{64} H_{idx} \boxminus_{64} H_{idx+1} \boxminus_{64} \ldots\ldots \boxminus_{64} H_{idx+q} \quad (5.25)$$

where $H_{idx}, H_{idx+1}, \ldots\ldots, H_q$ is corresponding hash values of deleted block(s).

The data owner sends a delete request to CSP along with re-calculated value S_{new}, location index idx, number of blocks to be deleted q. CSP deletes the corresponding block(s) using Eq. (5.26) in stored encrypted EHR and stores updated signature corresponding to stored EHR.

$$c = ||_{i=1}^{idx-1}(c_i)||_{j=idx+q+1}^{n}(c_j) \quad (5.26)$$

5.5 Results and Comparative analysis

The proposed scheme is evaluated in terms of turnaround time and energy consumption of the mobile device for block updation operations in comparison to Proxy re-encryption scheme (PReS) and I-PReS [114]. Khan et al. [114] assumed proxy as a fully trusted entity possessing secret keys which can decrypt messages of any entity suffering from key escrow problem. It involves expensive bilinear pairing operations and downloading of intermediate MAC values of all blocks from the cloud by mobile user for block modification operations, thereby increasing communication, computation and storage overhead on a mobile device. Android SDK is used for developing mobile cloud client application deployed on a Nexus smartphone with 1 GB RAM and 2.53 GHz processor. A web instance is hosted on GAE which provides cloud-based infrastructure for testing Android apps. A standard cryptographic library, MIRACL [34] has been used to compute running time of CL-iPRE scheme. In order to achieve 1,024-bit RSA level security for bilinear pairing based schemes, Type A pairing defined over the elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime and p is a 512-bit prime has been used. To achieve an equivalent security in ECC-based scheme, the verifiably random elliptic curve secp160r1 [141] over F_p where $p = 2^{160} - 2^{31} - 1$ has been employed. SHA3-512 has been used in proposed CL-iPRE scheme as hash function as it is more resistant to pre-imaging and collision attacks than its predecessors. SHA-2 was used in I-PReS. The turnaround time and energy consumption for files of different sizes as shown in Table 5.1 with 8 blocks in each file has been evaluated. The readings

have been taken while updating 1, 2 and 4 blocks represented as CL-iPRE(1), CL-iPRE(2), and CL-iPRE(4) respectively. Each experiment has been repeated five times and average readings are noted to plot graphs.

$$Total\ turnaround\ time = t_{read} + t_{encrypt} + t_{upload}$$

where t_{read} refers to file reading time, $t_{encrypt}$ represents a block updating time and t_{upload} represents block uploading time.

$$Energy\ consumed = E_{read} + E_{encrypt} + E_{upload}$$

where E_{read} refers to energy consumed in file reading, $E_{encrypt}$ represents the energy consumed in block updating and E_{upload} represents the energy consumed in file uploading.

Table 5.1: Data set used for file modification operations

File size in bytes	Total files	Block size in bytes
51200	50	6400
102400	50	12800
153600	50	19200
204800	50	25600

Figure 5.2 and 5.3 shows the variation in turnaround time (in seconds) and energy consumption (in %) with different file sizes. As depicted in Figure 5.2, average turnaround time (with files of varying sizes) in proposed CL-iPRE scheme is 26.49% of I-PReS and 4.1% of PReS with 1 modified block, 26.52% of I-PReS and 8.4% of PReS with 2 modified blocks, and 33.96% of I-PReS and 20.43% of PReS with 4 modified blocks. The average energy consumption (with files of varying sizes) in proposed CL-iPRE scheme is 67.22% of I-PReS and 11.03% of PReS with 1 modified block, 71.37% of I-PReS and 23.78% of PReS with 2 modified blocks, and 78.63% of I-PReS and 48.17% of PReS with 4 modified blocks as shown in Figure 5.3.

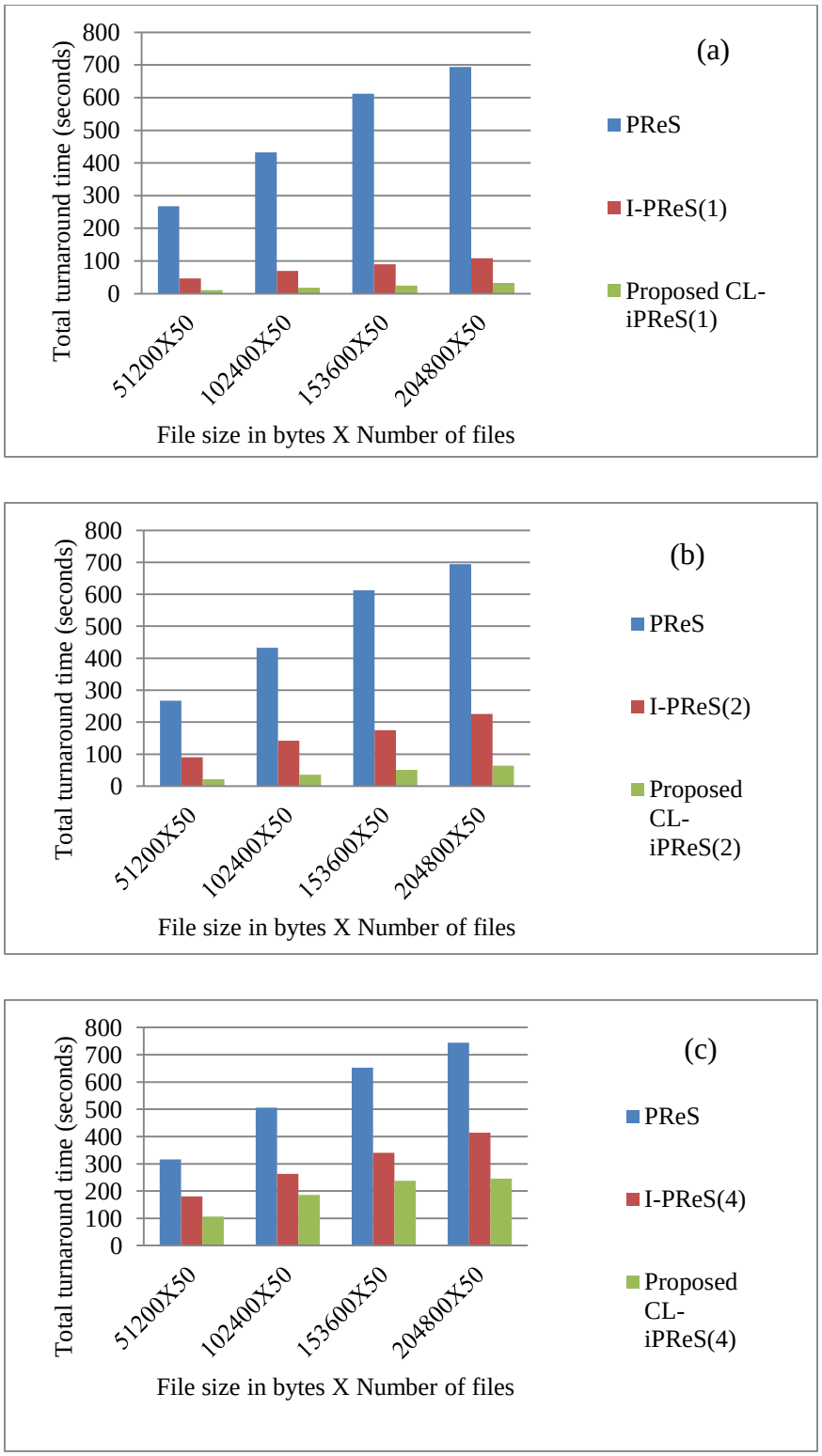

Figure 5.2: Turnaround time while block updating operations on dataset (a) Updating 1 block (b) Updating 2 blocks (c) Updating 4 blocks

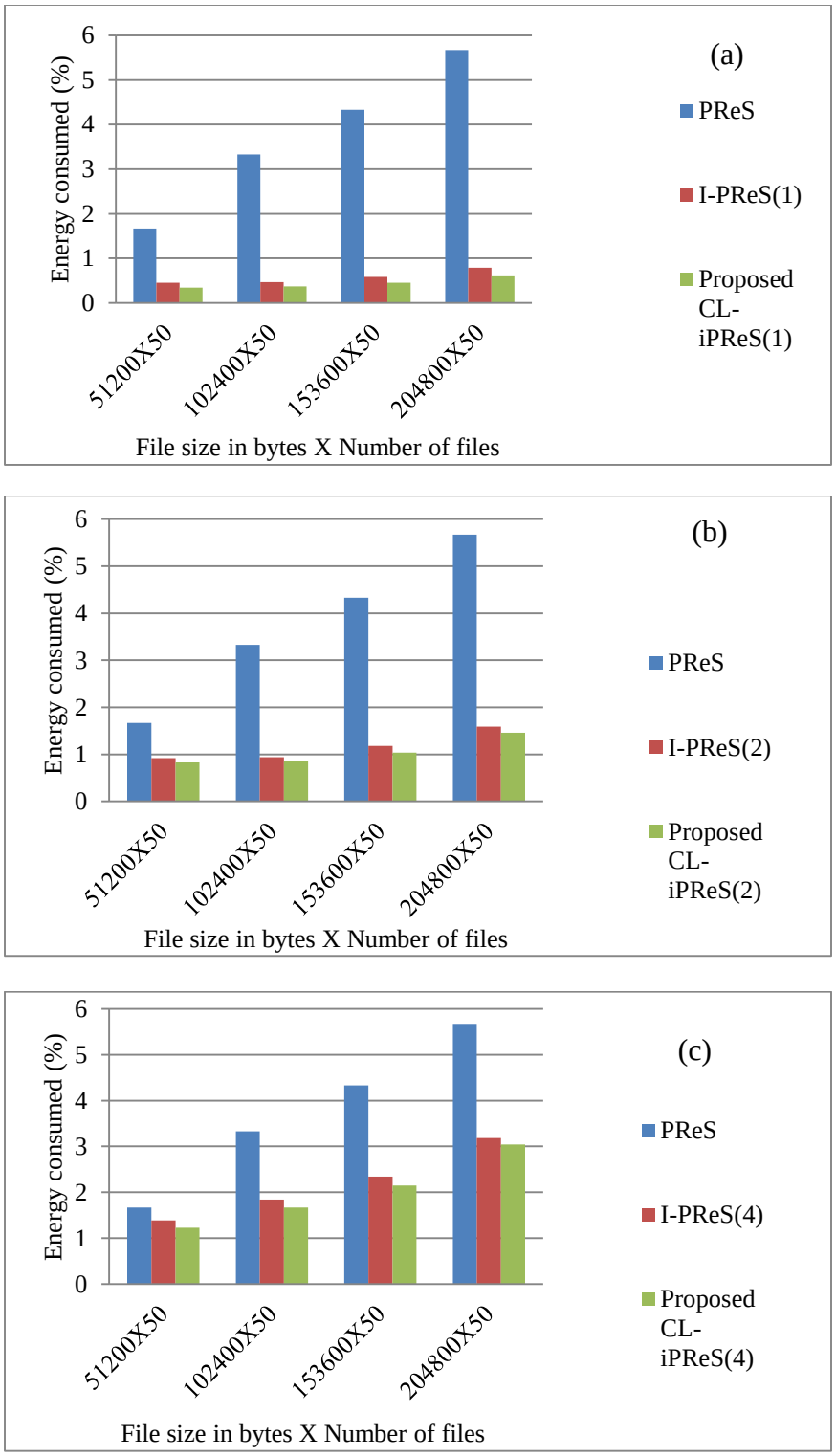

Figure 5.3: Energy consumption while block updating operations on dataset (a) Updating 1 block (b) Updating 2 blocks (c) Updating 4 blocks

This clearly depicts that the proposed scheme is computationally efficient than existing schemes PReS and I-PReS for file modification operations. The communication overhead of proposed CL-iPRE scheme is less as compared to I-PReS because individual MAC values need not to be uploaded on cloud for storage and cost of elliptic curve point multiplication is less as compared to pairing based scalar multiplication.

5.6 Summary

While utilizing cloud resources for storing and sharing encrypted EHRs, security is a critical issue. Incremental proxy re-encryption solutions hold great promise in the zettabyte era. Recently, proposed I-PRes scheme involves expensive bilinear pairing operations and downloading of intermediate MAC values of all blocks from the cloud by mobile user while block insertion, deletion or updation operation, thereby increasing communication, computation and storage overhead on a mobile device. An efficient and lightweight pairing free incremental CL-iPRE scheme based on ECC for secure sharing of EHRs in MCC paradigm has been proposed. The proposed scheme handles block modification operations efficiently and reduces storage overhead on mobile device and public cloud making it viable for modern resource-constrained mobile devices.

Chapter 6

Conclusion and Future Scope

The integration of mobile computing, wireless networks and cloud computing, known as MCC, is a rapidly emerging area. This chapter provides the major findings of thesis work and future research opportunities in this field. Several studies have been conducted for data security in cloud computing to build a level of trust between CSPs and consumers, but security issues for MCC environment have not been explored much, and there exist many challenges in the security policies as MCC is still in the preliminary stages of research. In this thesis, this gap has plugged by conducting a comprehensive study to analyze various lightweight cryptographic solutions for data security in MCC.

After having a thorough review of various existing approaches or frameworks for data security in MCC, it was realized that the symmetric key cryptography, public key cryptography, and identity based cryptography approaches had few limitations in secure sharing of data in MCC paradigm. The certificateless cryptography provides the security against key escrow problem and solves certificate management problem in traditional PKI. To fulfill the research objectives, two schemes have been cryptanalyzed and three data security schemes based on certificateless cryptography have been proposed. In terms of computation overhead, pairing operation is the most expensive operation, among other public key operations, such as elliptic curve point multiplication, elliptic curve point addition, etc. Therefore, the proposed schemes utilize ECC to reduce computation overhead. The contributions made in the thesis are:

- The first contribution is towards cryptanalysis and improvement of CLPSc for E-Prescription system in mobile cloud computing. Recently, a pairing free CLPSc scheme has been proposed by Yanfeng et al. [32] which claims its security against forgery under adaptive chosen-message attacks. In this thesis, it has been unveiled that the aforementioned scheme has failed to provide unforgeability. As an

improvement of their scheme, a novel pairing-free CLPSc scheme using ECC has been proposed for e-prescription system in mobile cloud computing. The proposed scheme is proven to be secure against indistinguishability under adaptive chosen-ciphertext attack and existential forgery under an adaptive chosen-message attack in the random oracle model against Type 1 and Type 2 adversaries through formal analysis. The proposed scheme outperforms the existing schemes in terms of computational efficiency, making it suitable for futuristic mobile cloud computing applications.

- The second contribution is towards cryptanalysis and improvement of a pairing-free CL-PRE Scheme for secure sharing of mobile personal healthcare records in cloud. First of all, cryptanalysis of Qin's scheme [33] is performed breaching confidentiality of their scheme. As an improvement to the scheme, a lightweight and pairing free single hop unidirectional CL-PRE scheme based on elliptic curves has been proposed for secure sharing of mobile PHRs with public cloud competent for low-power mobile devices. Its security has been proved through formal analysis against chosen ciphertext attack in the random oracle model. Also, the proposed scheme has been compared to the existing proxy re-encryption schemes which further proved that proposed CLPRE scheme is the most efficient.

- The third contribution is towards the development of secure Incremental Proxy Re-encryption for e-healthcare data sharing in MCC. An elliptic curve based pairing-free CL-iPRE scheme has been proposed for improving the file modification operations. The proposed scheme shows significant improvement in file modification operations in terms of turnaround time and energy consumption on mobile device.

The verification of the proposed schemes is done by making use of mathematical theorems and lemmas. The computation overhead has been measured by counting the number of basic operations such as the bilinear pairing, exponentiation, point multiplication, ciphertext size, re-encryption key size etc. which can be helpful in deciding the efficiency of the algorithm. The same is validated using experimental simulation setup. The

simulation setup includes Android SDK, Firebase, GAE and MIRACL cryptographic library.

Future Scope: The work discussed in this thesis opens new avenues for future research. Few of them have been summarized below:

- Designing an algorithm for broadcasting signcrypted messages which enable the sender to simultaneously encrypt and sign the message intended for a specific set of users in a single logical step.

- This thesis focuses on CL-PRE schemes in the random oracle model. The future work may include designing an efficient unidirectional and multi-hop provably CCA-secure CL-PRE schemes in the standard model.

- In future, a group-based incremental proxy re-encryption scheme for efficient data sharing can also be designed.

References

[1] Abolfazli S et al (2013) Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges. IEEE Communications Surveys & Tutorials 16(1), pp 1-32.

[2] Sanaei Z, Abolfazli S, Gani A, Buyya R (2014) Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Communications Surveys & Tutorials 16(1), pp 369-392.

[3] Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Communications and Mobile Computing 13(18), pp 1587-1611.

[4] Özsu MT, Valduriez P (2011) Principles of distributed database systems, 3rd edn. Springer Science & Business Media.

[5] Keshavamurthy BN, Khan AM, Toshniwal D (2013) Privacy preserving association rule mining over distributed databases using genetic algorithm. Neural Computing and Applications 22(1) pp 351-64.

[6] Yang S et al (2013) Fast dynamic execution offloading for efficient mobile cloud computing. In: Proceedings of the International Conference on Pervasive Computing and Communications (PerCom), IEEE, SanDiego, California, USA, pp 20-28.

[7] Khan AN, Kiah MLM, Khan SU, Madani SA (2013) Towards secure mobile cloud computing: a survey. Future Generation Computer Systems 29(5), pp 1278-1299.

[8] Somorovsky J et al (2011) All your clouds are belong to us: security analysis of cloud management interfaces. In: Proceedings of the 3rd ACM workshop on Cloud computing security workshop, New York, USA, pp 3-14.

[9] Sermersheim J (2006) Lightweight Directory Access Protocol (LDAP): The Protocol. RFC 4511 http://www.ietf.org/rfc/rfc4511.txt. Accessed on 10 January 2016.

[10] Grobauer B, Walloschek T, Stocker E (2011) Understanding Cloud-Computing Vulnerabilities. IEEE Security & Privacy 9(2), pp 50-57.

[11] Liu F et al (2013) Gearing Resource-Poor Mobile Devices with Powerful Clouds: Architecture, Challenges and Applications. IEEE Wireless Communications Magazine 20(3), pp 14-22.

[12] Modi C et al (2013) A survey on security issues and solutions at different layers of Cloud computing. The Journal of Supercomputing 63(2), pp 561-592.

[13] Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications 34(1), pp 1-11.

[14] Jansen W, Grance T (2011) Guidelines on security and privacy in public cloud computing. NIST special publication, 800(144):10-11.

[15] Argyroudis P, Verma R, Hellwich O, Tewari H, O'Mahony D (2004) Performance analysis of cryptographic protocols on handheld devices. In: Proceedings of the Third IEEE International Symposium Network Computing and Applications, Cambridge, MA, pp 169-174.

[16] Tewari H, Reid K, Ennis D (2012) U.S. Patent Application No. 14/348,288.

[17] Lamport L (1981) Password authentication with insecure communication. Communications of the ACM 24(11), pp 770-772.

[18] Ren W, Zeng L, Liu R, Cheng C (2016) F2AC: A Lightweight, Fine-Grained, and Flexible Access Control Scheme for File Storage in Mobile Cloud Computing. Mobile Information Systems 2016, pp 1-9. doi:10.1155/2016/5232846.

[19] Umar MS, Rafiq MQ, Ansari JA (2012) Graphical user authentication: A time interval based approach. In: Proceedings of the IEEE International Conference on Signal Processing, Computing and Control (ISPCC), IEEE, pp 1-6.

[20] Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers 51(5), pp 541-552.

[21] Jangirala S, Mukhopadhyay S, Das AK (2017) A Multi-server Environment with Secure and Efficient Remote User Authentication Scheme Based on Dynamic ID using Smart Cards. Wireless Personal Communications, pp 1-33.

[22] Halevi S, Krawczyk H (1999) Public-key cryptography and password protocols. ACM Transactions on Information and System Security 2(3), pp 230-268.

[23] Hwang MS, Li LH (2000) A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46(1), pp 28-30.

[24] Eyeprint ID. http://www.eyeverify.com/product-technology/core-technology. Accessed on 3 April 2016.

[25] BioID Web Services (BWS). http://www.bioid.com/products/bws.html. Accessed on 3 April 2016.

[26] Face.com Opens Free Facial Recognition API. http://www.programmableweb.com/api/face.com. Accessed on 3 April 2016.

[27] Animetrics Face Recognition (FaceR) API. http://animetrics.com/facer-api-for-face-recognition-applications/. Accessed on 3 April 2016.

[28] Cloud Based Biometric Authentication Solution for an Online Testing and Assessment company. http://www.calsoftlabs.com/resources/cloud-biometric-authentication-solution.html. Accessed on 3 April 2016.

[29] Zonouz S et al (2013) Secloud: A cloud based comprehensive and lightweight security solution for smartphones. Computers and security 37(9), pp 215-227.

[30] Cisco Visual Networking Index (2016) Global mobile data traffic forecast update, 2015–2020. White paper. Accessed on 6 February 2016.

[31] Mobile BPM Market by Solution, Service (Maintenance & Support, Integration & Design, Consulting, and Others), End User (SMBs and Enterprises), Vertical, Deployment Model (Public, Private, Hybrid), and Region-Global Forecast to 2020, Markets and Markets. http://www.marketsandmarkets.com/Market-Reports/mobile-bpm-market-31726258.html. Accessed on 15 February 2016.

[32] Yanfeng Q, et al (2013) Certificateless proxy identity-based signcryption scheme without bilinear pairings. China Communications, 10(11), pp 37-41.

[33] Qin Z, Wu S, Xiong H (2015) Strongly Secure and Cost-Effective Certificateless Proxy Re-encryption Scheme for Data Sharing in Cloud Computing. In: Wang Y, Xiong H, Argamon S, Li XY, Li JZ (eds) Big Data Computing and Communications, Springer International Publishing, pp 205-216.

[34] Shamus Software Ltd., Miracl library, https://certivox.org/display/EXT/MIRACL. Accessed on 20 September 2016.

[35] Simmons GJ (1979) Symmetric and asymmetric encryption. ACM Computing Surveys (CSUR) 11(4), pp 305-330.

[36] Neuman BC, Theodore Ts'O (1994) Kerberos: An authentication service for computer networks. IEEE Communications Magazine, 32(9), pp 33-38.

[37] Stallings, W (2006) Cryptography and network security: principles and practice. Prentice Hall.

[38] Krawczyk H, Canetti R, Bellare M (1997) HMAC: Keyed-hashing for message authentication. https://tools.ietf.org/html/rfc2104. Accessed on 10 January 2015.

[39] Pirzada AA, McDonald C (2004) Kerberos-Assisted Authentication in Mobile Ad Hoc Networks. In: Proceedings of the 27th Australasian Conference on Computer Science (ACSC '04), Dunedin, New Zealand, pp 41-46.

[40] Carman DW, Kruus PS, Matt BJ (2000) Constraints and approaches for distributed sensor network security. DARPA Project report (Cryptographic Technologies Group, Trusted Information System, NAI Labs) 1(1).

[41] Ali M et al (2015) SeDaSC: Secure Data Sharing in Clouds. IEEE Systems Journal PP(99), pp 1-10. doi:10.1109/JSYST.2014.2379646.

[42] Ren W et al (2011) Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing. Journal of Tsinghua Science and Technology 16(5), pp 520-528.

[43] Khan AN et al (2014) BSS: block-based sharing scheme for secure data storage services in mobile cloud environment. The Journal of Supercomputing 70(2), pp 946-976.

[44] Diffie W, Hellman ME (1976) New directions in cryptography. IEEE Transactions on Information Theory 22(6), pp 644-654.

[45] Koscielny C, Kurkowski M, Srebrny M (2013) Public key infrastructure. In: Modern Cryptography Primer, Springer Berlin Heidelberg. pp 175-191.

[46] Hellman M (1978) An overview of public key cryptography. IEEE Communications Magazine 16(6):24-32.

[47] Rivest R, Shamir A, Adleman L (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21(2), pp 120-126.

[48] ElGamal T (1985) A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory 31(4), pp 469-472

[49] Miller V (1985) Use of elliptic curves in cryptography. CRYPTO. Lecture Notes in Computer Science 85, pp 417-426.

[50] Lamport L (1979) Constructing digital signatures from a one-way function. Palo Alto: Technical Report CSL-98, SRI International.

[51] Merkle R (1990) A certified digital signature. In: Gilles Brassard G (ed) Advances in Cryptology -CRYPTO '89. Spring Verlag, pp 218–238.

[52] Mehuron W (1994) Digital Signature Standard (DSS). US Department of Commerce, National Institute of Standards and Technology (NIST). Information Technology Laboratory (ITL). FIPS PEB. 1994:186.

[53] Han JH, Kim YJ, Jun SI, Chung KI, Seo CH (2002) Implementation of ECC/ECDSA cryptography algorithms based on Java card. In: Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, Vienna, Austria, pp 272-276.

[54] Lee S et al (2010) Two factor authentication for cloud computing. Journal of information and communication convergence engineering 8(4), pp 427-432.

[55] Xiao S, Gong W (2010) Mobility can help: protect user identity with dynamic credential. In: Proceedings of the 11th International Conference on Mobile Data Management (MDM), IEEE, Kansas City, Missouri, USA, pp 378-380.

[56] Khan AN, Kiah MM, Madani SA, Ali M (2013) Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. The Journal of Supercomputing 66(3):1687-1706.

[57] Khan AN et al (2013) A study of incremental cryptography for security schemes in mobile cloud computing environments. In: Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA), Kuching, Malaysia, IEEE, pp 62-67.

[58] Khalid U et al (2013) Cloud based secure and privacy enhanced authentication & authorization protocol. Procedia Computer Science, 22, pp 680-688.

[59] Annappaian DH, Agrawal VK (2015) Multilevel Cryptography with Metadata and Lock Approach for Storing Data in Cloud. Transactions on Networks and Communications 2(6), pp 47-55.

[60] Lauter K (2004) The advantages of elliptic curve cryptography for wireless security. IEEE Wireless Communications 11(1), pp 62-67.

[61] Toorani M, Beheshti A (2008) LPKI-a lightweight public key infrastructure for the mobile environments. In: Proceedings of the 11th International Conference on Communication Systems (ICCS), IEEE, Guangzhou, China, pp 162-166.

[62] Krawczyk H (2005) HMQV: A high-performance secure Diffie-Hellman protocol. Advances in Cryptology – CRYPTO'05, LNCS 3621, Springer-Verlag, pp 546-566.

[63] Baek J, Steinfeld R, Zheng Y (2007) Formal proofs for the security of signcryption. Journal of Cryptology 20(2), pp 203-235.

[64] Myers M et al (1999) X.509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP, RFC 2560. http://www.ietf.org/rfc/rfc2560.txt. Accessed on 10 March 2016.

[65] Pinkas D, Housley R (2002) Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC 3379. http://www.ietf.org/rfc/rfc3379.txt. Accessed on 12 March 2016.

[66] Boneh D, Franklin MK (2001) Identity-based encryption from the Weil pairing. In Kilian J (ed) Advances in Cryptology-CRYPTO 2001. Springer Berlin Heidelberg, pp 213-229.

[67] Boneh D, Franklin MK (2003) Identity-based encryption from the Weil pairing. SIAM Journal on Computing 32(3), pp 586–615.

[68] Gentry C, Silverberg A (2002) Hierarchical ID-based cryptography. In: Zheng Y (ed) Proceedings of Asiacrypt 2002. Springer Berlin Heidelberg, pp 548-566.

[69] Ragini P, Mehrotra S, Venkatesan (2014) An efficient model for privacy and security in Mobile Cloud Computing. In: Proceedings of the International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, pp 1-6.

[70] Li H, Dai Y, Tian L, Yang H (2009) Identity-based authentication for cloud computing. In: Jaatun M (ed) Cloud computing. Springer Berlin Heidelberg, pp 157-166.

[71] Dong X et al (2015) SECO: Secure and scalable data collaboration services in cloud computing. Computers & Security 50(5), pp 91-105.

[72] Liu W et al (2016) SAKE: scalable authenticated key exchange for mobile e-health networks. Security and Communication Networks, 9(15), pp 2754–2765.

[73] Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Cramer R (ed) Advances in Cryptology-EUROCRYPT. Springer Berlin Heidelberg, pp 457-473.

[74] Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp 89-98.

[75] Lv Z, Hong C, Zhang M, Feng D (2012) A secure and efficient revocation scheme for fine-grained access control in cloud storage. In: Proceedings of the 4th International Conference on Cloud Computing Technology and Science (CloudCom'12), Taipei, Chennai, India, pp 545-550.

[76] Yang K, Jia X, Ren K, Zhang B, Xie R (2013) DAC-MACS: effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security 8(11), pp1790-1801.

[77] Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy Attribute-Based Encryption. In: Proceedings of 2007 IEEE Symposium on Security and Privacy, Berkeley, California, USA, pp 321–334.

[78] Hur J, Noh DK (2011) Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Transactions on Parallel and Distributed Systems 22(7), pp 1214-1221.

[79] Yu S, Wang C, Ren K, Lou W (2010) Achieving secure, scalable, and fine-grained data access control in cloud computing. In: Proceedings of IEEE Conference on Computer Communications, San Diego, California, USA, pp 1-9.

[80] Sun GZ (2011) CP-ABE based data access control for cloud storage. Journal of China Institute of Communications 32(7), pp 146-152.

[81] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2), pp 38-47.

[82] Huang D et al (2010) Mobicloud: building secure cloud framework for mobile computing and communication. In: Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering (SOSE), Nanjing, China, pp 27-34.

[83] Zhou Z, Huang D (2011) Efficient and secure data storage operations for mobile cloud computing. In: Proceedings of the 8th International Conference on Network and Service Management, Laxenburg, Austria, pp 37-45.

[84] Yu S, Wang C, Ren K, Lou W (2010) Attribute based data sharing with attribute revocation. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, pp 261-270.

[85] Li F, Rahulamathavan Y, Rajarajan M, Phan RC (2013) Low complexity multi-authority attribute based encryption scheme for mobile cloud computing. In: Proceedings of the 7th International Symposium on Service Oriented System Engineering (SOSE), IEEE, San Francisco, USA, pp 573-577.

[86] Chase M, Chow SS (2009) Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 16th ACM conference on Computer and Communications Security (CCS ’09), Chicago, Illinois, USA, pp 121-130.

[87] Lv Z, Chi J, Zhang M, Feng D (2014) Efficiently Attribute-Based Access Control for Mobile Cloud Storage System. In: Proceedings of the 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, Beijing, China, pp 292-299.

[88] Fei L, Rahulamathavan Y, Rajarajan M (2014) LSD-ABAC: Lightweight static and dynamic attributes based access control scheme for secure data access in mobile environment. In: Proceedings of the 39th Conference on Local Computer Networks (LCN), IEE, Edmonton, Alberta, Canada, pp 354-361.

[89] Dong X et al (2014) Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. Computers & Security 42(5), pp 151-164.

[90] Wu W et al (2017) Towards secure and cost-effective fuzzy access control in mobile cloud computing. Soft Computing 21(10), pp 2643-2649.

[91] Al-Riyami S, Paterson K (2003) Certificateless public key cryptography. In: Laih CS (ed) Advances in cryptology-ASIACRYPT. Springer Berlin Heidelberg, pp 452-473.

[92] Xu L, Wu X, Zhang X (2012) CL-PRE: A certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Republic of Korea, pp 87-88.

[93] Seo SH, Nabeel M, Ding X, Bertino E (2014) An efficient certificateless encryption for secure data sharing in public clouds. IEEE Transactions on Knowledge and Data Engineering 26(9), pp 2107-2119.

[94] Yang C, Wang F, Wang X (2007) Efficient mediated certificates public key encryption scheme without pairings. In: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), Niagara Falls, Ontario, Canada, pp 109-112.

[95] Tsai J (2015) A new efficient certificateless short signature scheme using bilinear pairings. IEEE Systems Journal, PP(99), pp 1-8, doi:10.1109/JSYST.2015.2490163.

[96] Zhu L, Tung B (2006) RFC 4556: Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). IETF Network Working Group.

[97] Harbitter A, Menasce DA (2001) The performance of public key-enabled kerberos authentication in mobile computing applications. In: Proceedings of the 8th ACM conference on Computer and Communications Security, Philadelphia, Pennsylvania, pp 78-85.

[98] Fox A, Gribble SD (1996) Security on the move: indirect authentication using Kerberos. In: Proceedings of the 2nd Annual International conference on Mobile Computing and Networking, Rye, New York, USA, pp 155-164.

[99] Park KW, Lim SS, Park KH (2008) Computationally efficient PKI-based single sign-on protocol, PKASSO for mobile devices. IEEE Transactions on Computers 57(6), pp 821-834.

[100] Yang J et al (2011) Provable data possession of resource constrained mobile devices in cloud computing. Journal of Networks 6(7), pp 1033-1040.

[101] Wang Q et al (2009) Enabling public verifiability and data dynamics for storage security in cloud computing. In: Proceedings of the 14th European Conference on Research in Computer Security (ESORICS '09), Saint-Malo, France, pp 355-370.

[102] Carts DA (2011) A review of the Diffie–Hellman algorithm and its use in secure internet protocols. http://www.sans.org/reading_room/whitepapers/vpns/review-diffiehellman-algorithm-secure-internet-protocols 751. Accessed on 10 April 2016.

[103] Nagaty KA (2014) Mobile Health Care on a Secured Hybrid Cloud. Journal of Selected Areas in Health Informatics 4(2), pp 1-9.

[104] Sujithra M, Padmavathi G, Narayanan S (2015) Mobile Device Data Security: A Cryptographic Approach by Outsourcing Mobile Data to Cloud. Procedia Computer Science 47, pp 480-485.

[105] Blaze M, Bleumer G, Strauss M (1998) Divertible protocols and atomic proxy cryptography. In: Nyberg K (ed) Advances in Cryptology-EUROCRYPT'98, Springer Berlin Heidelberg, pp 127-144.

[106] Green M, Ateniese G (2007) Identity-based proxy re-encryption. In: Katz J (ed) Applied Cryptography and Network Security. Springer Berlin Heidelberg, pp 288-306.

[107] Jia W et al (2011) SDSM: a secure data service mechanism in mobile cloud computing. In: Proceedings of the IEEE Conference on Computer Communications Workshops, Shanghai, China, pp 1060-1065.

[108] Tysowski PK, Hasan MA (2011) Re-encryption-based key management towards secure and scalable mobile applications in clouds. In: IACR Cryptology ePrint Archive 668.

[109] Khan AN, Kiah MM, Ali M, Shamshirband S (2015) A Cloud-Manager-Based Re-Encryption Scheme for Mobile Users in Cloud Environment: a Hybrid Approach. Journal of Grid Computing 13(4), 651-675.

[110] Bellare M, Goldreich O, Goldwasser S (1994) Incremental cryptography: the case of hashing and signing. In: Advances in cryptology-CRYPTO'94. Springer, Berlin, pp 216-233.

[111] Bellare M, Goldreich O, Goldwasser S (1995) Incremental cryptography and application to virus protection. In: Proceedings of the 27th annual ACM symposium on theory of computing, ACM, Riva, Italy, pp 45-56.

[112] Bellare M, Micciancio D (1997) A new paradigm for collision-free hashing: Incrementality at reduced cost. In: Advances in cryptology–EUROCRYPT'97. Springer, Berlin, pp 163–192.

[113] Itani W, Kayssi A, Chehab A (2010) Energy-efficient incremental integrity for securing storage in mobile cloud computing. In: Proceedings of the International Conference on Energy Aware Computing (ICEAC '10), IEEE, Cairo, Egypt, pp 1-2.

[114] Khan AN et al (2014) Incremental proxy re-encryption scheme for mobile cloud computing environment. The Journal of Supercomputing 68(2), pp 624-651.

[115] Zheng Y (1997) Signcryption and its applications in efficient public key solutions. In: International Workshop on Information Security. Springer, Berlin, Heidelberg, pp 291-312.

[116] Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems 27(2), pp 340-352.

[117] Mambo M, Usuda K, Okamoto E (1996) Proxy signature for delegating signature operation. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, pp 48–57.

[118] Shamir A (1984) Identity-based cryptosystems and signature schemes. In: Workshop on the Theory and Application of Cryptographic Techniques, Santa Barbara, CA, USA, pp 47-53.

[119] Jin Li et al (2015) Identity-based Encryption with Outsourced Revocation in Cloud Computing. IEEE Transactions on Computers 64(2), pp 425-437.

[120] Hwang MS, Tzeng SF, Tsai CS (2004) Generalization of proxy signature based on elliptic curves. Computer Standards & Interfaces 26(2), pp 73-84.

[121] Hu X et al (2017) An improved efficient identity-based proxy signature in the standard model. International Journal of Computer Mathematics 94(1), pp 22-38.

[122] El-Kamchouchi H, Gaber H, Ahmed F, El-Kamchouchi DH (2016) An Efficient Proxy Signature Scheme over a Secure Communications Network. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering 10(7), pp 1226-1229.

[123] Zheng Y (1997) Digital signcryption or how to achieve cost (signature & encryption)≪ cost (signature)+ cost (encryption). In: Proceedings of Annual International Cryptology Conference, Santa Barbara, California, USA, pp 165-179.

[124] Barbosa M, Farshim P (2008) Certificateless signcryption. In: Proceedings of the 2008 ACM symposium on Information, computer and communications security, New York, USA, pp 369-372.

[125] Yin A, Liang H (2015) Certificateless Hybrid Signcryption Scheme for Secure Communication of Wireless Sensor Networks. Wireless Personal Communications 80(3), pp 1049-1062.

[126] Islam SH, Li F (2015) Leakage-free and provably secure certificateless signcryption scheme using bilinear pairings. The Computer Journal 58(10), pp 2636-2648.

[127] Debiao H, Jianhua C, Jin H (2011) An ID-based proxy signature schemes without bilinear pairings. Annals of telecommunications 66(11-12), pp 657-662.

[128] Jing X (2011) Provably secure certificateless signcryption scheme without pairing. In: Proceedings of the International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT), Harbin, Heilongjiang, China, pp 4753-4756.

[129] Selvi SSD, Vivek SS, Rangan CP (2009) Cryptanalysis of certificateless signcryption schemes and an efficient construction without pairing. In: Bao F (ed), International Conference on Information Security and Cryptology, Beijing, China, pp 75-92.

[130] Xie W, Zhang Z (2010) Certificateless Signcryption without Pairing. IACR Cryptology ePrint Archive, 187.

[131] Shi W, Kumar N, Gong P, Zhang Z (2014) Cryptanalysis and improvement of a certificateless signcryption scheme without bilinear pairing. Frontiers of Computer Science, 8(4), pp 656-666.

[132] Gamage C, Leiwo J, Zheng Y (1999) An efficient scheme for secure message transmission using proxy-signcryption. In: Proceedings of the 22nd Australasian Computer Science Conference, Sydney, Australia, pp 420-431.

[133] Jung HY, Lee DH, Lim JI, Chang KS (2001) Signcryption schemes with forward secrecy. In: Proceedings of Information Security Application (WISA'01), Seoul, Korea, pp 403-475.

[134] Elkamchouchi HM, Abouelseoud Y (2008) A New Proxy Identity-Based Signcryption Scheme for Partial Delegation of Signing Rights. IACR Cryptology ePrint Archive:41.

[135] Wang C, Han Y, Li F (2009) A secure mobile agent protocol for m-commerce using self-certified proxy signcryption. In: Proceedings of Second International Symposium on Information Science and Engineering (ISISE'09), Washington, DC, USA, pp 376-380.

[136] Lin HY, Wu TS, Huang SK, Yeh YS (2010) Efficient proxy signcryption scheme with provable CCA and CMA security. Computers & Mathematics with Applications 60(7), pp 1850-1858.

[137] Lo NW, Tsai JL (2014) A provably secure proxy signcryption scheme using bilinear pairings. Journal of Applied Mathematics. doi:10.1155/2014/454393.

[138] Ming Y, Wang Y (2015) Proxy signcryption scheme in the standard model. Security and Communication Networks 8(8), pp 1431-1446.

[139] Zhou CX (2016) Identity Based Generalized Proxy Signcryption Scheme. Information Technology and Control, 45(1), pp 13-26.

[140] Yeh JH (2014) The insecurity of two proxy signcryption schemes: proxy credential forgery attack and how to prevent it. The Journal of Supercomputing, 70(3), pp 1100-1119.

[141] The Certicom Corporation, SEC 2: Recommended Elliptic Curve Domain Parameters. http://www.secg.org/SEC2-Ver-1.0.pdf. Accessed on 20 September 2016.

[142] Castiglione A, De Prisco R, De Santis A (2009) Do you trust your phone?. In: Proceedings of International Conference on Electronic Commerce and Web Technologies, Springer Berlin Heidelberg, pp 50-61.

[143] Bhatia T, Verma AK (2017) Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues. The Journal of Supercomputing 73(6), pp 2558–2631.

[144] Li J, Li J, Liu Z, Jia C (2014) Enabling efficient and secure data sharing in cloud computing. Concurrency and computation: practice and experience 26(5), pp 1052-1066.

[145] Li J, Jia C, Li J, Liu Z (2012) A novel framework for outsourcing and sharing searchable encrypted data on hybrid cloud. In: Proceedings of 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS), Bucharest, IEEE, pp 1-7.

[146] Lin HY, Tzeng WG (2012) A secure erasure code-based cloud storage system with secure data forwarding. IEEE transactions on parallel and distributed systems 23(6), pp 995-1003.

[147] Nuñez D, Agudo I, Lopez J (2017) Proxy Re-Encryption: Analysis of Constructions and its Application to Secure Access Delegation. Journal of Network and Computer Applications. doi:10.1016/j.jnca.2017.03.005

[148] Smith T (2005): DVD jon: Buy DRM-less tracks from Apple iTunes, January 2005. http://www.theregister.co.uk/2005/03/18/itunes pymusique. Accessed on 12 December 2016.

[149] Ateniese G, Fu K, Green M, Hohenberger S (2006) Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transactions on Information and System Security 9(1), pp 1-30.

[150] Heydt-Benjamin TS, Chae HJ, Defend B, Fu K (2006) Privacy for public transportation. In: Danezis G, Golle P (eds). Privacy Enhancing Technologies, Springer Berlin Heidelberg, pp 1-19.

[151] Canetti R, Hohenberger S (2007) Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM conference on Computer and communications security, ACM, pp 185-194.

[152] Libert B, Vergnaud D (2011) Unidirectional chosen-ciphertext secure proxy re-encryption. IEEE Transactions on Information Theory 57(3), pp 1786-1802.

[153] Deng RH, Weng J, Liu S, Chen K (2008) Chosen-ciphertext secure proxy re-encryption without pairings. In: Franklin MK, Hui LK, Wong DS (eds). Cryptology and Network Security, Springer Berlin Heidelberg, pp 1-17.

[154] Wang XA, Ma J, Yang X (2015) A new proxy re-encryption scheme for protecting critical information systems. Journal of Ambient Intelligence and Humanized Computing 6(6), pp 699-711.

[155] Chu CK, Tzeng WG (2007). Identity-based proxy re-encryption without random oracles. In: In: Garay JA, Lenstra AK, Mambo M, Peralta R (eds). Information Security, Springer Berlin Heidelberg, pp 189-202.

[156] Tang Q, Hartel P, Jonker W (2008) Inter-domain identity-based proxy re-encryption. In: Yung M, Liu P, Lin D (eds). Information Security and Cryptology, Springer Berlin Heidelberg, pp 332-47.

[157] Wang L, Wang L, Mambo M, Okamoto E (2010) New identity-based proxy re-encryption schemes to prevent collusion attacks. In: Joye M, Miyaji A, Otsuka A (eds). Pairing-Based Cryptography, Springer Berlin Heidelberg, pp 327-346.

[158] Mizuno T (2011) Secure and efficient IBE-PKE proxy re-encryption. IEICE transactions on fundamentals of electronics, communications and computer sciences 94(1), pp 36-44.

[159] Shao J, Cao Z (2012) Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption. Information Sciences, 206, pp 83-95.

[160] Liang K, Liu Z, Tan X, Wong DS, Tang C (2012) A CCA-secure identity-based conditional proxy re-encryption without random oracles. In: Kwon T, Lee MK, Kwon D (eds). Information Security and Cryptology, Springer Berlin Heidelberg, pp 231-246.

[161] Liang K, Liu JK, Wong DS, Susilo W (2014) An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In: Kutylowski M, Vaidya J (eds). Computer Security, Springer Berlin Heidelberg, pp 257-272.

[162] Wang C, Fang J, Li Y (2015) An Improved Cloud-Based Revocable Identity-Based Proxy Re-encryption Scheme. In: Proceedings of International Conference on Applications and Techniques in Information Security, Beijing, China, pp 14-26.

[163] Sur C, Jung CD, Park Y, Rhee KH (2010) Chosen-ciphertext secure certificateless proxy re-encryption. In: Proceedings of IFIP International Conference on Communications and Multimedia Security, Linz, Austria, Springer Berlin Heidelberg, pp 214-232.

[164] Zheng Y, Tang S, Guan C, Chen MR (2013) Cryptanalysis of a certificateless proxy re-encryption scheme. In: Proceedings of Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp 307-12.

[165] Guo H, Zhang Z, Zhang J, Chen C (2013) Towards a secure certificateless proxy re-encryption scheme. In: Susilo W, Reyhanitabar R (eds). Provable Security, Springer Berlin Heidelberg, pp 330-46.

[166] Yang K, Xu J, Zhang Z (2013) Certificateless proxy re-encryption without pairings. In: Lee HS, Han DG (eds). Information Security and Cryptology, Springer International Publishing, pp 67-88.

[167] Wang LL, Chen KF, Mao XP, Wang YT (2014) Efficient and provably-secure certificateless proxy re-encryption scheme for secure cloud data sharing. Journal of Shanghai Jiaotong University, 19, pp 398-405.

[168] Srinivasan A, Rangan CP (2015). Certificateless proxy re-encryption without pairing: Revisited. In: Proceedings of the 3rd International Workshop on Security in Cloud Computing, ACM, pp 41-52.

[169] Selvi SSD, Paul A, Rangan CP (2017). An Efficient Certificateless Proxy Re-Encryption Scheme without Pairing. In: Proceedings of the International Conference on Provable Security. Springer, Cham, pp 413-433.

[170] Liang X, Cao Z, Lin H, Shao J (2009) Attribute based proxy re-encryption with delegating capabilities. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, ACM, pp 276-286.

[171] Liang K et al (2015) A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. Future Generation Computer Systems 52, pp 95-108.

[172] Turner V, Gantz JF, Reinsel D, Minton S. The digital universe of opportunities: Rich data and the increasing value of the internet of things. IDC Analyze the Future. 2014.

[173] Cisco. Cisco VNI: Forecast and Methodology, 2015–2020. White Paper, June 2016. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html. Accessed on 23 March, 2017.

[174] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021. White Paper, March 2017. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Accessed on 3 March 2017.

[175] Hsueh SC, Lin JY, Lin MY (2011) Secure cloud storage for convenient data archive of smart phones. In: Proceedings of the 15th International Symposium on Consumer Electronics (ISCE), IEEE, Singapore, pp 156-161. doi: 10.1109/ISCE.2011.5973804

[176] Buonanno E, Katz J, Yung M (2001) Incremental unforgeable encryption. In: Matsui M (ed). Fast Software Encryption. FSE 2001. Lecture Notes in Computer Science, vol 2355. Springer, Berlin, Heidelberg, pp 109-124.

[177] Mironov I, Pandey O, Reingold O, Segev G (2012) Incremental deterministic public-key encryption. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp 628-644.

[178] PUB F. Secure hash standard. Public Law. 1995;100:235.

[179] FIPS P. 180-2, 2004. Secure Hash Standard, National Institute of Standards and Technology, US Department of Commerce. 2004.

[180] SHA N. Standard: Permutation-Based Hash and Extendable-Output Functions (DRAFT FIPS PUB 202). http://csrc.nist.gov/publications/PubsDrafts.html#FIPS-202. Accessed on 25 March 2017.

[181] Merkle R (2006) A digital signature based on a conventional encryption function. In: Advances in Cryptology-CRYPTO'87, pp 369-378.

[182] Bertoni G, Daemen J, Peeters M, Van Assche G (2014) Sufficient conditions for sound tree and sequential hashing modes. International Journal of Information Security 13(4), pp 335-353.

[183] Wang X, Yao AC, Yao F. Cryptanalysis on SHA-1. In Cryptographic Hash Workshop hosted by NIST 2005.

[184] CWI. Google announce first collision for Industry Security Standard SHA-1 Accessed on 16 February 2017.

[185] Mihajloska H, Gligoroski D, Samardjiska S (2015) Reviving the Idea of Incremental Cryptography for the Zettabyte era Use case: Incremental Hash Functions Based on SHA-3. In: Camenisch J, Kesdoğan D (eds). Open Problems in Network Security. Lecture Notes in Computer Science, 9591. Springer, Cham, pp 97-111.

[186] Koblitz AH, Koblitz N, Menezes (2011) A Elliptic curve cryptography: The serpentine course of a paradigm shift. Journal of Number theory, 131(5), pp 781-814.

[187] Dahl M, Ning C, Toft T (2012) On secure two-party integer division. In International Conference on Financial Cryptography and Data Security, Springer Berlin Heidelberg, pp 164-178.