SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI

A Thesis submitted to the

UPES

For the award

of

Doctor of Philosophy

In

Computer Science

By

Kapil Kant Kamal

September 2024

SUPERVISORS

Dr. Sunil Gupta
Dr. Padmaja Joshi
Dr. Monit Kapoor

SCHOOL OF COMPUTER SCIENCE (SOCS)
UPES
Dehradun Uttarakhand

SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI

A Thesis submitted to the

UPES

For the award

of

Doctor of Philosophy

In

Computer Science

By

Kapil Kant Kamal

(SAP ID: 500072285)

Internal Supervisor:
Dr. Sunil Gupta

Professor
University of Petroleum And Energy Studies

External Supervisor:
Dr. Padmaja Joshi
Scientist G
Centre for Development of Advanced Computing (C-DAC), Mumbai
Dr. Monit Kapoor
Professor
University Institute of Engineering and Technology, Chitkara University

SCHOOL OF COMPUTER SCIENCE (SOCS)
UPES
Dehradun Uttarakhand

DECLARATION

I declare that the thesis entitled **SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI** has been prepared by me under the guidance of Dr. Sunil Gupta, Professor, UPES, Dr. Padmaja Joshi, Scientist G, C-DAC and Dr. Monit Kapoor, Professor, Chitkara University, Punjab. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

Kapil Kant Kamal

School of Computer Science
UPES
Bidholi via Prem Nagar
Dehradun-248007,Uttarakhand, India
Date: 24.06.2024

UPES
UNIVERSITY OF TOMORROW

Great Place To Work. Certified FEB 2024-FEB 2025 INDIA

CERTIFICATE

I certify that Kapil Kant Kamal has prepared his thesis entitled **"SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out work at School of Computer Science, University of Petroleum & Energy Studies.

UPES
UNIVERSITY OF TOMORROW

Dr. Sunil Gupta
Professor
School of Computer Science
University of Petroleum & Energy Studies
Dehradun -248007, Uttarakhand
Date: 24/06/2024

प्रगत संगणन विकास केन्द्र
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING
इलेक्ट्रॉनिक्स और सूचना प्रौद्योगिकी मंत्रालय की वैज्ञानिक संस्था
भारत सरकार
A Scientific Society of the Ministry of Electronics and Information Technology
Government of India

गुलमोहर क्रास रोड सं. 9, जुहू
मुंबई - 400 049 , भारत
Gulmohar Cross Road No.9, Juhu
Mumbai - 400 049. India
दूरभाष/ Tel : (022) 26201604 / 1574/ 1488
फैक्स/ Fax : +91-22-26232195
http://www.cdac.in

24 June, 2024

CERTIFICATE

I certify that Kapil Kant Kamal has prepared his thesis entitled "**SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI",** for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the Department of Computer Science, University of Petroleum & Energy Studies.

External Supervisor

Dr. Padmaja Joshi
Scientist 'G'
Group Head (SENG)
Centre for Development of Advanced Computing (C-DAC)
Gulmohar Cross Road No.9, Juhu, Mumbai 400049.

डॉ. पद्मजा जोशी/Dr. Padmaja Joshi
वरिष्ठ निदेशक/Senior Director
प्रगत संगणन विकास केंद्र
Centre for Development of Advanced Computing
संचार और सूचना प्रौद्योगिकी मंत्रालय की वैज्ञानिक संस्था
A Scientific Institution of the Ministry of Communication and Information Technology
गुलमोहर [illegible] Road No.
[illegible] मुंबई/Juhu, Mumbai-400 049.

CHITKARA UNIVERSITY
PUNJAB

CERTIFICATE

I certify that Kapil Kant Kamal has prepared his thesis entitled "**SECURE AND EFFICIENT MOBILE AS A PERSONAL IDENTITY USING PKI**", for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the Department of Computer Science, University of Petroleum & Energy Studies.

External Supervisor

Department of Computer Science & Engineering
Chitkara University Institute of
Engineering & Technology
Chitkara University, Punjab 140401

Dr. Monit Kapoor
Professor and Dean
Department of Computer Science and Engineering,
University Institute of Engineering and Technology,
Chitkara University
Rajpura, Punjab 140401
Date: 24th, June 2024

University Campus
Chandigarh - Patiala National Highway (NH-7)
Punjab - 140 401, T +91.1762.507084
Fax +91.172.507085

Administrative Office
Saraswati Kendra, SCO 160 - 161
Sector 9-C , Chandigarh - 160009
T +91 172 4090900

www.chitkara.edu.in

ABSTRACT

Today, smartphones are mainly used for communication and access devices for various services. Most people access services using smartphones, not just calls, social media, or watching movies. It is also necessary to provide user identity/authentication through mobile devices.

With the new wave of digital governance and digital payment initiatives taken by the Indian government focusing on service delivery through mobile devices to increase reachability even to the citizens in rural areas, it has become more compelling to ensure the safety of the personal information shared by the users while authenticating using mobile phones. The information is usually stored in the device. However, many users know the risks of compromising privacy while exposing information during online transactions. Phone users have no control over the data stored in the device or collected by apps, and they are uninformed of how those companies handle their data.

This thesis focuses on bridging this gap and making the mobile device a safe medium for online transactions using a mobile-based identity. The thesis contribution is - (i) A framework for mobile identity on a handheld device, (ii) key pair generation and implementation within the handheld device using cryptography algorithms, (iii) identifying trusted and secure storage on the mobile device for securing private keys; and (iv) a mobile-based signing (mCK) solution for mobile devices.

While numerous mobile identity options are available globally, they all have limitations. Most rely on Hardware Security Module (HSM) systems, SIMs, and card-based systems that depend on external hardware and telecom service providers for user authentication and signing through mobile devices. These solutions, while functional, do not provide the level of control and security that our proposed framework can offer.

This research delves deeper into the demand for a novel mobile-based user identity framework that operates on mobile devices based on PKI to safeguard the user's sensitive information and signing requests. The framework incorporates biometric information and Elliptic Curve Cryptography (ECC). A crucial aspect of this framework is the user's full control over their identity and data during online transactions. This feature aligns with the current discourse on digital privacy and security.

Most of the currently available solutions generate key pairs outside the mobile device and store them in the SIM again, which is not under the owner's control.

Hence, the security and privacy of the data remain a major concern. In the research work done under this thesis, an architecture has been proposed that ensures secured storage of keys within the handheld device. This framework enables users to manage their own digital identities without depending on third-party providers.

The research work is also supported through the implementation of various use cases. The thesis also discusses various design decisions in its implementation. Analyzing the underlying trust assumptions is also necessary to fully emerge with mobile identity management. As a result, various trust-related issues have been studied in more depth throughout the thesis.

Keywords: Mobile Identity, ECC, PKI, Mobile Signing, Privacy,Authentication , Key Security, Key storage;

ACKNOWLEDGEMENT

I express my deepest gratitude to my supervisors, Dr. Sunil Gupta, Dr. Padmaja Joshi, and Dr. Monit Kapoor, for their invaluable guidance, support, and expertise throughout this research. Their exceptional knowledge, patience, and mentorship have been instrumental in shaping this PhD thesis and my academic journey. I am fortunate to have had the opportunity to work under their supervision. Their insightful feedback, constructive criticism, and constant encouragement have significantly supported the development of this thesis. Their dedication and commitment to academic excellence have been a continuous source of inspiration for me.

I am extremely grateful to my family, for their unwavering love, support, and understanding. Their sacrifices, encouragement, and belief in my abilities have been the foundation of my success. Their constant presence and unwavering support have given me the strength and motivation to pursue my academic goals. Furthermore, I would like to thank God for granting me strength, guidance, and blessings throughout this research journey. The divine presence in my life has been a source of inspiration and comfort during both challenging and joyous moments.

Additionally, I would like to express my gratitude to Dr. Vinod Patidar, the Ph.D. coordinator, for their support, guidance, and administrative assistance throughout the doctoral program. Furthermore, I would like to acknowledge the members of my SRC panel, Dr. Ajay Prasad, Dr. Adarsh Kumar, and Dr. Ved Prakash Bharadwaj, for their valuable insights, constructive feedback, and thorough evaluation of my research. Their expertise and critical evaluation have greatly contributed to the refinement and quality of this thesis.

I am also grateful to my mentor and friend for his invaluable support, encouragement, and insightful discussions. His guidance and expertise have been instrumental in shaping my research direction and enhancing the quality of my work. His friendship and mentorship have made this journey both rewarding and enjoyable.

I wholeheartedly thank my well-wishers, friends, and colleagues for believing in and encouraging me through their motivational words, which have helped me sail through the research journey. Last but not least, I am thankful and indebted to all those who helped me directly or indirectly complete this research study.

Contents

List of Figures

List of Abbreviations

AE Authenticated Encryption

AAKA Anonymous Authenticated Key Agreement

AKA Authenticated Key Agreement

API Application Programming Interface

CA Certificate Authority

CW Control Word

CRL Certificate Revocation List

DES Data Encryption Standard

DoS Denial-of-Service

DSA Digital Signature Algorithm

DMS Distributed Measurement System

e-Gov eGovernance

ECC Elliptic Curve Cryptography

EECC Enhanced Elliptic Curve Cryptography

ECDSA Elliptic Curve Digital Signature Algorithm

ECDLP Elliptic Curve Discrete Logarithm Problem

eID Electronic Identification

ESP e-Sign Service Provider

FIDO Fast Identity Online

GPS Global Positioning System

HSM Hardware Security Module

HCC Hyperelliptic curve cryptography Algorithm

IT Information Technology

ICT Information and Communication Technology

ID Identity

IETF Internet Engineering Tasks Force

IMS Identity Management Systems

IdP Identity Providers

IoT Internet of Thing

IP Internet Protocol

IIA Identity Issuing Authority

IMEI International Mobile Equipment Identity

IBS Identity Based Signature

IdM Identity Management

KGC Key Generation Centre

mID Mobile Identity

MNO Mobile Network Operators

mCK Mobile based Signing

m-ID Mobile ID

MSS Mobile Secure Space

mSign Mobile Signing

NFC Near Field Communication

NIST National Institute of Standards and Technology

NS Nanosecond Time

OTP One Time Password

OIS Owner Identification Service

OCSP Corresponding Public Keys

PKI Public Key Infrastructure

PIN Personal Identification Number

PII Personally Identifiable information

PPP Public-Private Partnership

PK Public Key

QR Quick Response

RSA Rivest, Shamir, Adleman

RDI Real Digital Identity

RA Registration Authority

SFA Single Factor Authentication

SIM Subscriber Identity Module

SP Service Providers

SAML Security Assertion Markup Language

SE Secure Element

SSO Single Sign-On

SK Secret Key

SS Secure Space

SK Private Key

TEE Trusted Execution Environment

TZ Trust Zone

TSP Telecom Services Provider

VIM Various Identity Management

VA Validation Authorities

Chapter 1

Introduction

Over the previous two decades, technological advancement, transformation, and innovation have predominantly powered economies worldwide. Thus, the two technological innovations, mobile phones or smartphones and the Internet, have dramatically altered how people interact with the government. The transformation and fusion of these technologies have resulted in globally expanding Information Technology(IT) services. The advancement of ICT Information and Communication Technology (ICT) has impacted governments' services, processes, operations, and effectiveness. One of the main effects of ICT-driven practices has been m-government.

It was predicted that the number of smartphone users worldwide would constantly rise between 2024 and 2029, adding 1.5 billion people (or 30.6 percent). Following fifteen consecutive years of growth, the number of smartphone users is projected to reach 6.4 billion, marking a new record in 2029. Significantly, the number of individuals utilizing smartphones has steadily risen in recent years. Smartphone users in this context refer to individuals of any age who use a smartphone to access the internet. Figure1.1, shows the same.

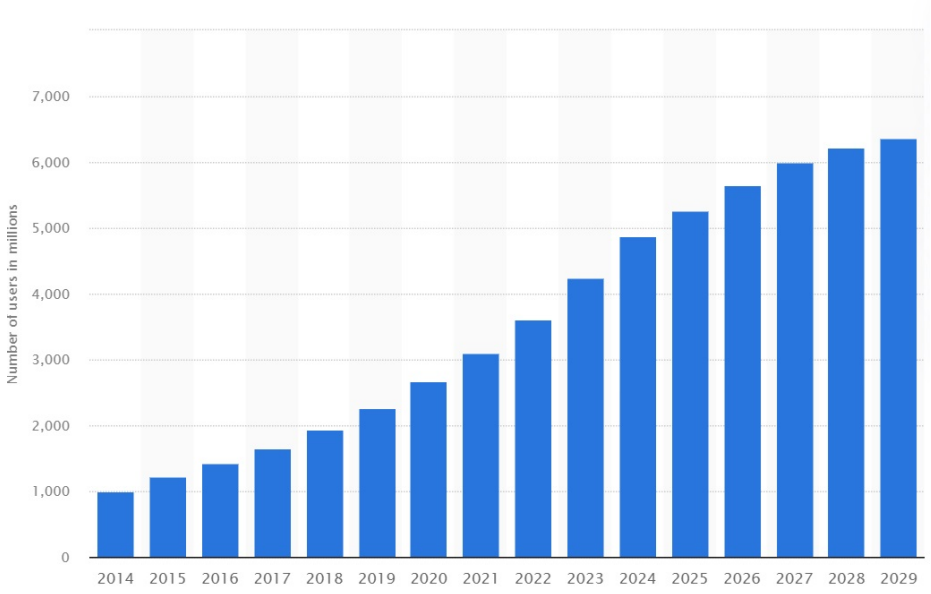


Figure 1.1: Number of smartphone users worldwide (Statista, 2024)

m-Government offers an option to access government data and services and do online transactions as per user requirements by utilizing wireless and mobile technology platforms. Using mobile technology in place of current e-government procedures has helped close the digital divide in society and contributed to the democratization of technology and the development of a society with equal access to and usability of services (Lee et al., 2010).

The internet does not have a trusted digital identity, even though almost all our daily activities are done through it. Cyberspace is currently the only arena in which a person can easily trust anybody or whatever they wish to be uploaded. Social media profiles are self-identifying, so the user has full control over their credentials, like name, birth date, address, and career, which are all created by the user. However, some are genuine, and some are fake. In a self-sovereign system, there is no way to verify that a person belongs to the claimed entity. These forged identities cause many serious problems while providing e-Governance services.

This work provides a framework for mobile identification for mobile-based transactions that includes a PKI-based communication model, authenticating users, signing documents, and improved secure storage. In addition, the suggested solution is low-cost, scale-able, and independent of external hardware. The goal of the study is to comprehend and evaluate how mobile identity affects mobile-based transactions in

the context of current m-Governance practices.

The focus of this work is the mobile ID-based authentication of e-Gov programmes. Governments use national identity management strategies to manage citizens' mobile electronic identities. Digital identity system development and implementation are essential in the e-Governance and m-Governance initiatives. We created a mobile ID authentication strategy for mobile client-server contexts by combining PKI infrastructure, Enhanced ECC (EECC), and tokens used for authentication in our research. The aim of this research is to provide an ecosystem that uses mobile devices to access online services with secure citizen identification. To comprehend the process of re-energizing current e-governance practices using mobile technology. This chapter explores the background, significance, objectives, hypothesis, methodology, motivation, and scope of this research work.

1.1 Prerequisites

This section provides information about important prerequisites.

1.1.1 Digital Identity

Digital Identity Includes privacy-sensitive identity attributes. It represents a single person and can be used for both identification and authentication. It must be safeguarded with high levels of security and privacy.

Users frequently request services from multiple service providers (SP) daily. In addition, they necessitate access to several systems and resources. In order to create new accounts or perform transactions, users must verify their identity by providing SPs with privacy-sensitive personal information. These identifying qualities might be considered personal information.

- User - Tries to access a system or a resource or requests a service.
- Identity Provider - Government body for issuing user identities for accessing services.
- SP - Offers identity verification, authentication, and prompt responses to service requests.

Mobile Identity

One of the biggest problems facing cybersecurity experts is identity because of the inherent anonymity of the internet. Governments, industry, and individuals want to know how to manage and trust a user's identity. Identity can be faked or impersonated in today's technologically embedded social and corporate situations. People want to know if the people communicating on social media or interacting with websites are who they say they are. Industry and governments aim to verify the identity of those granted access to their networks or websites. A digital identity, or electronic identification (eID), is the virtual representation of an individual or organization's real-world identity. An entity refers to a corporate organization, a governing body, or a tangible item. Every device that is linked to a network requires the utilization of a digital identity.

Mobile ID can be the digital equivalent of a passport, license, or other ID. It contains a Digital Certificate that confirms the validity and includes the holder's identity (name, email, address, etc). A Private Key that can be used to sign documents is also included with a Mobile ID (Verzeletti et al., 2018). (Boontaetae et al., 2018). Figure1.2, shows the same. Mobile ID Enablers:

- Users:
  - User no longer need to carry physical identification documents because they can carry their digital identities on their mobile devices.
  - Users may have more control over their personal data through mobile identity solutions.
- Services:
  - Mobile identity can be used across multiple services (government services, financial transactions, healthcare, travel, etc.).
  - Citizens accept and ask for e-services.
- m-ID Security:
  - The legal setting is conducive to the adoption of m-ID.
  - The m-ID security component is handled most efficiently to maintain information availability, confidentiality, and integrity.

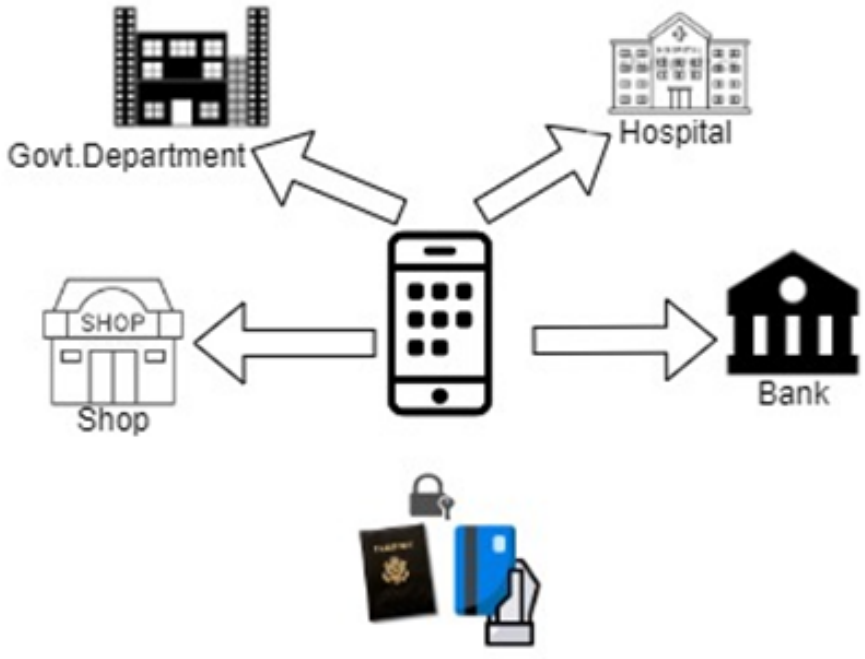


Figure 1.2: Mobile ID Enablers

1.1.2 Mobile-Based Authentication

There is a huge potential for providing public services via mobile phones due to the widespread use of mobile phones, increasing internet usage, and more affordable and capable smartphones (Martens, 2010). The use of various applications on smart mobile devices is becoming increasingly important (Martens, 2010). (En-Nasry & Dafir Ech-Cherif El Kettani, 2011). Mobile devices have thus become an emerging potential player in the field of digital identity for the provision of public services. Numerous institutions, organizations, and sectors use mobile phones to verify online users. Users must authenticate themselves using their mobile phones, typically where one-time password (OTP) and/or passwords and personal identification number (PIN) are primarily used, to access services via mobile phones or mobile applications. Identity management systems typically use username-password, digital signature, OTP,PIN, grids, and cryptography algorithms to authenticate an entity during online transactions. These mechanisms do, however, have inherent limitations, particularly when it comes to granting access to e-government services, where information or data leaks can severely impair the service. As a result, there is a chance that a false identity will

be provided during mobile transactions, which could result in online fraud (En-Nasry & Dafir Ech-Cherif El Kettani, 2011). (Bicakci et al., 2014). Therefore, it is essential to consider mobile security when suggesting a mobile device as a digital identity.

The most commonly used online transactions and user identity verification method is a password. Passwords are a single-factor authentication (SFA), which limits the number of ways that a person can be identified when requesting access to resources. Logging into emails, social media, bank accounts, and shopping websites requires passwords to authenticate users. We are urged to utilize it while ensuring it is unique for every online transaction to provide user identity.

Mobile-based Authentication techniques used with mobiles are described ahead.

- User Name and Password: A username and password combination is used to authenticate the subject in the case of a mobile holder. Users can access any mobile application by using their chosen username and password. This method is uncommon for mobile devices because it can be uncomfortable for some people to type passwords on a smaller screen without a full-sized keyboard. Additionally, many users would use the same simple passwords from their desktop accounts on all mobile applications and services, posing a serious security risk to the average user and defeating the purpose of authentication.

- OTP: Using an OTP (one-time password) that they received via email or mobile, users can demonstrate their identity. OTP-based authentication will be used in conjunction with each type of authentication. A one-time password is a string of letters or numbers that a computer program or system has randomly generated for use only once. One-time passwords essentially use what a person may have in possession and OTP users' mobile devices to verify that they are using the correct device. OTP is not just a random number; rather, algorithms using suitable seeds for use only once and within a set time frame produce it. TOTP, Ping Pong-128, and HOTP are well-known algorithms for generating OTP (Lee et al., 2010).

- Mobile Token/Certificate with Password: Using hardware tokens, users can verify their identities (along with their PIN). Tokens, carry digital certificates and can be used for digital signatures. It is also a similar kind of authentication as that of OTP, as the token has a user. Additionally, hardware tokens can be

obtained through USB crypto, SD cards, or SIM extensions. Based on a mobile PKI, these applications will allow for secure mobile commerce and anytime, anywhere transactions for banking, payments, and other services. Anywhere a mobile-based secure transaction is required, mobile PKI can be used. Additionally, it can be used for digital signing and authentication (Fioravanti & Nardelli, 2008). (Silasai & Khowfa, 2020). (Bhargav-Spantzel et al., 2006). (Arabo et al., 2009).

- Quick Response (QR) Code-based: The user is shown a QR code created by the application. After that, it waits for the code to be scanned or until the timeout period is up. The OR code can be read using mobile application scanners, which can then be used to take the necessary action. Usually, this authentication mechanism is used for the payments. The QR code is associated with the payment account.

- Biometric-based Authentication: Physical and behavioral biometrics fall under this category. Physical biometric authentication uses a person's fingerprint, face, peri-ocular region, and/or iris as distinctive characteristics. People can also be identified using behavioral biometrics, which records a person's actions or manner of behavior, such as voice, keystroke dynamic, and touch dynamic (Husni, 2016). Behavioral biometrics is used as a support factor for authentication, whereas physical biometrics can be used as the only factor for authentication. Mobile devices have access to various sensors, including face and fingerprint recognition. Mobile device sensors adhere to universally recognized standards, so the data they collect can be regarded as authentic to use biometrics to verify an individual's identity (Silasai & Khowfa, 2020). Figure 1.3 represents the same.

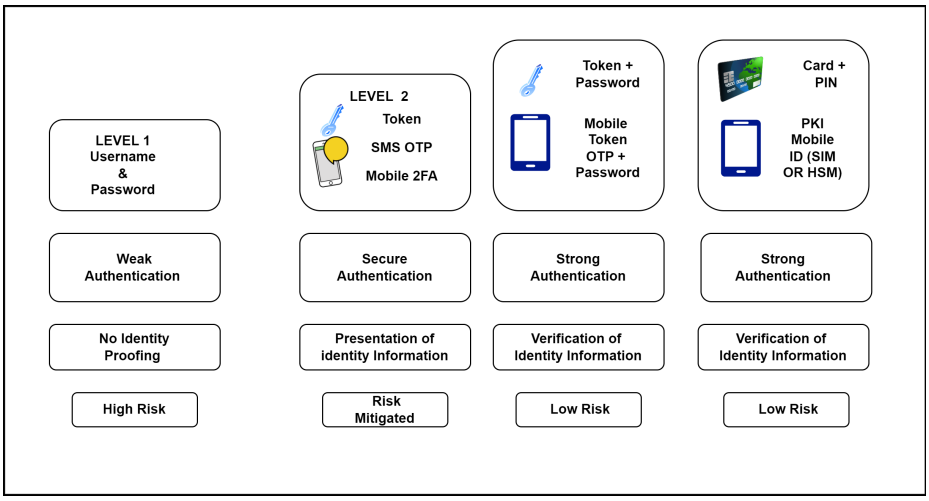


Figure 1.3: Assurance levels and Authentication Methods

1.1.3 Cryptography and Key Management

Cryptography is the science of protecting data. Messages have been obscured for generations to prevent unintended recipients from viewing them. Encryption refers to the technique of encoding a communication so that it is unintelligible to those who are not authorized. Modern cryptography is based on mathematical theory and computer science ideas. Key management involves overseeing the whole lifecycle of cryptography keys and guarding against their loss or unauthorized usage. The key lifetime encompasses creation, distribution, use, storage, archiving, and deletion.(Al-Khouri, 2011). Cryptography has 4 properties that are: confidentiality, Authenticity, integrity, and Non-repudiation.

- Confidentiality: It ensures that a message is only read by those who are supposed to read it.
- Integrity establishes that a message has not been altered unauthorized.
- Non-repudiation ensures that the legitimacy of an electronic message or transaction cannot be denied.
- Authentication verifies the identity of entities involved in the communication.

PKI technology identities can be securely verified online. To use public-key encryption securely, it defines the policies and practices required to create, maintain,

verify, and distribute certificates. PKI management is based on the X.509 certificate standard, which enables external parties to verify a private key's ownership certificate authority CA. A data structure called an X.509 certificate is described as tying public-key values to subjects (e.g. domain name, organization name). A trusted CA demonstrates the binding by digitally signing each certificate. This claim may be supported by the CA thoroughly validating the private certificate holder's identity. (Verzeletti et al., 2018).

There are three types of cryptographic keys used for safe and secure communication/storage:

- Symmetric key - One key can be used to encrypt and decode information. This means the key to encrypting the information must also be used like the key to decrypting.

- Asymmetric cryptography- often known as public key cryptography, is a technique that encrypts and decrypts a message using two related keys, one public and one private, to prevent unauthorized access or usage. A public key is a cryptography key that enables an individual to encrypt a message that can only be deciphered by the intended receiver using their private key. A private key, also called a secret key, is exclusively shared with the individual who initiated the key.

- Hash keys – This algorithm does not use any keys. Hashed data is mostly used for data integrity verification. It is used with MD5, SHA-1, SHA-2, NTLM, and LANMAN algorithms to protect the authenticity of data and transactions.

1.1.4 Digital Certificate

Digital signatures are cryptography methods that give digital documents or messages non-repudiation, integrity, and authentication. Digital certificates are electronic credentials that provide details about the subject, validity, certificate holder, serial number, and services that use the certificate to connect a person's identity, computer, or service to a public key. Digital signatures, encryption, and authentication can all be accomplished with certificates. The X.509 has defined standards that the certificates issued in PKIs are constructed to fulfill.(Verzeletti et al., 2018).

The Digital certificate Contains the following Information:

- The version number of the certificate: There are numerous variations of certificates. The certificate's version number specifies the fields and formatting, as well as how the applications that use it are to interpret the certificate.
- Certificate serial number: Each certificate is given a special serial number. As a specific point of reference, this field is used when the certificate is revoked.
- Signature algorithm: The algorithm used to sign documents digitally through a certificate is specified in this field, such as RSA with MD5.
- Issuer X.500 Name: This is the issuing company's name according to the X.500 naming convention.
- Public key information: Contains details of certificate holder public key.
- Issuer unique identifier: The issuer is identified by this special code.
- Subject unique identifier: The subject is identified by this special code.
- Digital signature: The certificate's digital signature is located in this field.

1.2 Opportunities of Mobile ID

Government-issued identity for citizens includes passports, identification cards, driving licenses, and election ID on paper or cards. Physical ID does not meet these standards but provides some process modifications in a restricted. Mobile ID is significant because it allows users to identify themselves safely and remotely while fulfilling all four criteria. (Kerttula, 2015). (Wu et al., 2018). (e-Estonia, n.d.).
Mobile ID streamlines infrastructure decreases deployment costs, allows access to various services, and gives government organizations a more secure way to identify residents. The usage of Mobile IDs can lead to the dematerialization of any public services involving citizen authentication and identity because most citizens already have mobile devices that can hold Mobile IDs However, its application lies in various categories, including health emergency assistance, booking service, information services, billing, management, games, and potential financial applications (Group, n.d.-a). ((ITU), n.d.). Some potential applications in the realm of m-Governance are discussed below.

- Digital Voting: Governments can provide anonymous or pseudonymous identity, authentication, and/or authorization for online voting.
- Social Welfare Services: Govt can enable secure access to citizens to avail of all types of services through a mobile device and provide digital identity.
- Financial: The financial sector can use digital identity to provide all financial services through mobile phones, which will boost security and minimize fraud costs through digital ID. However, Mobile ID can provide a convenient form of digital signature that will enable new secure online financial transactions and banking applications.
- Travel Security: Governments can use it for the secure identification of airlines and trains. There may also be visa application possibilities.
- Tax Collection: The government can collect income tax, tax deductions, declarations, and other information through Mobile ID.
- Police Services: Multiple systems, such as motor vehicle information, criminal histories, insurance records, health records, and weapons registries, may be accessible to police. This proposal can be used by police forces to derive accurate and appropriate data.

1.3 Research Gaps

- In handheld devices, a fake digital identity issue poses a security risk.
- Mobile based signing not available.
- In current scenarios, key generation outside the Mobile device on external infrastructure Key Generation and storage of the keys are not under the user's control.
- Existing mobile phone-based cryptography solutions are all SIM-based and provided by telecom service providers, leading to third-party dependency.
- Electronic Signature key generation and signing are carried out by an e-Sign Service Provider (ESP) who is a CA under Public Key Infrastructure. Another

issue is the Aadhaar-based authentication for each certificate created at (ASP) side, which can cause a delay in receiving an SMS due to load on servers.

- The current approach user has no control over key generation and doesn't bind the user's identity with the mobile phone.

1.4 Problem Description

Providing cryptography-based solutions for key generation, user authentication, and signing on Mobile devices with user control over the complete mechanism.

1.5 Objectives of the Work

The present research aims to design a mobile-based user identity framework and digital signing processes, ensuring a seamless and secure user experience independent of hardware and telecom operators working on smartphones.

- Design a framework to provide efficient and secure mobile user identity and signing solutions through handheld devices.
- Analyzing and enhancing key pair generation that binds handheld devices.
- Identifying trusted and secure storage on the mobile device for securing private keys.
- Mobile-based Signing (mCK) solution for mobile devices.

1.6 Our Contributions

This section highlights our major contribution to addressing the identified problems.

This research proposes a novel way of authenticating a person and signing documents using mobile devices. It proposes a technique for key pair generation inside the mobile phone device that is independent of the external hardware. Currently, many SIM card-based user authentication schemes and e-sign schemes are available. However, these solutions bind the users to the TELCOs. None of the existing solutions give flexibility to the user. An electronic signature is used which has legal validity

as per the IT Act. However, it is termed as an Electronic Signature against the token-based Secure Digital Signature. This is because the key generation and signing are carried out by an ESP who is a CA under Public Key Infrastructure offered by the Controller of Certifying Authority. This research proposes an application-layer solution for cryptography-based authentication and document signing using mobile devices. The proposed solution's security is verified using automated formal verification methods against known cyber-attacks.

The significant contributions of this research work are as below.

1.6.1 Designing a Framework to Provide Efficient and Secure Mobile-user Identity and Signing Solutions through Handheld Devices.

- Generation of keys within the handheld device using cryptography algorithms.
- Application layer Mobile ID solution that will be in the user's control. The solution will be hardware agnostic but will require Android as the OS.
- Mobile devices may now store and safeguard keys without relying on additional hardware.
- Enhanced ECC – ECC creates the user's key pairs, i.e., a PK (public key) and SK (private key), whereas EECC has a third key that is a secret key that binds with the phone device to enhance the solution security.

1.6.2 Analyzing and Enhancing Key pair generation that binds handheld devices

- Identify the best-suited cryptography algorithm for key generation and management on mobile devices.
- A comparative study of existing cryptography algorithms and identify the best-suited algorithm. A comparative study of various cryptography algorithms was carried out. ECC was found to be the best-suited ECC curve and, hence, was chosen for implementation.

- EECC method based on an ECC algorithm to create the key pairs to meet the user's security criteria.

1.6.3 Mobile-based Signing (mCK) solution for mobile devices

- In the proposed scheme, the generation of certificates is initiated by the user and signed through the mobile device itself.
- Experimental results demonstrate the viability of the proposed solution that is used for signing purposes.
- Added Mobile ID unique Identifier and Device ID in the x509 certificate.

1.6.4 Identifying trusted and secure storage on the mobile device for securing private keys

- Explore and identify the solution to securely storing the keys on the mobile device.
- Provide an implementation to store the keys with the desired proposed solution.
- Accessed by the respective application involved in Cryptography operations like encrypting, signing, and hashing.

1.7 Organization of the thesis

The organization of this thesis is as follows:

Chapter 2, provides a comprehensive review of the literature survey, encompassing traditional mobile ID systems.
Chapter 3, introduces a "framework to provide efficient and secure mobile user identity and signing solutions for handheld devices." This framework is related to Mobile ID architecture using PKI technology, Enhancing elliptic curve cryptography (EECC), and token-based authentication. This approach is based on PKI and independent of SIM, card, or external hardware components. Further, we also evaluate

the proposed methods for mobile identity authentication.

Chapter 4, introduces: "Analyzing and Enhancing Key Pair Generation that Binds Handheld Devices". Enhance ECC is suggested in our effort. Traditional ECC generates two keys (public and private), whereas EECC generates an additional key that binds mobile phones for cipher-based text encryption. This unique key is created by combining the final two digits of the IMEI number and the device ID. Furthermore, this is key to the Caesar cipher. Encryption becomes more sophisticated with this approach, and decryption becomes more difficult.

Chapter 5, Mobile-based Signing (mCK) Solutions for Mobile Devices. Mobile devices are not just communication tools; they are increasingly becoming our digital identities. This shift in usage patterns underscores the importance of our research, which focuses on linking mobile to obtaining a digital identity. By doing so, we enable users to identify themselves in modern digital interactions.

Chapter 6, discusses "Trusted and Secure Storage on the Mobile Device for Securing Private Keys." Trusted execution environments (TEEs) are important to device security architecture. It provides a separate space for the execution of critical apps like user authentication, payments, and signing that run in a secure area and are not accessed by any other application. Moreover, it is now necessary to establish a robust security framework for this setting, especially when taking into account secure application execution.

Chapter 7, we conclude our work, emphasizing key findings and summarizing contributions. We also illuminate prospective directions and challenges within the realm, pointing toward future research and exploration areas.

Chapter 2

Literature Survey

The research aims to assist users in online transactions requiring identification. These activities require high privacy protection because very sensitive personal information is disclosed in those situations. As a result, the treated subdomain, which represents the context of this research, is concerned with identification processes, security, and privacy concerns that must be addressed when using mobile devices.

We present a comprehensive overview of all the related work, meticulously examining the use of mobile IDs for e-Gov and other applications. Our systematic literature review has identified loopholes in (i) traditional mobile eID solutions used in Government services and (ii) technologies and procedures utilized to offer lightweight key generation for mobile eID solutions. We have also expanded the scope of the study to determine how critical attributes are saved and retrieved. In the following sections, we list all relevant material and provide a detailed account of how our work distinguishes itself from previous research.

2.1 Related Work in Mobile ID

Many mobile identification options are available globally, relying on Hardware Security Module (HSM) systems and SIM-based servers. Many mobile identity systems have been created, mostly based on SIM-based and server-based HSM solutions. SIM card solutions are often favored due to their simplicity in implementation for mobile ID and provide tamper resistance. Nevertheless, adopting Mobile eID with such cards and SIM results in high costs and dependence on mobile network providers. (Kerttula, 2015). Certain nations have been setting the bar for mobile ID use in mGovernance.

We investigated the use of Mobile IDs for m-Governance and e-Government in many nations. This is a summary of the mobile ID solutions utilized in various nations.

- Estonia: Estonia uses Mobile-ID, which is accepted legally and is used for offering public services, including e-voting. The solution is known as Digidoc (e-Estonia, n.d.). In this solution, a SIM card chip stores a key pair. The MobileID owner uses the private key stored in the SIM to sign the documents digitally.

- Finland: All e-government services are accessible by integrating Finnish bank credentials with the eGov portal. Mobiilivarmenne (Group, n.d.-a). is a PKI-based authentication system used by DNA, Elisa, and Telia, three Finnish mobile providers. The foundation of Mobiilivarmenne consists of certifications found on the SIM cards of its subscribers.

- Oman: The national eID card is a mobile PKI SIM card-based ID that adds true mobility for eGov services in Oman ((ITU), n.d.).

- Azerbaijan: ASAN Imaza et al. (MobileID) system ((ITU), n.d.). The stakeholders in this Public-Private Partnership (PPP) model include three national Mobile Network Operators (MNO), this service provider, the governmental certification authority, and E-service providers. The mobile operator is linked to the system via their secure SIM cards.

- Iceland: Many of Iceland's services are accessible via MobileID. All Iceland mobile phone users utilize strong authentication and legally binding signatures to identify themselves when accessing government services or completing duties such as online banking. ((ITU), n.d.).

- Turkey: Consumers access their accounts online through MobileID. Turkcell, a cellular company, provides a mobile signature service, MobilImza, to offer a remote way to complete transactions with a signature equivalent to an "original" signature on a hard copy. This solution makes it possible to sign documents and authenticate the user using a mobile phone as MobileID, which is legally approved, secure, easy, and convenient (Turkcel, n.d.).

- Moldova: The government Center collaborated with the leading mobile network operators and the Center for Special Telecommunications to create the

mobile eID using a client-side approach, with the private key on a special crypto processor-enabled SIM card. eID is utilized by citizens and small enterprises who do not need to simultaneously sign several documents because the solution must be used to sign a single document simultaneously. (Group, n.d.-b).

- Spain: In Spain many municipal services are provided viz, towing information, a duplicate copy of vehicle tax payment form, mobile payment of taxes and fines, checking residents' registers and electoral census details, and residence certificates using Mobile ID.

- Austria: Austrian mobile e-ID is operated by a trusted service provider (SP) for qualified certificates, which manage cryptography keys on behalf of the citizen. Unlike other methods, the cryptography keys are kept and operated in an HSM device at the TSP, but the signatory controls it. ((ITU), n.d.).

With all these possibilities, it is clear that a mobile device can contribute to authenticating an individual. Mobile ID Solutions are available worldwide based on SIM cards as a secure element, as mentioned in the table 2.1.

Country Name	Solution	Based on
Estonia	Mobile-ID	SIM Card
Azerbaijan	Asan Imza	SIM Card
Iceland	Skilriki Service	SIM Card
Austria Norway	Mobile Phone Signature Bank ID	HSM & SIM Card
Lithuania	Lithuanian EI	SIM Card
Turkey	Signature Mobillmza	SIM Card
Moldove	Mobile e-ID	SIM Card
Switzerland	Swiss Mobile-ID	SIM Card
Finland	Mobiilivarmenne	SIM Card
Norway	Norway's BankID	SIM Card

Table 2.1: Related Work

2.2 Related Studies

2.2.1 Works Related to Identity and Signing Solutions through Mobile Devices

A digital identity is a primary focus of e-Government (Fioravanti & Nardelli, 2008). as it is highly important to identify who to contact precisely. Digital Identity must be handled well when providing online services to satisfy "security, privacy, and dependability" and the basic requirements. Hence, the key ideas in modern identity management are identification, authentication, and authorization. (Bhargav-Spantzel et al., 2006). claimed that the life cycle of identity comprises "enrollment, generation, storage, retrieval, provisioning, verification and revocation of identity attributes."

As per (Arabo et al., 2009). Identity Management (IdM) is a comprehensive framework that encompasses business processes, rules, and technologies. Its purpose is to enable organizations to manage and regulate user access to important online applications and resources while also protecting sensitive personal and corporate data from unauthorized individuals. In the realm of electronics, it pertains to how individuals conduct themselves in their daily endeavors and necessitates the seamless integration and compatibility of systems.

(Selvakumaraswamy & Govindaswamy, 2016). recommended algorithms for wireless device PKI. It has been discovered that the ECC algorithm and its variations can offer strong cryptography capabilities and maximize processing speed and space. Applications with limited resources can use ECC successfully. The authors have compared the RSA and ECC algorithms. The results show that RSA signatures are taking more time than ECC based signatures. The public key of the certificate is likewise smaller and more adaptable. Utilizing RSA-based certificates for verification results in increased processing time, particularly when using larger key strengths . In RSA, the private key operations exert a greater burden compared to the public key activities.

Saquib et al., 2017. suggested a one-time mobile originating PKI that allows users to generate key pairs on mobile devices and use them as sign documents on their phones. It signs content digitally with the user's private key (PK) and is verified with PKI standards using application level protocols. This solution offers a trustworthy and safe method of online transaction authentication. PKI service providers can

digitally sign and authenticate on a mobile device by utilizing an embedded ,tamper-proof, user-friendly, and multi-tenant solution.

(Verzeletti et al., 2018). proposed a national identity management strategy system for electronic services. The system concentrated on utilizing the FIDO, TEE, and SAML standards to provide a solution for security, usability, and privacy. The results show that the suggested eID solution can protect citizens' privacy. The author did discover through experimentation that the user's comprehension of alert messages had an impact on usability.

(Boontaetae et al., 2018). proposed a decentralized PKI system for real digital identity (RDI). The authors developed RDI using decentralized PKI protocols to address two issues: (1) false ID and (2) the traditional single point of trust paradigm, which violates privacy norms. The authors then examined the limits, stating that it would be difficult to update the RDI smart contract in case of an error. Pricing for information authoring varies. Including the block's revoked data may pose a serious danger.

(Wu et al., 2018). proposed a single user's generated key in two parts and stored across numerous devices by lightweight authentication protocols for mobile devices. Three parties comprise most of the system model for the proposed authentication scheme: Service provider (SP), a dependable service provider who gives registered users private keys. To be more precise, (SP) creates private keys in two parts for the different devices belonging to the registered user. To keep keys in sections, two devices are used. This will protect the key from thieves. The suggested solution offers two-factor security, which prevents an opponent from passing the verification of (SP) even if the adversary has access to one of the user's mobile devices, to ensure the security of the user's private key. Additionally, the protocol is designed to protect against key exposure attacks, which occur when a hacker gets access to a user's devices.

(AlMajed & AlMogren, 2020). Asymmetric cryptography takes a significant amount of computing power and storage space. The study aimed to develop a trustworthy method for authenticated encryption (AE) by strengthening the process of mapping plain text to an EC curve to fend off various encryption assaults. The authors prioritized enhancing the security of the encoding phase to protect against different encryption attacks, while previous methods neglected this crucial aspect. An analysis was conducted to evaluate the suggested method's security increase with existing methods in terms of execution speed, storage capacity, and power usage.

Healthcare system based on IoT, (Kavitha et al., 2019). have presented a framework, strong and secure authentication for group key agreements utilizing the hyperelliptic curve cryptography algorithm (HCC). The method that is being discussed addresses security flaws by combining a public key based on HCC with a digital signature algorithm (DSA) that ensures safe group communication and entity authentication. Through the use of appropriate procedures and effective security measures, comparative and performance analyses was carried out.

(Gong et al., 2020). investigated the issue of creating a private key (PK) and secret key (SK) on three trustworthy systems at the same time so that a passive listener might hear the entire public conversation. These legal terminals and assistants take note of the linked source sequences coming from the discrete memory less source (DMS) outputs. The SK-PK key capacity region is fully characterized by this model, and the source coding scheme further demonstrates the attainability of the knowledge domain. By tracking the rates of key leakage, the authors have evaluated the security performance of the two created keys. The fundamental concept utilized in the paper's module can be applied to various real-world scenarios (IoT, D2D, and vehicle networks).

(Jia et al., 2019). proposed Key management and encryption system. The AAKA "Anonymous Authenticated Key Agreement" system, which safeguards the exposure of the user's personal data and ensures the authenticity of his identity, has been proposed. For encryption, authors have used the ECC algorithm method with bilinear pairing. The ECC algorithm has enhanced operational effectiveness. The authors also established that ECC implements an identity-based cryptosystem designed for user privacy and traceability.

(Cooijmans et al., 2014). presented an analysis of Android-based secure key storage solutions. The author of this post examines the various secure key storage options that devices offer. Moreover, the author offers the Android Keystore to bind with the device to protect against root attackers by using the Trust Zone (TZ).

(Ji et al., 2019). presented a result of the volume of private data that modern mobile devices store; secure storage is required to protect them from hacking attempts. Today, the most popular method for safe storage on mobile devices is a Trusted Execution Environment (TEE). At the user layer, MicroTEE offers trusted access to services for daily activities performed by applications with security services like crypto services and key management. Through crypto services, TAs are given access

to fundamental cryptography operations,hashing, encryption, and signing.

The proposed article comprehensively explains the MicroTEE architecture, whose key elements are the TEE OS, Monitor, Root job, and security services. The study also includes performance analysis and a well-designed prototype implementation mechanism. Even though the suggested work is excellent, security measures should be somewhat improved. The PKI keys are stored in SIM or server, as indicated in most studies. This makes it more dependent on external devices and third parties. In contrast, the suggested approach focuses on storing the user's identity on the device itself. Some of the studies on this strategy highlight the difficulties it faces.

US Patent Application US 10/090422 (Assefa et al., 2016). This document outlines a technique and system that enables users to create a key pair (PK) private and (SK) and a corresponding user certificate. This certificate is used to verify his/her identity and sign his own certificate issued by a Certification Authority (CA).

US Patent (Sandberg & Rodberg-Larsen, 2004). Application 7,024,226 B2 describes a method for enabling SIM cards where a telephone operator generates a one-time activation code on a server. A SIM Card user's activation code was sent to their cellular phone via registered mail. Upon inputting the activation code into their mobile device, the user transmits it to the server for the purpose of authentication. After completing the verification process, the server transmits an activating command to the phone, which triggers the intended SIM card component. This may unlock PKI functions previously buried in the SIM card and, therefore, unavailable to the user. The user can then enter his signing PIN for authentication, encryption, and transaction signing. When the activation code is verified, the private and public key generation and certification setup occurs.

US Patent (Kim et al., 2019). describes a mobile device and method for providing certificate-based cryptography, including a receiver operative to receive a wireless transmission. Included within the wireless transmission is a certificate revocation notification. The mobile device and method include an authenticate operative to receive the certificate revocation notification and an operative to authenticate signed comparison data within the certificate revocation notification. Upon authentication, an updater is operative to update data representing at least one private or public key based on the certificate revocation notification.

US Patent Application US 10/497674 (Hiner et al., 2019). describes a method for distributing the key of an asymmetric key pair, along with a private key, from

a mobile station to a key-managing computer. The method involves the following steps: (1) The key managing computer securely communicates a password (OTP) to the mobile station of a registered user, establishing a shared secret. (2) Both the mobile station and the key managing computer generate a first code (MAC1) and a second code (MACT1) using the same predefined method. These codes are derived from the password (OTP). (3) The mobile station transmits the public key and the first code (MAC1) to the key-managing computer. 4. The key managing computer receives the public key and the first code (MAC1) from the mobile station. It verifies the authenticity of the registered user by comparing the first code (MAC1) with the second code (MACT1).

The RDI protocol(Boontaetae et al., 2018). involves three key players: End-users, Trusted Source Certificate Authorities (TSCAs), and Service Providers (SPs). End-users manage their identity through applications employing the RDI protocol, while TSCAs, such as banks or universities, authenticate and certify end-users. SPs authenticate end-users using the RDI protocol. End-users and TSCAs follow the same identity structure and protocol for creating, registering, and revoking identities. The protocol employs Ethereum key pairs for entity registration, with hot-keys for active operations and cold-keys for revocation. End-users can certify each other or rely on TSCAs for certification, which involves authentication and verification processes. Identity and entity revocation are conducted using respective keys, ensuring validity checks for identities and certificates. Verification processes involve two-way verification to ensure the authenticity and validity of identities, considering factors such as revocation status and certificate trustworthiness.

(Kotwal et al., 2017). discusses the identification of data streams generated by the Aadhaar system and its applications across various sectors, as well as incentives for public and private sectors to release open data. It proposes principles and an implementation framework for the responsible release of open data through Aadhaar, emphasizing the need to safeguard individual privacy while disclosing data. The privacy framework for open data highlights the limitations of traditional anonymization techniques in preventing re-identification risks and suggests approaches to minimize privacy threats. It advocates for excluding personally identifiable information (PII), such as Aadhaar numbers, from open datasets and recommends redacting identifying information or releasing aggregate statistics to mitigate re-identification risks. The paper also proposes a monitoring and enforcement framework, including creating reg-

ulations and committees to oversee the implementation of Aadhaar-related open data policies and address potential violations. The aim is to facilitate both public and private sector participation in releasing anonymized aggregate statistics to enhance research, policy-making, and public accountability within the Aadhaar ecosystem.

The Mobile eID Management System proposed(Verzeletti et al., 2018). for the Brazilian Electronic Government for a national Identity Management system that is based on interoperability standards and e-PING architecture. It incorporates technologies such as TLS, AES, X.509, SAML, XML, and JSON for secure communication, encryption, digital certificates, and data exchange. Key features include a centralized Identity Provider (IDP), FIDO UAF and secure elements on mobile devices, biometric authentication using fingerprints, pseudonym generation for user privacy, and reliance on trusted Certificate Authorities like the Brazilian Public Key Infrastructure. By leveraging UAF FIDO and secure elements in mobile devices, the system enhances security in key generation and storage, offering increased computational capabilities compared to secure elements like SIM cards. The system follows a user-centric identity management model in four phases: 1) Mobile eID registration, 2) e-Gov SP registration, 3) accessing services, 4) revocation of ID. Registration involves identity confirmation at authorized agencies, key pair generation, and association with government attribute databases, while usage requires authentication through the IdP and attribute confirmation. Revocation entails the exclusion of registered keys from the IdP and SP, restricting further authentication and document signing capabilities.

In Distributed Ledger Technology discusses applications by (Dunphy et al., 2018).on for digital identity and proposes research to explore its future potential. The focus is on understanding how DLT interacts with existing challenges in digital identity, such as security, trust, interoperability, and deployment, rather than on intrinsic DLT engineering challenges like scalability. The proposed research aims to refine the DLT properties leveraged for identity, evaluate deployment in light of PKI challenges, support secure delegation of credentials, gather new requirements for user experience, and evaluate exposure to public permission less DLTs. Key considerations include questioning how DLT properties like transparency, immutability, and decentralization manifest in digital identity, understanding the implications of decentralization in centralized systems, and assessing the sustainability and scalability of DLT-based identity schemes.

The article(Tseng et al., 2017). details a protocol for Identity-based Authenti-

cated and Key Agreement (ID-AKA) tailored for multi-server environments comprising trusted PKGs, powerful servers, and mobile clients. The protocol consists of three phases: 1) Setup, 2) Key extract, and 3) Authenticated key agreement (AKA). The PKG generates system parameters in the setup phase, including a master key, public key, and hash functions. The key extract phase involves the PKG generating and distributing private keys to clients and servers. Participants' private key is derived from their identity and a random number. The authenticated key agreement phase ensures secure communication between clients and servers. The protocol utilizes ephemeral secrets and private keys to compute session keys, providing client-to-server and server-to-client authentication, along with the key agreement. The security analysis demonstrates that the protocol resists attacks under the random oracle model.

This research paper (Chandrashekhara et al., 2021). provides a thorough examination of Digital Signatures and their advantages in addressing these challenges. The Digital Signature Concept is crucial for secure transactions over open networks, ensuring data integrity and authenticating the identity of senders. These techniques are integral to cryptography protocols, offering entity authentication and key agreement services. With the increasing use of mobile devices for internet access, the risk of unauthorized access to critical documents is escalating. However, current implementations of Digital Signatures are not entirely effective. While symmetric data transfer mechanisms are commonly used, there's a need for a more robust mechanism for secure document transfer and verification.

(Husni, 2016). presents a method for creating digital signatures that mimic physical signatures in official agreements, enabling parties to work with digital documents as they would with physical ones. The approach integrates a signing service into users' national identities using cloud computing and mobile devices. The operation of the signing cloud involves three main categories: registration, basic function, and main function. Registration involves inputting user information and key pairs, while the basic function includes confirmation processes and data signing. The main function extends these capabilities to provide Mobile ID services for user authentication and document signing. Implementation involves building a web application with Apache, PHP, PostgreSQL, and OpenSSL for the backend, and a mobile application using Google Android SDK and Google Play Services. Mobile ID offers advantages in user authentication and document signing processes, providing a modular and expandable solution for various purposes through HTTP API communication.

The system model (Nishimura et al., 2018). focuses on ensuring the identity of device owners and securely sharing keys between devices. Problems in pairing two devices arise due to potential key theft through careless pairing methods or intentional actions by users. To address this, a Trusted Third Party called the Owner Identification Service (OIS) is introduced to supervise key-sharing and issue owner certificates. These certificates, conforming to the X.509 format, contain unique identifiers associated with individuals and are protected in the secure world. The OIS, often operated by mobile network operators or government PKI systems, ensures high identity assurance. Security threats, including network and malware attackers, are considered in the key-sharing process. A peer-to-peer key-sharing approach is proposed to mitigate these threats, utilizing secure channels established through proximity communication technologies like NFC or BLE. Detailed procedures for securely copying keys between devices involve encryption, decryption, and mutual authentication based on owner certificates, ensuring keys are safeguarded throughout the process.

The proposed protocol by (Jia et al., 2019).on identity-based anonymous authentication scheme protocol. The protocol introduces three main entities: the Trusted RC (Registration Center), mobile users, and MEC (Mobile Edge Computing) servers. Mobile users and MEC servers must register with the RC to access system services. The RC issues long-term secret keys based on their identities. Mutual authentication occurs directly between mobile users and MEC servers without involving the RC. Security requirements include mutual authentication, session key agreement, user anonymity, untraceable, perfect forward secrecy, Single Sign-On (SSO) functionality, no online RC dependency, and resistance against various attacks. The protocol's security model is based on a game between a challenger and an adversary, with interactions and oracles reflecting various functionalities used to determine protocol security.

Critically evaluates of three cryptography schemes proposed by (Wang et al., 2016). In their analysis, they uncovered significant security vulnerabilities in each scheme. They also presented the AINA'12 scheme, which fails to resist known session-specific temporary information attacks, key compromise impersonation attacks, and poor usability. Li et al.'s privacy-preserving scheme, from GLOBECOM'12, suffers from severe efficiency problems, rendering it impractical for real-world use. Despite its purported security guarantees, Zhang et al.'s "provably secure" roaming services scheme, introduced at SCN'15, is susceptible to collusion and replay attacks. The

authors emphasize that merely patching protocols to resist known attacks without addressing fundamental design flaws does not guarantee robustness. They argue that providing formal proof of security, as in Zhang et al.'s case, is not a foolproof solution and that understanding potential threats during protocol design is crucial. The paper proposes improvements and addresses identified vulnerabilities while maintaining reasonable efficiency and usability. However, they find no straightforward fixes for the issues in Li et al.'s and Zhang et al.'s schemes. Ultimately, the authors suggest that their enhanced scheme offers a more promising solution for practical applications, as it effectively mitigates security loopholes without imposing excessive costs.

Identity-Based Signature with Server-Aided Verification (IBS-SAV) scheme by (Ramadan et al., 2020). aiming to enhance security in 5G mobile systems while minimizing computational costs. By combining IBS (identity-based signature) and SAV (server-aided verification) techniques, the scheme achieves efficiency and robustness against various attacks. It outlines two mobility modes: home state and roaming state, each addressing users within the same or different RANS (Radio Access Networks). The scheme's construction involves setup, key generation, user signing, and verification processes, ensuring correctness and security under the Computational Diffie-Hellman (CDH) assumption. Mathematical proofs validate the scheme's security against existential forgery and collusion attacks, making it suitable for 5G applications.

(El Haddouti & El Kettani, 2019). examines and compares three popular identity management Systems utilizing Blockchain technology: uPort, Sovrin, and ShoCard. Through an analysis of their features, the paper aims to assist readers in selecting the most suitable system for specific scenarios. Blockchain, known for its decentralized nature, has garnered attention in identity Management due to its potential for secure and transparent record-keeping. Each system offers distinct advantages, such as self-sovereign identity control and decentralized identity verification. However, challenges remain, including a lack of focus on user experience, unclear data protection measures, and privacy concerns. Addressing these issues is crucial for establishing a consistent, privacy-preserving approach to Identity Management in Blockchain applications.

Sovereign Digital Identity (NexGenID) architecture discussed by (Toth & Anderson-Priddy, 2019). is designed to uphold the self-sovereignty of digital identities through a comprehensive set of functions and mechanisms. It addresses self-sovereignty properties to allow the user to access their identities, which are safeguarded within per-

sonal devices running identity engines. These identities can be securely transferred, presented, registered, and verified, enabling users to manage them effectively. The architecture supports interoperability across applications and services, facilitating collaboration between users. User interfaces are designed to be user-friendly, emulating the methods used in the actual world to acquire and issue IDs. Counterfeit prevention is ensured through public/private key cryptography, while synchronous and asynchronous identity verification methods enable reliable collaboration. Third-party proofing and attestation assure identity integrity, and secure identity transfer and transactions are facilitated using encryption and digital seals. Overall, the NexGenID architecture demonstrates feasibility in satisfying the properties of self-sovereignty for digital identities.

(Naik & Jenkins, 2016). discusses the necessity of evaluating IAM (Identity and Access Management) standards for Mobile Computing and proposes several criteria for effective evaluation. These criteria include extensive support for authentication and authorization across various cloud delivery and service models, user-friendly SSO functionality, lightweight standards for mobile applications, platform independence, scalability, and support for both web and native mobile apps. Additionally, the criteria emphasize the importance of support for different business models, immediate revocation of access, end-user authorization, confidentiality, verification, and data integrity. The study emphasizes the distinctive obstacles encountered in Mobile Computing, such vulnerable wireless connection and the extensive utilization of mobile devices, which require specific deliberations in IAM standards.

(Böger et al., 2014). discusses the implementation of SecFuNet identity management, which focuses on the user-controlled release of attributes stored in smart cards for enhanced security. It utilizes OpenID protocols to establish trusting relationships among users, identity providers, and service providers, aiming to address security risks associated with traditional username/password authentication and phishing attacks. In the proposed model, user attributes are stored in smart cards rather than Identity Providers' databases, giving users control over attribute release during computing sessions. The authentication model relies on secure elements and virtualization, establishing a mutually authenticated secure channel between smart cards and SecFuNet Identity Providers. A prototype implementation is described, highlighting components such as the SecFuNet Identity Providers, Authentication Server, Identity Selector, and Smart Card. Experimental results demonstrate the viability of

the infrastructure, though performance optimizations are ongoing. Future work includes extending protocols for federated identity management to establish trusting relationships among SecFuNet identity providers.

2.2.2 Works Related to On-device Key Generation

The scheme proposed by (Albakri et al., 2019). is based on polynomial key management with blockchain, tailored explicitly for the Hyperledger Fabric model. The scheme utilizes three bivariate polynomials to facilitate different types of communication within the blockchain network. These polynomials are tailored for two-way communication between client applications and all peers in the network. The key management scheme supports general and additional communication flows, with two main phases: token generation and key establishment. During the token generation phase, the Key Generation Center (KGC) creates tokens using bivariate polynomials for all entities in the network. In the key establishment phase, entities utilize these tokens to compute shared pairwise keys for secure communication. The scheme enables secure communication between client applications and endorsing peers, even allowing for communication between different entities. Security analysis indicates that the scheme enhances security using three independent bivariate polynomials, making it resilient against insider and outsider attacks. Measures such as tamper-proof mechanisms can further bolster security by securely storing tokens within entities, mitigating the impact of potential breaches.

The proposed authentication protocol (Wu et al., 2018). between mobile client (app) and server environments is designed for efficient privacy preservation. The system generates two key, zero-knowledge, and Paillier homomorphic encryption (HE) proofs in three distinct phases: Initialization, registration, and mutual authentication of the system. During System Initialization, the server generates system parameters and its public-private key pair. In the Registration phase, users register with the server using two devices, generating key shares for each device. In the Mutual Authentication phase, users communicate with the server using both devices, exchanging encrypted messages and proofs to establish authenticity. The protocol utilizes homomorphic properties to ensure security during message exchanges. Additionally, it incorporates zero-knowledge proofs for verifying cryptography relations. The security analysis proves the protocol's resistance against various attacks, including those

involving corrupted devices, by demonstrating the computational problems' hardness. The protocol achieves mutual authentication security under the assumption of hard computational problems like the CDH problem.

(Gong et al., 2020).proposed the creation of a SK (secret key) and a PK (private key) at the same time, using three authorized terminals and a trusted helper. This is done even when a passive eavesdropper can listen to all public conversations. Through a Distributed Measurement System (DMS), the terminals observe correlated source sequences and aim to generate keys securely. The system model is characterized by a joint probability mass function for observations and imposes secrecy and uniformity conditions on the generated keys for security. The paper defines the key capacity region and provides achievable schemes for generating SK and PK pairs, considering various scenarios and complexities introduced by the presence of a trusted helper. Furthermore, the study examines the security performance regarding key leakage rates and emphasizes the proposed model's relevance to practical scenarios such as IoT, car networks, and D2D networks.

In SaPKI (Server-aided PKI) designed by (Cai et al., 2005) to alleviate computational burdens from mobile clients in networks like GSM and CDMA. SaPKI enables mobile devices to leverage a powerful server for tasks such as RSA key generation and signing. SaPKI is typically deployed in the infrastructure of mobile service providers, offering efficient key generation and signing capabilities to multiple clients. The architecture allows for the amortization of costs over many client machines, promoting scalability and cost-effectiveness. SaPKI provides interfaces for key generation, certificate initialization, and signature generation, streamlining cryptography operations for mobile clients. Implementation details include using "Modadugu's" key generation protocol and the RSA algorithm with exponent 3 for digital signatures. The architecture is well-suited for cell phone networks, where communication with fixed infrastructure is necessary for each call. By offloading cryptography tasks to servers, SaPKI enhances the performance and security of mobile applications like cell phone banking within existing network environments.

(Zhang et al., 2019). Introduced a novel key protection framework KPaM, tailored for mobile devices, aiming to address existing security issues efficiently. KPaM utilizes a two-party computation algorithm, dividing the private key (PK) into two parts and stored in the device. Distributing key components minimizes the risk of complete key exposure. The mobile device employs a TEE instead of a SE, ensuring

secure storage and access control mechanisms. The design goals of KPaM prioritize tolerance to key loss, extensibility across different cryptography algorithms, and maintaining acceptable efficiency. The system architecture comprises three main components: TEE for local key management , Rich OS serving as a bridge, and the cloud for cryptography computations and storage. Secure channels are established between the mobile device and the cloud using lightweight SSL protocols. The framework's security evaluation demonstrates resilience against various attack scenarios, including malware in TEE or Rich OS, communication channel attacks, and cloud compromise. Additionally, efficiency evaluations compare KPaM with other key protection methods, highlighting its effectiveness despite slight performance overhead. KPaM offers a robust, adaptable, and efficient solution for safeguarding private keys on mobile devices. It demonstrates versatility by supporting various cryptography algorithms and shows promising feasibility and efficiency in experimental implementations.

(Chudamani & Tatini, 2019).introduced a method to enhance data transmission security through a combination of encryption and steganography techniques. It begins with encrypting the message using the affine cipher algorithm, generating ciphertext. This ciphertext is then inserted in the image and encrypted using a key. The key is derived from a digital image matrix, and the embedded data and key are transmitted. The key is derived from the initial transmitted image at the receiver's end, and the ciphertext is extracted from the second image for the purpose of decryption. The affine cipher algorithm maps characters to numerical values, encodes them, and rearranges them into binary codes. Data embedding employs the least significant bit (LSB) methodology ,while data extraction retrieves LSBs from pixel values in the image. This approach merges cryptography and steganography effectively, ensuring improved security in transmission with minimal image distortion.

(Batina et al., 2003). presents an overview of hardware implementations of RSA and ECC algorithms based on modular arithmetic. It dives into the mathematical underpinnings and techniques required to build these cryptosystems before going into the various hardware architectures proposed in the literature. Over the years, significant progress has been made in the hardware implementation of public key cryptography, largely due to advancements in VLSI technology and improved algorithms. Cryptography hardware accelerator modules have become commonplace in applications like VPN, e-commerce, and bank transactions, even finding their way into smart card co-processors. However, while there have been advancements in hardware implemen-

tations, challenges remain, particularly in creating small and energy-efficient versions appropriate for personal devices and environments that support intelligent systems. Additionally, there is a need for efficient implementations that can withstand sophisticated side-channel attacks. The paper concludes that while fundamental hardware concepts from the 1980s, such as systolic array and RNS realizations, remain relevant, there is still room for improvement, especially in ECC hardware implementations.

secret-key generation using a compound source by (Tavangaran et al., 2018). where participants do not know the exact statistics of the source but assume it belongs to a known compound set. The scheme aims to ensure both reliability and security of the generated secret key (SK) and each part of the compound set while also considering an eavesdropper's side information and adhering to a public communication rate constraint. The SK capacity is expanded to include arbitrary compound sources with a finite set of marginals, defined as a function of the forward communication rate between legal users. The study focuses on situations where the size of the compound set can be any value, and the set of marginals is limited in number. It accurately determines the SK capacity by considering the communication rate constraint. It is noted that if the set of marginals is also infinite, the SK rate may be lower due to potential estimation errors by Alice, leading to different compound sets and decoding results. An extra regularity condition is imposed on infinite compound sources in the quantum regime to derive the multi-letter secret-key capacity in scenarios without communication rate constraints.

(Lai & Ho, 2012). introduced a novel approach to key distribution among terminals using secure routing, linking the problem to the multi-commodity in graph theory problem. Using the "Max Bi-Flow Min Cut" Theorem and building a matching outer-bound, it is proved that the proposed strategy achieves the key capacity region for establishing two keys. The achievable sum rate approaches a constant factor to an upper bound for scenarios requiring more than two keys. The proposed protocol provides secure routing key propagation and transforms the key agreement problem into a multi-commodity flow problem over the constructed graph. This equivalence optimizes achievable key rates by strategically selecting routes and flow amounts. Considering the PIN model, a secure routing-based key propagation protocol is proposed, effectively converting the problem into a multi-commodity flow problem. The paper demonstrates the optimality of the proposed approach for generating two keys and its proximity to an upper bound for the general case of multiple keys.

In the field of key generating (Zhang et al., 2017). explores the problem of generating multiple keys simultaneously in a cellular source model with the assistance of a helper. Four models are considered, differing in secrecy requirements and whether symmetric or asymmetric key generation is employed. The models include terminals X0, X1, X2, and X3, with X0 aiming to generate two different types of keys (K1 and K2), respectively, with X1 and X2 assisted by X3. Achieving secrecy from eavesdroppers and, in some cases, from the helper itself is crucial. The research determines the optimal capacity region for each model by developing a unified technique consistent with cut-set outer bounds, thereby simplifying the process. The paper extends the study to scenarios where more than two symmetric keys must be established simultaneously between multiple mobile terminals and a base station. It provides a theorem characterizing the key capacity region for this model, addressing the secrecy constraints imposed by an eavesdropper. The proof uses the Ordered Set Rate Binning (OSRB) method and the submodularity of some functions to match achievable regions with cut-set outer bounds.

2.2.3 Works Related to Mobile-based Signing Solutions

PKI Infrastructure for wireless devices is crucial. PKI (Selvakumaraswamy & Govindaswamy, 2016). involves procedures for creating, distributing, and revoking digital certificates and binding public keys with user identities through a CA. Third-party validation authorities (VAs) are responsible for ensuring distinct user identities within each CA domain. The registration and issuance process, conducted by software or human supervision at a CA, establishes this binding. ECC is considered optimal for resource-constrained applications compared to RSA. With quicker verification, ECC-based signatures and certificates are smaller and faster to create, leveraging the EC Discrete Logarithm Problem for enhanced security and computational efficiency.

One Time Mobile PKI Originated system (Saquib et al., 2017). utilizes a block architectural model for data signing and verification. Initially, requests by SP to mobile device key pair generation, followed by a Certificate CSR sent to the CA to obtain a valid digital certificate. Based on the SP's request, sign data on a mobile device and send to CA for verification. The authentication server authenticates the signed data, and upon receiving the response, the SP acknowledges it accordingly. After completing the transaction ,the mobile device erases the keys and certificate.

The signal model depicts the process of exchanging data involving the mobile device, SP, CA, and authentication server. A mathematical model involving RSA key pair generation and X509 certificate creation ensures short-term validity. Verification involves comparing signed message digests, with the CA validating the signature and responding accordingly to the SP, which then acknowledges the mobile user.

(Theuermann et al., 2019). outlines a solution for generating a remote QES (Qualified Electronic Signature) using a mobile device as the sole authentication method. The architecture includes various components such as the SP, Signer Interaction Component (SIC), Secure Element (SE), Server Signing Application (SSA), Signature Activation Module (SAM), and SCDev. These components ensure a secure connection between the device and server. Authentication involves multiple factors, including knowledge and possession, and cryptography techniques are used to ensure security. The implementation includes authentication protocols for secure communication between the SP, SIC app, and remote SSA. Communication interfaces are established using JSON elements over HTTP and URL schemes for inter-process communication (IPC). This solution provides a flexible and secure method for remote signature generation using mobile devices, fulfilling stringent security requirements and regulatory standards.

(Rumbao et al., 2011). outlines mechanisms for achieving electronic security on mobile devices through digital signatures and certification. The platform development focuses on implementing a digital signature mechanism on a Nokia N95 using Java for broad compatibility. Components of the platform include the mobile phone, a conventional certificate from a national authority, a Tomcat server, and NetBeans IDE for development. On the client side, the MozartPKI MIDlet enables users to digitally sign messages using their private key, with installation facilitated via Bluetooth. Signed messages are sent to the ServerPKI Java Servlet on the server side for processing, and verification tools like eSign Viewer ensure the integrity and authenticity of the signed data, ultimately enhancing digital security in mobile environments.

To address these challenges of different authentication methods by (Thant & Zaw, 2018). in IOV (Internet of Vehicles) are examined. These include proxy mobile IP and cryptography hash functions. It delves into authentication based on PKI, detailing the role of certification authorities (CAs) in verifying the authenticity of public keys. Additionally, it explores authentication via Identity-Based Encryption (IBE) systems, which derive public keys from unique identifiers without requiring

prior coordination. The drawbacks of IBE systems, such as the risk of compromised private key generators (PKGs) and the absence of non-repudiation, are discussed. The paper provides valuable insights into enhancing authentication and privacy in IoV communications through various cryptography mechanisms.

Mobile authentication protocol designed by (Bicakci et al., 2014). to counter Man In The Middle (MITM) attacks without user intervention. Leveraging the secure element on mobile devices, Mobile-ID signs messages incorporating context information from the service provider, enabling the Mobile-ID server to detect ongoing attacks and notify genuine service providers. Protocol specifications outline assumptions about the secure operating environment of mobile devices and the roles of involved parties such as users, mobile terminals, secure elements, service providers, and the mobile-ID server. Users prepare by obtaining a secure element with digital signature capability and installing a mobile browser plug-in for communication with the Mobile-ID server and the secure element. The protocol involves selecting a service provider, establishing secure connections, signing a nonce value, appending context information, and verifying signatures, effectively preventing MITM attacks by comparing context information.

In this study, author (AlMajed & AlMogren, 2020). goal was to come up with a strong method for authenticated encryption (AE) by better mapping phase of plain text to an EC curve. This way, attacks like the Chosen Plain text Attack and the Chosen Ciphertext Attack would not work. The proposed scheme underwent evaluation and analysis regarding security requirements, and comparisons were made with other schemes to assess security characteristics and performance metrics. The scheme has nine steps: setting up the system parameters, turning plain text into numbers, mapping points on the EC, encrypting the mapped points, and signing the combined points. It ensures that communication is safe and can be used for IoT and industrial IoT applications. The study's main contribution lies in providing a secure and efficient encryption scheme using ECC, particularly emphasizing the creation of a shared key for parties involved in group communication, which is often overlooked in previous studies. The proposed scheme enhances security by focusing on encoding plain text to EC and addressing flaws vulnerable to CPA and CCA attacks. An analysis of the scheme's security showed that it was resistant to certain encryption attacks, and an analysis of its performance showed that it had about the same amount of computational overhead as other schemes but was safer without sacrificing efficiency.

Mobile Home Agent (MHA) based on Mobile-PKI system introduced by (Ray & Biswas, 2011). to provide security similar to wired PKI for mobile phones. It utilizes an MHA and Registration Authority (RA) to streamline processes and minimize workload. The proposed system's Wireless Certificate Management Procedure involves certificate request generation, verification, and issuance. Symbols and definitions are established, outlining operations like hash and signature functions. Certificate life cycle management, validation procedures, and application scenarios are detailed, emphasizing secure communication between mobile devices and servers. Security analysis highlights features such as mutual authentication, Proof of Possession (POP) verification, confidentiality, and prevention of replay attacks, ensuring robust security measures for mobile PKI.

Wireless Public Key Infrastructure (WPKI) model designed by (Lee et al., 2007). for mobile phones, addressing challenges in applying wired PKI technology to mobile devices. This WPKI model features a two-level hierarchical architecture for certificate management optimization and verification of mobile phones and servers. It minimizes data sizes processed on mobile phones, reducing module sizes to be installed on the device while maintaining security levels equivalent to wired PKI systems. The model utilizes X.509 certificates for mobile phones, enabling certificate verification by servers without burdening the mobile device. It adopts the ECDSA algorithm to reduce computational complexity on low-performance mobile devices and implements short-lived certificates and OCSP for efficient certificate validation. The proposed WPKI technology is not limited to M-commerce. Still, it can also be applied to various wireless communication scenarios, such as mobile hospitals and government services, leveraging wireless internet connectivity on mobile phones.

(Otterbein et al., 2017). introduces an approach to implementing the German eID concept on Android devices without needing physical identity cards and card readers. Despite identifying two non-critical vulnerabilities in the architecture, the security evaluation assures no sensitive information is compromised. The design section outlines the initialization process, involving the installation of security domains and applets on the smartphone, facilitating communication between different entities like the issuer, TSM, and service provider. The authentication procedures are detailed, emphasizing secure connections, PIN entry, and chip authentication for user verification with the eID offered. The technical evaluation discusses requirements and existing solutions for hardware components, such as ARM TrustZone and secure

elements, alongside the eID Android application. However, limitations are acknowledged, particularly regarding access to TrustZone for third-party developers and the lack of support for embedded secure elements on Android. Nonetheless, the proof of concept demonstrates compatibility with the PersoApp, suggesting potential solutions and avenues for future research to enhance the security and implementation of mobile eID on Android devices.

(Aminuddin, 2020). proposed a protect Android app asset data using a cryptography algorithm. Using Android Studio, the assets are encrypted at compilation time, and the Android device's application runtime decrypts them afterward. The proposed algorithm combines the RSA asymmetric algorithm with the AES symmetric algorithm, which offers the best security for protecting Android application assets. The methodology involves several steps, including encryption during compile-time and decryption during run-time, utilizing various cryptography techniques such as key generation, hashing, and XOR operations. Performance evaluation reveals that the proposed RSA and AES algorithms outperform alternatives like RSA and DES, achieving encryption speeds of 106.82 MB and decryption speeds of 44.42 MB per second. The proposed method enhances asset protection and performance for users and developers by ensuring application security and faster operation.

2.2.4 Works Related to Trusted and Secure Storage

(Cooijmans et al., 2014). examines the security provided by several methods for secure key storage on Android, utilizing either Android's key storage service. The Key Storage Test application was developed to assess key stores on Android devices, initially checking if cryptography algorithms like RSA, DSA, and ECDSA are truly bound to the device. The application allows the generation and deletion of key pairs within the keystore under designated aliases. Testing involves two instances of the application, with one generating a key pair and the other controlling it to produce valid signatures. This process is repeated for various attacker models with escalating privileges. Results indicate that while Bouncy Castle's default format protects against tampering, it doesn't guard against inspection, as certificates stored in the keystore can be accessed. Ensuring keystore integrity requires a password during storage and verification. The application must provide the password to utilize the stored keys. Access to the Keystore file is restricted to the application-specific data directory, pre-

venting unauthorized access by other apps without root permissions. However, with root permissions, access to the keystore file becomes possible, enabling the creation of valid signatures using the private key. Bouncy Castle's user-provided password for keystore storage offers better security, but interception or brute-forcing by a root attacker remains a concern. AndroidKeyStore implementations vary across device manufacturers, with examples of Qualcomm and Texas Instruments explored. While access to private keys may be restricted, root attackers can gain control over keys stored on the same device, compromising device-binding requirements. Furthermore, software fallback scenarios raise additional security concerns, particularly regarding encryption and protection against root attackers.

MicroTEE (Ji et al., 2019) introduced a TEE operating system on the microkernel architecture, aiming to enhance security by isolating TEE OS services. MicroTEE utilizes a microkernel to provide core services like space management, communication, cryptography, and key management, which are implemented as user-layer applications. This design prevents the compromise of the TEE if any one service is vulnerable. A monitor facilitates the switch between the secure and normal worlds. The prototype was implemented on the i.MX6Q Sabre Lite board demonstrates improved cryptography performance compared to Linux, particularly with small data sizes. MicroTEE's design, implementation, and evaluation emphasize its stability and efficiency, providing basic cryptography services to trusted applications (TAs) while maintaining robust security measures.

Android storage models (Kamal et al., 2023). explores of security vulnerabilities and proposes TEE as a solution, focusing on a novel approach to mobile ID using TrustZone. It outlines a design analysis considering system vulnerabilities and presents the proposed mobile ID solution, which involves user registration, demographic data collection, and verification through an Identity Issuing Authority (IIA). A prototype implementation on the Android simulator is discussed, with primary results indicating stable throughput, albeit slightly decreasing as file size increases. The paper suggests secure mobile device storage elements like keys and certificates for sensitive data. It emphasizes the advantages of mobile ID in terms of cost-effectiveness, user friendly, and ease of use by agencies and users for digital transformation. Mobile ID solution deployment is portrayed as facilitating electronic services for citizens, encompassing various aspects such as user registration, credential issuance, authentication, ID management, authorization, encryption, and signature in different scenarios.

software based framework (SofTEE) introduced by (Lee & Park, 2020). to provide TEE-based applications, safeguarding them from potential attackers, including compromised kernels. SofTEE achieves this through kernel deprivation, which delegates privileged operations like memory management to a security monitor, minimizing the kernel's exposure to security threats. SofTEE efficiently manages the ASID (address space identifier) to optimize performance, reducing switching overhead between the normal kernel and the security monitor. The other three deal with the connection within the security monitor-managed address space. These provide guaranteed confidentiality and integrity for security-sensitive applications. The framework is a Raspberry Pi 3 board prototype, demonstrating its viability in real-world scenarios. Performance evaluation reveals a modest overhead of about 3 security-sensitive applications with longer execution times and 23 for those with shorter execution times. Unlike previous solutions, SofTEE doesn't rely on specialized hardware like Intel SGX or ARM, making it adaptable to various environments while requiring hardware support such as root-of-trust and random entropy.

The novel approach for mobile ID solutions proposed by (Kamal et al., 2022). with PKI technology, eliminating the need for physical ID cards or card readers. It prioritizes security, privacy, and usability to protect PII data on handheld devices. The solution aims for security, cost-efficiency, and privacy compared to current scenarios, establishing secure identities among users, authorities, and SP in both the public and private sectors. Key aspects include registration with an IIA, biometric-like face recognition for user authentication, and key pair generation using the ECC algorithm. The approach ensures online, secure, and auditable transactions, storing sensitive information securely within the device's TEE.

(Enck et al., 2009). explores the complexities of Android security and identifies refinements made by Google developers, which enhance convenience but complicate overall security comprehension. Despite these refinements, holistic security concerns remain largely unaddressed, such as interpreting permission labels and controlling access to them. To tackle these issues, Kirin is developed as a security framework to assess an application's compliance with phone-wide security policies. Kirin utilizes a formalized model to generate automated proofs of compliance, reducing the need for user intervention during installation. Using Kirin, the authors identified vulnerabilities in base Android applications and collaborated with Google to rectify these flaws, contributing to ongoing research into Android security. While Android offers

comprehensive security features, navigating its intricacies requires effort, and tools like Kirin are essential for ensuring its evolution into a secure operating system for future computing platforms.

(Yalew et al., 2017). introduced T2DROID, an Android dynamic analyzer, to detect malware by analyzing traces of API functions and kernel system calls. Leveraging the ARM TrustZone security in T2DROID, runtime analysis within the secure world is conducted to protect against malware. By using API function calls and system calls together, T2DROID makes it possible to fully observe operations with clear semantics, improving the ability to detect them. Machine learning classifiers, such as kNN, enable T2DROID to detect malware without manual rule development, facilitating easy reconfiguration for new threats. Experimental evaluations demonstrate high accuracy and precision of 0.98 and 0.99, respectively, with a low false positive rate. Performance overhead evaluations reveal minimal impact on system resources, with integrity verification operations taking slightly above 1 second. The ROC curve analysis shows that T2DROID works better when using the API call and syscall features than individual feature sets. This provesit is a reliable way to find malware on Android devices.

Secure enclave architecture (SecTEE) designed by (Zhao et al., 2019). to provide robust security comparable to hardware-based solutions without specialized security hardware. It offers essential trusted computing primitives, including integrity measurement, data sealing, remote attestation and secrets provisioning. SecTEE utilizes the CPU's isolation mechanism, such as ARM TrustZone, making it adaptable to various CPU architectures. It ensures integrity through signature verification and measurement, enabling remote attestation for verifying enclave states. While TEE OS rollback attacks are not addressed, SecTEE suggests countermeasures like hardware monotonic counters. Modifications to the kernel's memory management ensure protection against memory access-based side-channel attacks. Evaluation results show acceptable performance overhead, primarily from memory protection mechanisms. Overall, SecTEE offers a promising approach to secure enclaves, focusing on security and performance.

Digital trust technologies examining by (Shepherd et al., 2016). their role in advancing secure computing and their application in emerging domains such as the IoT and cyber security systems. Various technologies like the Trusted Platform Module (TPM), SE, TEE, GlobalPlatform, and Intel SGX have been analyzed. The threat

model for IoT and CPS includes both on-device and off-device attackers. This led to the creation of evaluation criteria such as user and centralized control, access management, verification methods, hardware security, storage protection, execution isolation, and defense against attacker changes. Despite the evolution of trust technologies, no single technology meets all criteria; however, designers can combine multiple technologies to achieve desired security levels. The analysis serves as a roadmap for future research, identifying deficiencies in safe and trustworthy computing and suggesting potential areas for additional investigation in IoT and CPS security.

mHealth Apps Security Framework (MASF) (Hussain et al., 2018). introduced a conceptual framework designed to enhance the security and privacy to protect sensitive health data for medical data within Android mHealth applications. Three distinct phases are outlined: design, implementation, and evaluation. The framework is designed in Phase I, comprising the SML (Security Module Layer) and SIL (System Interface Layer) to address security requirements. Phase II involves implementing MASF by programming and integrating it into the Android OS, followed by deployment onto real devices. Finally, Phase III evaluates MASF's effectiveness and efficiency through testing against various attacks and measuring performance metrics. The article also provides background information on Android platform architecture and security models, highlighting the intersection of mHealth apps and security issues. The methodology adopted for the research study is outlined, emphasizing the importance of literature review, framework design, prototype implementation, and evaluation. The suggested framework is intended to guide the security of mHealth applications on Android platforms, with future plans for simulation and implementation.

(Lambrinoudakis et al., 2003). showcases the utilization of Public Key Infrastructure (PKI) to address the security needs of an integrated e-government platform. It employs an organizational framework to categorize e-government services based on their security requirements, which is then applied to the case study of the 'Webocrat' system to design a PKI-based security architecture. The complexity of designing security mechanisms for such a platform is highlighted, indicating that traditional Risk Assessment (RA) methodologies are not directly applicable due to the diverse and interconnected nature of e-government systems. The paper uses the 'e-GOV-OFSR' architecture to thoroughly compile security needs relevant to the entire e-government platform. These requirements are organized based on service stages and actor cat-

egories. A uniform security policy can be developed for e-government by selecting security measures aligned with these requirements. It has also been shown that PKI security services can effectively meet most of the identified e-government security requirements. Extra steps are only needed for infrastructure-related issues or very specific security-critical aspects.

2.3 Other Works ECC Related

ECC explores by (Prabu & Shanmugalakshmi, 2010). and highlighting its advantages over other cryptography algorithms RSA. ECC algorithms offer higher security with less key sizes and faster cryptography operations with less consumption power. The performance of ECC is particularly notable as its inverse operation becomes harder and faster with increasing key length compared to other algorithms like Diffie Hellman and RSA. The paper discusses ECC's security aspects and future enhancements, emphasizing its effectiveness in providing security while maintaining efficiency. Additionally, the application of ECC, specifically in the context of ECDSA algorithm, is used for key generation, signature, and verification processes.

In the proposed framework for (Khan et al., 2020). patient authentication with an activation sensor device transmits sensor values to the cloud server.Biometric information is included alongside the username and password for authentication, utilizing the SHA-512 algorithm for integrity. Two encryption methods, substitution seaser cipher and IECC (improved ECC algorithm), secure sensor data transmission, with IECC employing an additional secret key for heightened security. The computational cost of the scheme is (4H + Ec + Dc), which is lower than existing schemes, with an average correlation coefficient value close to zero, demonstrating algorithm strength. Encryption and decryption times are measured at 1.032 s and 1.004 s, respectively, and performance is evaluated against existing RSA and ECC algorithms.

(Singh et al., 2020). introduces a smart card authentication scheme that works on ECC encryption, providing multiple security attributes such as confidentiality, non-repudiation, integrity, mutual authentication, anonymity, availability, and forward security. It is shown through security analysis that the protocol effectively mitigates various attacks, including password guessing, impersonation, replay, desynchronization, insider, known key, and MITM attacks. The proposed ECSSP-SC protocol focuses on achieving mutual authentication in a multi-server environment. Still, it

can also be adapted for the machine-to-machine and user-to-user communication with slight modifications. It is divided into setup, registration, mutual authentication, and the update process. Entities involved include the user with a smart card, a server providing services, and a registration center. In the encryption-based mutual authentication phase, users and servers authenticate each other over an open communication channel, leveraging elliptic curve-based encryption for security and performance benefits. When compared to other smart card security systems, ECSSP-SC effectively meets security requirements and defends against different attacks while lowering computing and communication costs by a large amount. The protocol's combination of elliptic curve and signcryption proves effective in securing smart card systems, offering the required security functions with minimal computational and communication costs. Overall, ECSSP-SC outperforms existing security, efficiency, and resource utilization protocols, making it a promising solution for smart card authentication in diverse environments.

(Wang et al., 2006). disagrees with the idea that public key schemes can not be used in sensor networks by showing how a public-key access control system using ECC was set up on the TelosB sensor network platform. The implementation is evaluated for performance compared to other implementations on TelosB, focusing on optimization efforts due to limited processor resources. The ECC cryptosystem is implemented on TelosB mote powered by MSP430 microcontroller, emphasizing computation optimization for ECC operations. The ECC-based scheme is more efficient than symmetric-key-based authentication, as shown by measurements of power use, execution time, and code size used for performance evaluation. Despite the longer authentication time of ECC, the paper concludes that public-key cryptography is feasible for sensor network applications, paving the way for more sophisticated access control schemes to address security concerns.

(Gupta et al., 2019). presents a novel method to enhance OTP security by introducing a two-way authentication process involving user-provided and server-generated codes. This approach divides the OTP into two segments: two digits known to the user in advance and four digits randomly generated by the server. Users can create numerous unique combinations by combining these two segments in various ways, significantly bolstering security. When a user initiates a transaction through a web browser, the web server sends OTP to the user's registered mobile number for identity verification. The user then inputs the OTP into the browser window to validate

the transaction, after which the web server verifies the digits and establishes a secure connection. The proposed model offers a two-layer security mechanism, leveraging user-known secret digits and server-generated random digits, enhancing security while maintaining user-friendliness.

Identity-based authentication and access control protocol for wireless network sensors proposed by (Al-Mahmud & Morogan, 2012). for user , utilizing the IBS (Identity-Based Signature) scheme with ECC signature algorithms. This protocol facilitates user registration, authentication, session keys, data access, and revocation. During system initialization, the base station (BS) generates master secret keys, registers users and sensor nodes, and broadcasts registration information. User authentication and session key establishment involve a key exchange protocol, where users sign authentication request messages, and sensor nodes verify them based on registration history, timestamp checking, and signature verification. Data access is regulated based on user privileges, and user revocation is managed through the expiration of access time and identification of compromised users. The protocol's key contributions include its authentication and access control mechanisms, leveraging IBS for security, and simulation-based evaluation demonstrating energy efficiency and security. Future work includes standardizing the protocol's modules and implementing them in real-world scenarios to assess their practical impact on security, energy consumption, efficiency, and durability in wireless networks.

(Kavitha et al., 2019). proposed a framework addressing security issues using a hyper EC-based public key cryptosystem, integrating the Signing algorithm and 'Elgamal' approaches for authentication and secure group communication. The authentication algorithm employs hyper EC and a mapping function to generate and verify digital signatures. The process involves selecting private and public keys, generating signatures, and verifying them using the curve points. Group key construction utilizes a decentralized approach, ensuring equal privilege among members and eliminating a single failure. The proposed method facilitates secure group communication by combining HcDSA and HcElg algorithms. Performance analysis compares HcDSA with ECDSA in terms of efficiency and security, demonstrating the proposed approach in IoT healthcare systems is more effective. The study concludes that the proposed methodology meets security requirements for sensitive IoT-based healthcare systems, offering robust authentication and protection against known attacks.

Wireless Sensor Networks anonymous user authentication challenge addresses by

(Gope & Hwang, 2016). proposes a comprehensive authentication protocol for WSNs, ensuring security levels are user anonymity, forward/backward secrecy, and untraceability. The proposed scheme has 4 parts: 1) registration, 2) authentication, 3) key exchange, and 4) add of nodes dynamically. Each set outlines such as user registration, the session key establishment, password renewal, and the addition of new sensor nodes. The design objectives include achieving mutual authentication while preserving user anonymity and untraceability and providing practical solutions with reduced computation and communication costs. It also covers potential risks such as impersonation attacks via key compromise.

2.4 Conclusion

In summary, this literature review chapter has comprehensively analyzed the existing research on mobile digital identity. Digital identity in mobile has become more important due to increased connectivity today. As smartphones and other mobile devices multiply, there arises a need for secure and efficient ways of identifying and confirming users in different applications and services. In the literature review will discuss the main ideas, difficulties, authentication, authorization, privacy, security,interoperability, and progress made recently in the area of mobile identity. A systematic literature review has shed light on the landscape of mobile IDs for e-government and other applications. We have identified shortcomings in traditional mobile eID solutions within the e-government context and in the technologies and methods employed for lightweight key generation in mobile eID solutions. The research focuses to address the gaps in existing mobile identity solutions (such as mobile-based identification systems, outside key generation, Lightweight algorithms, signing, vendor lock-in, cost, and digital ID frameworks) and enhanced security and user-controlled authentication. Furthermore, our study has expanded its focus to scrutinize the storage and retrieval of critical attributes. The subsequent sections enumerate relevant findings and elucidate the distinctions between our work and previous research efforts. By addressing these gaps and offering insights, our study contributes to advancing mobile ID solutions to more secure and efficient e-government services and beyond.

Chapter 3

Designing a Framework to Provide Efficient and Secure Mobile User Identity and Signing Solutions Through Handheld Devices.

3.1 Introduction

The main objective of this chapter is to provide an analysis of the research findings and suggest a strategy for uniting government agencies. So, they can work together to administer and share citizen identification information in their identity management system based on mobile devices. The system will enable dispersed autonomous e-government systems to connect to a single external interface via the e-government directorate. We created a mobile ID architecture using public-key infrastructure (PKI), Enhanced elliptic curve cryptography (EECC), and token-based authentication. This approach is independent of SIM/card or hardware components.

We also look over the proposed methods for mobile identity authentication.

3.1.1 Mobile as a User ID Solution

The proposed solution offers a secure authentication for performing online transactions via mobile phones, including banking, payments, and services, at any time and location. This innovative approach, based on mobile PKI technology and indepen-

dent of external hardware or SIM cards, ensures the user's personal data, including demographic information and mobile details, is saved securely and confidentially in a dedicated location within the device. The suggested system, with its novel architecture, promises to be cost-effective, secure, user-friendly, and interoperable, providing a positive outlook for the future of mobile identity and signing solutions.

In the earlier section, we illustrated the non-SIM and non-HSM options. However, the proposed framework could aid in the spread of the Mobile ID solution for citizens. Service providers and government agencies can deliver online services via mobile handsets using a non-SIM, non-HSM-based framework scheme. With this approach, the user may easily share their digital identity, which is device-specific and connected to their demographics, to use online services. Other biometrics, such as a sensitive authentication mechanism, may be employed to verify the user. These days, mobile devices also offer biometric authentications like fingerprint and facial recognition. The entire process is shown in Figure 3.1. Here, in the diagram, we show the functioning of the proposed architecture.

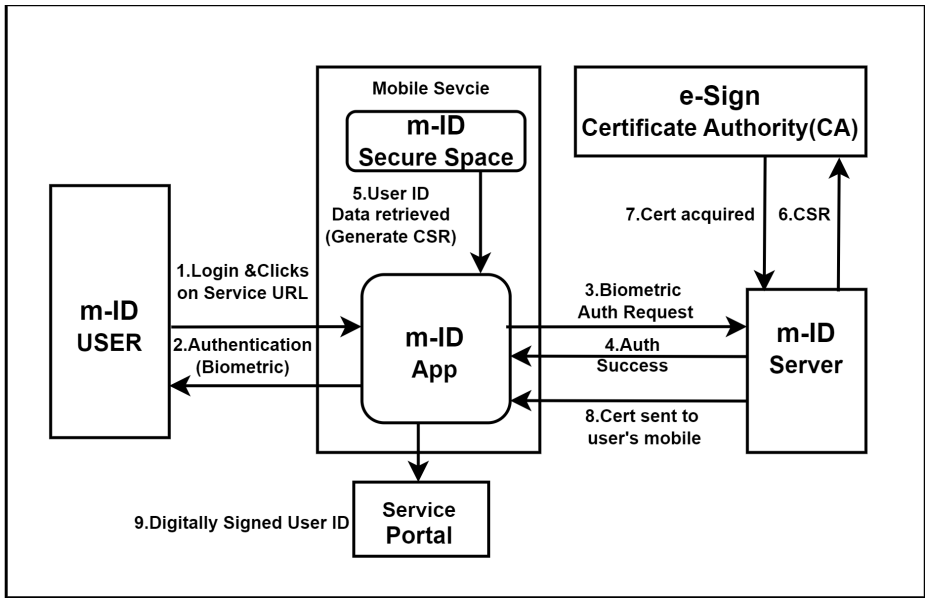


Figure 3.1: Mobile as User ID Solution Prototype

3.1.2 Business Modeling

The research findings, figure 3.2 illustrate a business model domain encompassing multiple parts focused on the Secure and Efficient Mobile as a Personal Identity uti-

lizing PKI framework. The framework consists of various interconnected components. The description of each component is given below:

- User layer: Mobile app that is used to generate key pairs (PK, SK) based on user data and facilitates the provision of user confidentiality information during an online transaction. Cryptography keys are also kept on devices like Secure Element. Keys should not be exported and only used for cryptography after being stored in the Keystore. It also offers options for restricting when and how keys can be used, subjecting key usage to user authorization or capping key usage to particular cryptography modes. Users have full control over revoked/destroyed keys at any time.

- Communication Layer: Layer is used for communication between the user, server, and facility layers.

- Mobile id (m-ID) Server layer: The user device is connected to the communication to the m-ID Server on a Secure connection. The user details and encrypted value of the device IMEI Number, ID, associated installation ID and return token are likewise stored on the m-ID Server.

  - The token becomes invalid after a predetermined duration from the day it was created.
  - Server generated token will be delivered to the device for the subsequent action to be performed.
  - After use, a token is immediately invalidated.

- Facility layer: This layer contains governmental organizations, hospitals, healthcare centers, banks, transport departments, etc.

  All the layers communicate with each other on a secure connection. The proposed user authentication, encryption, and signing scheme for mobile device data comprises four stages: (a) User Authentication, (b) Encryption process, (c) Decryption process, and d) Signing.

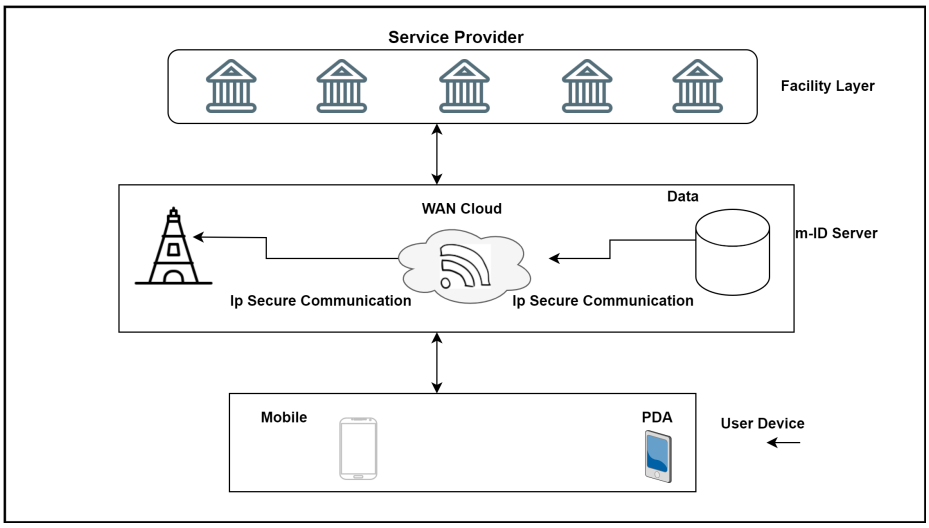


Figure 3.2: Business Components of the proposed solution

The proposed scheme has three steps: 1) user registration, 2) login, and 3) verification. First, the user registers his or her details through a mobile application. After registration, the user logs into the app with a username and password. When each user is registered on the servers, the user demographic information will be verified through the Identity Issuing Authority (IIA), and user details will be stored in the m-ID server and its database.

The server verifies the user using the hash function for authentication and verification. When a user is authenticated, a token is generated and linked to the user's device and the m-ID server for communication. To protect against attacks, the device information is encrypted and transferred to the m-ID server. Initially, user data from the mobile device is encrypted using the substitution cipher.

In cryptography, the substitution cipher is a mechanism for encrypting plain text by replacing it with cipher text. Following that, the ciphered data is encrypted using the EECC technique. The server collects encrypted user data and decrypts it before sending it to the server. subsequently, the server transfers decrypted data to the SP. Figure 3.3, Demonstrates the sequence diagram of the proposed authentication mechanism.

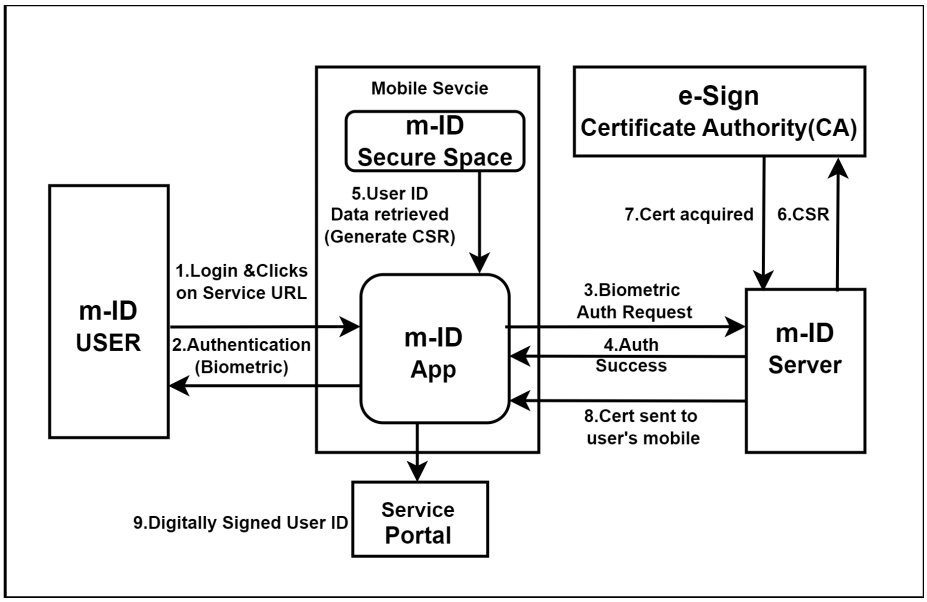


Figure 3.3: Proposed Workflow for m-ID

The proposed solution has different components that are elaborated further below.

- Registration: User demographic data is gathered and verified through the IIA (Identity Issuing Authority). Biometric details and demographic information will be used for user authentication on mobile devices. A successful registration and verification application will generate a key pair with device information and be stored in MSS (Mobile Secure Space). Figure 3.4 shows the registration process flow.

- Identity Issuing Authority (IIA): IIA is a governmental body that provides a physical identity (PAN, election ID, driving license, health ID, etc) unique to every citizen. users' submitted details will verified by IIA; then the registration process is completed.

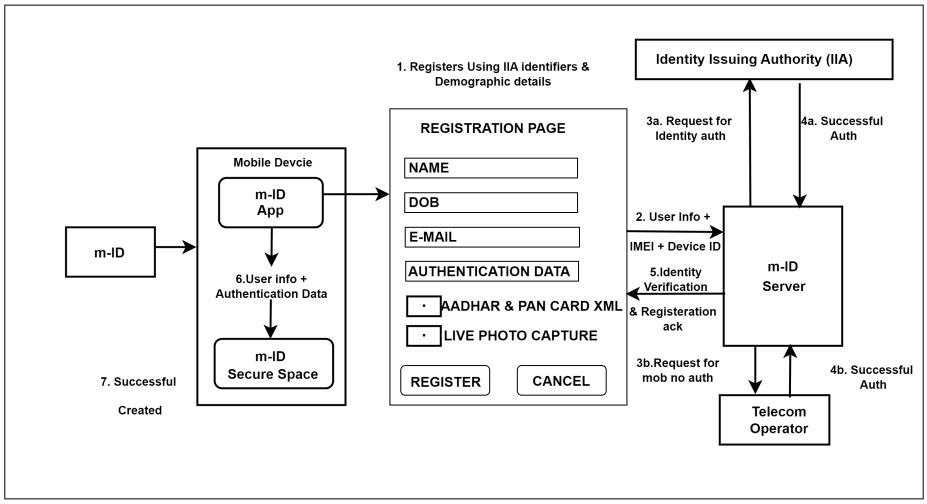


Figure 3.4: Registration Process

- Generation of Key Pair: The size, speed, and efficiency of key generation, a crucial component of the entire system, depends on the algorithm to generate the keys on mobile devices. The private key in the device is under user control and retains confidentiality. The proposed solution suggested that the asymmetric cryptography technique, ECC, is lightweight in computation. Figure 3.5. Shows the key generation process.

- Certifying Authority (CA) - After the generation of the key pair on a handheld device, a (CSR) certificate signing request is sent to CA to obtain certificate X.509 to perform transactions or signing on the device. CA is a trustworthy agency that provides digital certificates. A CSR generates x.509 certificates by combining a public key, device, and user information. CA included the Mobile ID, Unique Identifier, and Device ID parameters to the proposed solution's x.509 certificate.

- SSS (Storage Secure Space): Create a secure space within mobile memory that can be accessed by the m-ID app only after receiving the user's authorization and authentication request.

- Key Revocation: The user has the ability to erase the private key from a mobile device. If a mobile device is lost or stolen and after completing the transaction, the user can commence the process of revoking the key from the device by using the m-ID app.

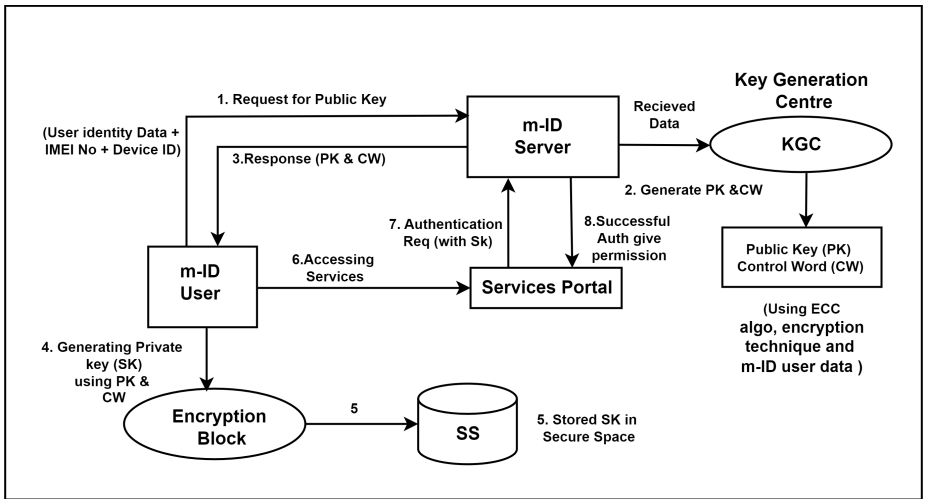


Figure 3.5: Key Generation

3.2 Contributions

To address the online identity authentication problem, we first analyze the design requirements and user perspective. Based on this study, we propose an architecture for Mobile ID based on PKI with EECC for the generation and maintenance of key pairs onboard a user's mobile device

- We propose an EECC method to create the user's key pair that binds with the device to meet the user's security criteria to assure the key security.
- This scheme overcomes the problem of generating the key on SIM or external hardware. Further, the generated key is securely stored inside the device in secure storage.
- We perform informal analysis to demonstrate that our framework can prevent various security attacks.
- We demonstrated that our framework can ensure user authentication and key security using the BAN logic and Scyther verification tool.
- We demonstrated that our procedure performs significantly better than other relevant methods through theoretical and experimental examination

3.3 Proposed Approach

This section provides detailed explanations about the proposed approach.

3.3.1 Proposed Algorithm for key Generation

Step 1) Begin.
Step 2) Starts encryption
  2.1) Generation of the hash value r_p Hash Function Hash (m) on Sever (where m is demographic details of user and hash function like SHA2)
  2.2) Generation of Encryption constant (e_{is}) using mID user device details like IMEI_NO and SIM_ID
  2.3) Stop encryption
Step 3) Evaluation of Private Key (SK) using ECC.
  3.1) Compute private key equation using t-Degree Polynomial of ECC
  3.2) Calculate $SK = \sum_{i=1}^{t} a_i (r_p)^i$ for each value of i, ai is selected randomly from the group G (G is an algebraic group of no's on the polar graph of ECC).
Step 4) Evaluation of Public Key (PK) using Private Key.
  4.1) Calculate $PK = SK \times e_{is}$ where SK is a private key and eis is encryption const.
Step 5) Store Private Key inside the mobile memory secure area.
Step 6) Repeat step 2 to 4 for each new user of mID app.
Step 7) End

Figure 3.6: Generation of Public Key

3.3.2 Proposed Verification of Key Pair

```
Step 1) Begin.
Step 2) Generation of X.509 certificate
     2.1) Submit key and CSR to authorized CA
     2.2) Collect cert from CA and install on user mobile device
Step 3) Authentication request
     3.1) User will send the request to server to get permission
          to access our services
     3.2) Request contains X.509 certificate with another user details
     3.3) Request done
Step 4) Verification of user
     4.1) mID server will verify user X.509 certificate with Public Key
     4.2) Server will try to match its Public Key component with
          data inside that certificate
     4.3) If it gets matched it will be successful authenticated
          and verified
     4.4) Verification done
Step 5) If X.509 certificate is not getting verified with the
        public key user will be rejected and blocked.
Step 6) End
Step 7) End
```

Figure 3.7: Verification by Public Key

3.3.3 Detail Explanation about ECC

Public keys based on ECC encryption techniques are built faster, smaller, and more efficiently. These keys perform security functions that include encryption, authentication, and digital signatures. They allow the verifier to verify the signer's authenticity without knowing their secret key. Common public key cryptosystems include ECC, RSA, and ElGamal. ECC is efficient and ideal for resource-constrained devices like IOT, smart cards, and PDAs. ECC is based on elliptic curve mathematics. ECC's security comes from the difficulties of solving the elliptic curve discrete logarithm issue.

ECC offers comparable security with smaller key sizes. Furthermore, it enables a wide range of solutions adapted to specific needs. This feature ECC runs on mobile hand-held devices. An ECC equation is shown below:

$$y^2 = x^3 + ax + b \tag{3.1}$$

$$4a^3 + 27b^2 \neq 0 \tag{3.2}$$

The strengths of Elliptic Curve Cryptography are as follows:

- Elliptic curve discrete logarithm: Its security comes from its one-way multiplication, which provides a powerful trapdoor function. A trapdoor function is simple to calculate in one way but difficult to calculate in the opposite direction (i.e., finding its inverse) without enough knowledge, also known as the "trapdoor." Trapdoor functions are frequently used in cryptography.
- Similar strength: It provides the same strength as RSA using a smaller key size.
- Enhanced security: It is often regarded as the most secure cryptosystem.
- Increased efficiency: It is more effective than the initial generation cryptography systems.
- Less CPU resources: It uses less memory and shorter encryption keys than other algorithms like Diffie-Hellman, resulting in faster than RSA.

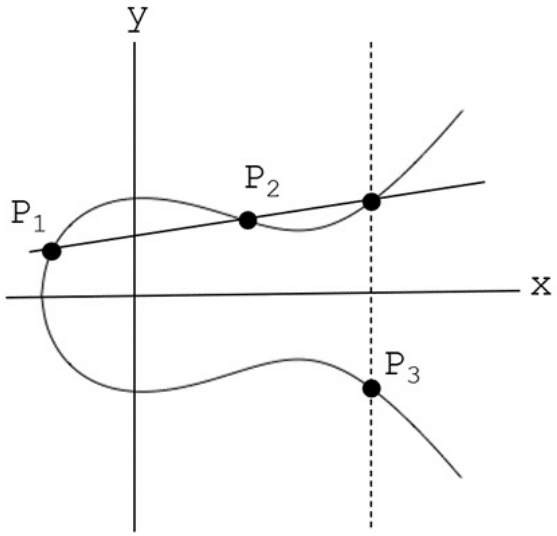


Figure 3.8: Elliptic curve

Comparison between ECC and RSA

ECC is a cryptography scheme that offers higher safety per bit compared to other widely used encryption methods, such as RSA. ECC-160 offers enhanced security compared to RSA-1024, while ECC-224 delivers more security than RSA-2048. The ECC algorithm is a robust cryptosystem that ensures the same level of security as RSA while using 1/6 of the key size.

Table 3.1: Comparison of ECC and RSA

Bits	RSA (Key Size)	ECC (Key Size)	Size Ratio
80	1024	160	1:7
112	2048	224	1:10
128	3072	256	1:12
192	7680	384	1:20
256	15360	521	1:30

3.4 Security and Performance Analysis

In this section, we present the experimental process and results.

3.4.1 Informal Security Analysis

The informal security analysis for the known threats is as follows:

- Stolen Device: In this attack, Adversary A_d obtains or steals mID. Then, A_d can retrieve the saved parameters r_p in mID. However, A_d cannot gather any information on Ui because all information stored in mID is masked (IDi, PWi) and Bi such as $A = (RNuh(IDi||PWi||hb(Bi)))$. As a result, our scheme can withstand attacks using stolen mobile phones.
- Message Replay attack: We assume that A_d communicates with U with mID . Messages are used for operations like spoofing and impersonation. We use three variables that protect from replay attacks. The first variable is $Token(T)$, the second is a (SK), and the last is device details m. The receiver can easily validate the freshness of received messages with the help of these variables. Consequently, our suggested approach is resistant to message replay attacks.
- Man in the Middle (MITM) attack: We assume that an $A_(d)$receives messages from U and mID for communication. In the proposed scheme, if mID receives UID, r_p, T, G, he/she will not get or read inside data due to hashing and encryption through ECC. Using PK, e_is, A_d due to insufficient information, the SK could not be calculated for $T2, UID, r_p$. As a result, our scheme is protected from MITM attacks.
- User impersonation attack: Impersonate as an authorized user, an A_d captures mS variables are $r_p, IMEI, sID, G$. Using these variables, an A_d cannot generate valid

requests PK, SK, mID, G . An A_d attempts at imitation fail as a result. The suggested scheme is hence protected from user impersonation attacks.

- Server impersonation attack: Impersonate as an authorized Server, an A_d captures the public message and public information, such as the Mobile phone. Using this variable, an A_d cannot generate a valid reply $mIDnew = m, PK, T1$ for the user due to the inadequacy of variable $sk - S$. As a result, an adversary's user impersonation attempts fail. Similarly, an adversary A_d cannot decrypt the message U and mID due to the non-availability of the Server. As a result, a server impersonation attack is unlikely to harm the proposed scheme.
- Mutual authentication: The suggested method accomplishes mutual authentication. The server mS calculates PK and validates by SK. mS also validates the U and m of the r_p . The Ui authenticates mS by verifying $r_p, IMEI, sID$ The received key is used to calculate the PK. We, therefore, claim that the proposed scheme satisfies device and server mutual authentication.

3.4.2 Formal Verification of m-ID framework Using Scyther Tool

Scyther is a security protocol tool that operates on cryptographic assumptions; it evaluates all security protocols based on the assumption that no adversary can read encrypted communication unless the decryption key is valid. Scyther is useful in finding flaws with the protocol's construction, which are generally intractable. However, numerous methods have been proven to be correct, and future attacks have been identified. Security attributes or security claims for the security protocol are specified as specifications in "Security Protocol Description Language" and saved in the ".spdl." file.

Security Claim Events: We must evaluate all the security protocols that must be satisfied to build a secure scheme. In the Scyther verification tool, these security properties are wrapped as claim events and made part of the protocol definition.

- Secret: This concept, often referred to as "secrecy," asserts that when a message is transmitted through an untrusted channel, it remains concealed from potential attackers.

- Authentication: Authentication can manifest in various forms in literature, but in essence, it serves as a confirmation that an entity engaged in communication truly exists. According to protocol definitions, the presence of at least two communicating entities is essential.
- Synchronization: This is a requirement within a security protocol that ensures that two interconnected entities have successfully exchanged the necessary messages. Moreover, it verifies that the received data is from the correct sender and that all messages arrived at the recipient in the correct sequence.
- Non-injective Synchronization: This specification emphasizes that the trace should encompass all events stipulated in the protocol definition. Since coordination primarily deals with the content and order of content, it is susceptible to replay attacks. An attacker may intercept a message and later resend it to the recipient; typically, this is not considered a form of coordination attack.
- Injective agreement: The authors suggest that to achieve agreement between two communicating entities, namely, the initiator (I) and responder (R), concerning a dataset (ds), both I and R must have previously completed a protocol run in their respective roles (i.e., R serving as the responder and consenting to the dataset ds, with each initiation run having a unique corresponding run involving R).

  The security analyses of the proposed solution are performed using the Scyther tool. The proposed solution can avoid a variety of security attacks and the results are presented 3.9 and 3.10. Whereas the verification results are shown in figure 3.11.

File Verify Help

Protocol description Settings

```
1 const pk: Function;
2 secret sk: Function;
3 inversekeys (pk,sk);
4 hashfunction H;
5
6 protocol m-ID-Scheme(A,S)
7 {
8 role A
9 {
10 const imei: Nonce;
11 const id: Nonce;
12 const I: SessionKey;   // Assumed I is the installation ID of mobile client is shared between A & S over https and it is confidential
13 var T1: Nonce;
14 var Hm: Nonce;
15 const T2: Nonce;
16 const DCSR: Nonce;
17 var CERT: Nonce;
18
19 send_1(A, S, {A,S, H(imei, id, I), I}pk(S));
20 recv_2(S, A, {S, A, {H(T1)}sk(S), T1, Hm }I);
21 send_3(A, S, {A, S, DCSR, pk(A), H(T1), {H(T2)}sk(A), T2 }pk(S));
22 recv_4(S, A, CERT, {T2}sk(S));
23 //recv_4(S, A, {CERT, {T2}sk(S)}pk(A));
24
```

Figure 3.9: Protocol Verification-I

```
25 claim_A1(A,Secret,imei);
26 claim_A2(A,Secret,id);
27 claim_A3(A,Secret,I);
28 claim_A4(A,Secret,Hm);
29 claim_A5(A,Secret,T1);
30 claim_A6(A,Secret,T2);
31 claim_A7(A,Secret,DCSR);
32 claim_A8(A,Secret,pk(A));
33 claim_A9(A,Niagree);
34 claim_A10(A,Nisynch);
35 }
36
37 role S
38 {
39 var imei: Nonce;
40 var id: Nonce;
41 const I: SessionKey;   // Assumed I is the installation ID of mobile client is shared between A & S over https and it is confidential
42 const T1: Nonce;
43 const Hm: Nonce;
44 var T2: Nonce;
45 var DCSR: Nonce;
46 const CERT: Nonce;
47
48 recv_1(A, S, {A,S, H(imei, id, I), I}pk(S));
49 send_2(S, A, {S, A, {H(T1)}sk(S), T1, Hm}I);
50 recv_3(A, S, {A, S, DCSR, pk(A), H(T1), {H(T2)}sk(A), T2 }pk(S));
51 send_4(S, A, CERT, {T2}sk(S));
52 //send_4(S, A, {CERT, {T2}sk(S)}pk(A));
53
54 claim_S1(S,Secret,imei);
55 claim_S2(S,Secret,id);
56 claim_S3(S,Secret,I);
57 claim_S4(S,Secret,Hm);
58 claim_S5(S,Secret,T1);
59 claim_S6(S,Secret,T2);
60 claim_S7(S,Secret,DCSR);
61 claim_S8(S,Secret,pk(A));
62 claim_S9(S,Niagree);
63 claim_S10(S,Nisynch);
64 }
65 }
66
```

Figure 3.10: Protocol Verification-II

Scyther results : verify

Claim				Status		Comments
m_ID_Scheme	A	m_ID_Scheme,A1	Secret imei	Ok	Verified	No attacks.
		m_ID_Scheme,A2	Secret id	Ok	Verified	No attacks.
		m_ID_Scheme,A3	Secret I	Ok	Verified	No attacks.
		m_ID_Scheme,A4	Secret Hm	Ok	Verified	No attacks.
		m_ID_Scheme,A5	Secret T1	Ok	Verified	No attacks.
		m_ID_Scheme,A6	Secret T2	Ok	Verified	No attacks.
		m_ID_Scheme,A7	Secret DCSR	Ok	Verified	No attacks.
		m_ID_Scheme,A8	Secret pk(A)	Ok		No attacks within bounds.
		m_ID_Scheme,A9	Niagree	Ok	Verified	No attacks.
		m_ID_Scheme,A10	Nisynch	Ok	Verified	No attacks.
	S	m_ID_Scheme,S1	Secret imei	Ok	Verified	No attacks.
		m_ID_Scheme,S2	Secret id	Ok	Verified	No attacks.
		m_ID_Scheme,S3	Secret I	Ok	Verified	No attacks.
		m_ID_Scheme,S4	Secret Hm	Ok	Verified	No attacks.
		m_ID_Scheme,S5	Secret T1	Ok	Verified	No attacks.
		m_ID_Scheme,S6	Secret T2	Ok	Verified	No attacks.
		m_ID_Scheme,S7	Secret DCSR	Ok	Verified	No attacks.
		m_ID_Scheme,S8	Secret pk(A)	Ok	Verified	No attacks.
		m_ID_Scheme,S9	Niagree	Ok	Verified	No attacks.
Done.		m_ID_Scheme,S10	Nisynch	Ok	Verified	No attacks.

Figure 3.11: Results

3.4.3 Formal Verification of m-ID framework Using BAN-Logic

This subsection demonstrates that our system accomplishes the property of mutual authentication by utilizing BAN logic. The BAN logic generates trust between the parties involved in the communication between the app and server. The system operates based on accurate inference rules and assumptions to achieve objectives.

Step 1: Begin
This is the starting point of the algorithm.
Step 2: Start Encryption
This step initiates the encryption process.
Step 2.1: Generation of Hash r_p
Calculate a hash value r_p using a hash function (e.g., SHA2) on demographic.
The server stores information about the user ('m'). This hash value is likely used for security purposes.
Step 2.2: Encryption process
Compute an encryption constant using user mobile device details like IMEI NO and Device ID. This constant might be used in the encryption process.
Step 2.3: Stop Encryption
This step appears to indicate the end of the encryption setup.
Step 3: Evaluation of SK (Private Key) using ECC.
This step focuses on generating an SK using ECC.
Step 3.1: Compute SK equation using ECC t-Degree Polynomial
computes the private key equation using a polynomial of degree 't' related to Elliptic curve cryptography. The specific details of this equation are not provided.
Step 3.2: Computes SK = $\sum_{i=1}: a_i\ (r_p)$
Compute the SK by summing up the product of random values a_i selected from an algebraic group 'G' with the corresponding hash value r_p This step is using the private key equation calculated in the previous sub-step.
Step 4: Generate a Public Key (PK) using a SK
Generate a PK using the SK and the encryption constant e_{is} calculated in step 2.2.
Step 4.1: Calculate PK = SK × e_{is}
The is computed as the product of the SK and the e_{is}.
Step5: SK stored inside the mobile memory secure area to protect it from unauthorized access.
Step 6: Repeat steps 2 to 4 for every new user of mID app
This step suggests that the entire process from step 2 to step 4 should be repeated for every new user of the mID app, allowing each user to have unique private and public keys.
Step 7: End

Notation	Definition
r_p	*Encrypted user demographic information of m-ID users.*
mN	*Mobile number of users*
mS	*Mobile System*
IMEI, sID	*IMEI Number of used systems, SIM-ID*
m	IMEI ‖ sID ‖ mN}
G	*Group of elements derived from ECC polar plot*
t	*The upper limit in the summation equation of key.*
p	*Used to denote terms in private key equations.*
e_{is}	*Constant member of encryption*
Q_t	*t-Degree polynomial using the ECC algorithm.*
a	*Element in group G.*
pk-A	*Public Key of A*
T1, T2	*Verification Token*
sk-A	*Private Key of A*
pk-S	*Public Key of Server*
sk-S	*Private Key of Server*
PK	*User Public Key*
SK	*User private key*
tO	*Telecom operator*
U	*m-ID android App user*
mID	*Mobile Identity no for each user*
$\oplus$	*Exclusive or operation*
||	*Concatenation operation*
A_d	*Adversary*

Figure 3.12: Notations of BAN logic

1. Begin
2. Verification with telecom operator
   (a) Generation of the verification request to $tO^{r_p(IMEI,sID,}\ mN)$

$$r_p(IMEI, sID, mN).$$

   (b) $AKtO(IMEI||sI||mN\ )$ generated for authentication signal 1 or 0 (denotes YES or NO).

(c) $\oplus^{AKtO}(IMEI, sID, mN)$ AKtO $(IMEI, sID, mN)$ in m-ID server (Secure Storage)

3. Generation of Hash and request of CSR

(a) H(m) where m = IMEI‖sID‖mN, T1 = Verification Token

(b) mS receive signal (sig1). The annotation rules yield that
$\text{sig}\,1[\text{H(m)}, \{\text{H(T1)}\}sk - S, T1]$ send to mS (Mobile System).
$\text{mS} < \text{sig}\,1[\text{H(m)}, \{\text{H(T1)}\}sk - S, T1$

(c) Since we have the hypothesis:
mS **believes** $\{\text{H(T1)}\}sk - S$ **where** mS[sk-S] S

$$\frac{\text{mS}| \equiv \text{S}| \equiv (\text{sig}\,1[\text{H(m)}, \{\text{H(T1)}\}sk - S, T1])}{\text{mS[sk-S]} \quad \text{S}}$$

- The message rule for shared keys is applied, yielding the results shown below:

  mS **believes** $\{\text{H(T1)}\}sk - S$ **where** mS[(pk-S,sk-S)] S.

$$\frac{\text{mS} \mid\equiv \{\text{H(T1)}\}sk - S}{\text{S[(pk-S,sk-S)]} \quad \text{S}}$$

- Hypothesis:
  mS **believes** fresh (T2)

- The nonce-verification rule applies and yields.
  mS **believes** S **believes** [{H(T1)} sk-S, T1]

$$\text{mS}| \equiv \text{S}| \equiv [\{\text{H(T1)}\}\ \text{sk-S},\ \text{T1}]$$

- We break a conjunction once more to get the following:
  mS **believes** $\{\text{H(T1)}\}sk - S$ **where** mS[(pk-S, sk-S)mS] S.

$$\frac{\text{mS}| \equiv \text{S}| \equiv [\{\text{H(T1)}\}\ \text{sk-S, T1}]}{\text{mS}| \equiv \text{mS}[(\text{pk-S,sk-S})\text{mS}] \quad \text{S}}$$

- Then, we instantiate $(pk, sk)S$ to $(pk - S, sk - S)mS$ in the hypothesis mS **believes** S **controls** [{H(T)} sk, T]

- Deriving the more concrete
  mS **believes** S **controls** [{H(T)} sk, T]

- Finally, the jurisdiction rule is applied, which results in the following:
  mS **believes** S **controls** [{H(T)} sk, T]

$$\text{mS} \Rightarrow \text{S}$$

$$\frac{\text{mS}| \equiv \text{S}| \equiv [\{\text{H(T)}\}\ \text{sk, T}]}{\text{S} \equiv \text{mS}[(\text{pk,sk})\text{S}] \quad \text{S}}$$

This concludes the analysis of signal(sig1). mS Send the token to S, together with sending another message Hash(m). Initially, mS can generate a Key with token T2:
S **believes** mS **believes** [{H(T2)} sk-S, T2]

$$\text{mS} \equiv \text{S} \mid\equiv [\{\text{H(T2)}\}\ \text{sk, T2}]$$

(d) Generated keys sk-A and pk-A stores into m-ID secure space.

4. Generation of acknowledge signal sig2

   (a) Encrypt [CSR ‖ pk-A ‖H(T1)‖{H(T2)} sk-A ‖m] where " m = demographic info = $m = IMEI\|sID\|mN''$.

   S ◁ Encrypt [CSR || pk-A | H(T1) —— {H(T2)} sk-A | m]

   (b) mS **believes** S **believes** Encrypt [CSR $\|pk - A\|H(T1)||\{\text{H(T 2)}\}sk - A\|$m]

   mS| ≡ S| ≡ Encrypt [CSR || pk-A ‖H(T1)‖{H(T2)} sk-A | m]

5. Validation of CSR from eSign certifying authority CA

(a) CA **believes** S **believes** sig 3{CSR, m} and validate.

CA| ≡ S| ≡ sig 3{CSR‖m}

(b) Generates demographic verification response 1 or 0, CERT.

Generates X509 digital signing certificate.

(c) ⊕ **Storage (CERT, 1 or 0, pk-A)** in m-ID Server Secure Storage.

⊕ Storage (CERT, 1 or 0 , pk-A)

6. Generation of CERT signing verification at mS signal sig3

(a) mS receive signal (sig4). The annotation rules yield that

sig4[CERT —— {H(T2)}$sk - S$] send to mS (Mobile System).

mS◁ sig4[CERT ‖{H(T2)}$sk - S$]

(b) Since we have the hypothesis

mS **believes** {H(T2)}$sk - S$ **where** mS[sk-S] S.

$$\frac{\text{mS} \equiv \{\text{H(T2)}\}sk - S}{\text{mS[sk-S]} \quad \text{S}}$$

- Message rule for shared keys: applies and yields the following:
  mS **believes** {H(T2)} sk- S **where** mS[(pk-S,sk-S)mS] S.

- Moreover, we have the following hypothesis:
  mS **believes** decrypt(T2)

- The nonce-verification rule used and yields
  mS **believes** S **believes** [{H(T2)} sk-S, T2]

- Again, we break a conjunction, to obtain the following:
  mS **believes** {H(T2)}$sk - S$ **where** mS[(pk-S,sk-S)mS] S.

- Then, we instantiate (pk, sk) S to (pk-S, sk-S) mS in the hypothesis.
  mS **believes** S **controls**[{H(T)} sk, T]

- Deriving the more concrete
  mS **believes** S **controls** [{H(T)} sk, T]

- Finally, the jurisdiction rule is applied, and yields the following:
  mS **believes** S **controls** [{H(T)} sk, T]

7. End

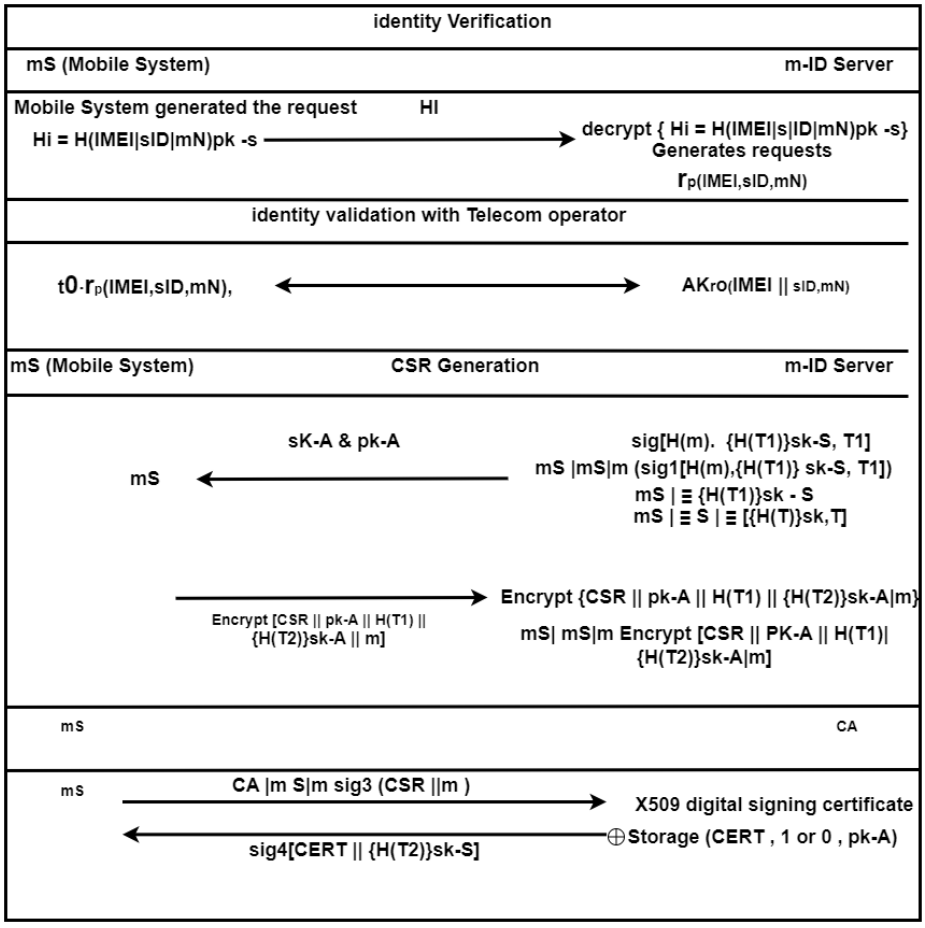


Figure 3.13: Proposed Scheme: Key Generations and Authentication Phases

.

3.4.4 Verifying Secure and Key Generation and Protection Framework

This section evaluates the security effectiveness of the proposed solution.

- user $M = \{M_1, M_2, M_3, ..., M_u\}$ be a set user of m-ID app. Any user of 'M' can register on the m-ID server anytime through app.
- The authority, called the m-ID server of users, is denoted by S_{mID}. The protocol executions for key agreement, users joining, and users leaving are regarded as different instances.
- M enters and computes private and public keys with correctly formatted messages.
- M denotes the set of users,M_n wants to establish a connection with the m-ID server with a private key n^{th} data stored on the m-ID server, which is a public key.
- M_n interacts with S_{mID} by adapting queries to the m-ID server machine and acquiring the needed information.

Symbol	Meaning
M	Set of mobile ID android users
mID	Mobile Identity
S_{mID}	mID Server
M_n	n^{th} user in M set
AK	Attacker
hash(m)	Hash value component
SK	Private Key
Msg	Request with identity information

Figure 3.14: Notations in the Security Model Symbol Meaning

The steps/queries issued by Mn and attacker AK are listed as follows:

1. hash(m): During the registration process, M_n sends a message request to the m-ID server, providing their identity data. The m-ID server then verifies whether M_n has been previously registered. If the answer is affirmative, then return the same value. Alternatively, it generates a hash value, represented as a number, for the input message m and then sends it back to M_n.

2. Extract (ID): The attacker AK has the ability to modify an entity and obtain the hash value of the private key for the (m) component of the entity. When AK sends a request for information to the m-ID server using the identity ID, the m-ID server returns the private key, which is the hash value of (m) corresponding to the ID.
3. Send (SK, Msg): In this scenario, the attacker AK can initiate an active attack. When AK transmits a message request, Msg, along with SK, to the m-ID server, the server responds with the output that would be generated by the actual protocol. AK can begin the communication process by issuing a Send SK inquiry.
4. Reveal (SK): This approach elucidates the vulnerability of session key verification to attacks. Upon receiving this query from AK, the server will provide the verified status of the instance's session key if it has been successfully authenticated. Alternatively, it will generate an error.
5. Execute(M_n,m-ID): Passive eavesdropping on a public channel. When AK performs such a query, the m-ID server follows the processes to execute the protocol between instances M_n and the m-ID server. It returns all of the messages exchanged throughout the operation.
6. Test (AK, mID): The attacker, AK, can only submit this query once. The mID server selects a random bit $R_b \in 0, 1. If$ R_b =1, the server returns a session key for verification. Otherwise, it returns a random value of equal size.

Note that we have replaced the compromised server with Extract (ID) and Send P_k, Msg. Long-term security in PKI-based identity depends on the private key generated during the Extract (ID) procedure.

To establish a meaningful notion of security, we must first define freshness.

- Freshness: An instance M_n is considered fresh if the session key has been confirmed and neither the Reveal nor Extract operations have been requested for M_n and the m-ID server.
- Definition of Security: The security of a protocol is determined by the following factors.
  - Upon conducting the Extract query, the assailant will solely acquire the Hash component of the private key. A private key, which includes the Hash component, has been created on the mobile device. Furthermore, we have incorporated

a constant term for secret encryption to prevent possible attackers from accessing it.

– Subsequently, the attacker will attempt to Reveal and Execute on the server; nevertheless, this endeavor is not authenticated due to the incorrect Private Key.

– At some point, the attackers send a Test query to a newly established server. After the Test query, the adversary AK produces a binary guess bit $R_b^{'0,1}$

3.5 Conclusion

In conclusion, this chapter has articulated a comprehensive strategy to foster collaboration among government agencies to seamlessly administer and share citizen identification information through mobile devices. The proposed framework, which falls under the e-government directorate, creatively integrates various e-government systems. Leveraging a mobile ID architecture on PKI, enhanced elliptic curve cryptography (EECC), and token-based authentication, this strategy ensures robust security and interoperability on online transactions while remaining independent of specific hardware components, such as SIM cards. The existing solutions are based on SIM cards provided by telecom operators and keys stored in external hardware like SIM without the user having control over key generation and revocation. Through the evaluation of our proposed methods for mobile identity authentication, we have demonstrated their efficacy and potential to enhance the efficiency and reliability of government services in the digital era. By advocating for this unified approach, we anticipate significant advancements in the realm of e-government, fostering greater connectivity and accessibility for citizens while maintaining the integrity and security of their personal information. The framework leverages comprehensive and detailed user information by integrating biometric data, key pair generation, and signing. The results prove the proposed method's superior performance in securing mobile identity verification.

Chapter 4

Analyzing and Enhancing Key Pair Generation that Binds Handheld Devices

4.1 Introduction

Enhancing ECC is suggested in our effort. The EECC is built on a curve with a certain base point that was created by combining prime number functions and solving specific equations. Traditional ECC generates two keys (public and private), whereas EECC generates an additional key, a 'unique key', used for cipher-based text encryption. This unique key is created by combining the final two digits of the IMEI number and the device ID. Furthermore, this is key to the Caesar cipher. Encryption has become more sophisticated in this approach, and decryption has become more difficult. This makes detecting the original data extremely difficult. As a result, the data's security is automatically improved. The ECC and operation of the Caesar cipher with the unique key are then described. itemize

4.2 Contributions

The proposed framework emphasizes using the enhanced ECC algorithm to generate and maintain key pairs within handheld devices. ECC is renowned for its robustness and efficiency, making it an ideal choice for securing sensitive cryptography operations

in mobile environments.

- We proposed two kinds of encryption to securely send user information substitution-ceaser cipher and enhanced ECC. An additional key (secret key) is generated in enhanced ECC to enhance the system's security.
- We proposed an additional key, a 'unique key,' based on the IMEI number and the device ID that binds user identity with the device.
- Unlike traditional approaches, the framework empowers users by directly enabling key generation and certificate issuance within the handheld device. This user-centric approach enhances security by minimizing reliance on external systems and granting users greater control over cryptography processes.
- We conducted a comparison of different algorithms with our algorithms. Following the results, our framework provides less time, lower communication costs, and better security.
- We conducted a comparison of the available solution with our framework with different parameters.
- The proposed framework represents a significant advancement in mobile user identity management and signing. By leveraging cutting-edge cryptography techniques, empowering users, and incorporating robust identifiers, it sets a new standard for security and efficiency in mobile-centric environments.

4.3 Proposed Methodology

This section (m-ID), which is a PKI-based solution for Mobile Device-based identity, has been detailed. This would give users a safe authentication method to conduct online activities like banking, payments, and other services from their mobile devices whenever and wherever they choose. The suggested solution will be based on Mobile PKI without any dependency on SIM or additional hardware. In this method, a secured element will be the user information, which includes demographics, SIM numbers, and IMEIs. The confidential data will be stored in a secure location on the device. This suggested solution is secure, low-cost, economical, user-friendly, interoperable, and secure. Figure 1 depicts the m-ID architecture. An authentication and

encryption mechanism is presented to ensure a safe connection between mobile phones and m-ID servers. Three steps comprise the proposed authentication and encryption strategy for mobile device data: authentication, encryption, and decryption.
ECC algorithm is built on a curve with an established base point generated using prime integer functions. The ECC algorithm is also more complex and challenging to construct, which expands on the ideas in 4.1. the likelihood of implementation mistakes, lowering the algorithm's security. Thus, Enhance ECC is suggested to increase security.

4.3.1 Enhance ECC (EECC) for Encryption

Enhancing ECC is what we suggest in our effort. EECC produced a specific base point curve by combining prime number functions and resolving particular equations. Traditional ECC generates two keys (public and private), whereas EECC generates an additional key, a 'unique key', that is used for cipher-based text encryption that binds with mobile devices. This unique key is created by combining the final two digits of the IMEI number and the device ID. Furthermore, this is the key to the Caesar cipher. Encryption becomes more sophisticated with this approach, and decryption becomes more difficult. This makes detecting the original data extremely difficult. As a result, the data's security is automatically improved. The ECC and operation of the Caesar cipher with the unique key are then described.

Elliptic Curve Cryptography:

Non-repeatable roots over the algebraic equations can be used to describe an elliptical curve (EC) E_{Curve} prime finite field $Field_n$.

A curve is expressed in the form $E(Field_n)$ and is defined over the finite binary field $Field_n$ of order n. Two curve points T (x, y) in equation 4.1

$$y^2 = (x^3 + ax + b) mod\ n \tag{4.1}$$

In addition, $a, b \in Field_n$ are two constants that fulfill the 4.2 requirement. The additive EC curve group G_n is described as

$G_n = (x, y) : x, y \in Field_n$ and $(x, y) \in E\} \cup \{O_i\}$
where O_i is a point at infinity $Field_n$ (Paar et al., 2010, Koblitz, 2000.)

$$4a^3 + 27b^2 (mod\ n) \neq 0 \tag{4.2}$$

The EC performs two operations: (1) point addition and (2) scalar point doubling. During the execution process, the double and add functions of the ECC are utilized to carry out scalar multiplication on the large random number and the points on the curve. Our method for creating public keys involves utilizing scalar point multiplication, defined as $n \times T = T + T... + T$ for n times.

EC point multiplication provides great anonymity, and when the adversary has $n \times T$ and T, the adversary cannot recover the value of n in polynomial time. Moreover, the difficulty of resolving the ECDLP is equivalent in security to RSA but requires a smaller key size.

4.3.2 Generation of Key Pair

During our study, we intend to present an improved public key implementation for secure communication. We use the ECC algorithm for the following reasons: - Compared to other public key methods, the key size required for equal security is less. (ii) ECC has low CPU and memory utilization.

The public key of ECC is made using equation 4.3.2, where k is the long input string of user credentials, and G is the generator point.

Private Key = Rn+ $I_{ID} \oplus d_{id}$(4.3)

And, the generation of the Public key is done using 4.4.

$$PublicKey = Pk * G \tag{4.4}$$

Enhancements

Literature is evident that ECC can be affected by MITM or replay attacks. Therefore, to overcome this problem, we provide an additional layer of security to the encrypted data in our proposal. Here, we make use of a ceaser cipher and unique key. The equation to make a unique key is as given in equation 4.5. In our proposal, plain text first goes through the cipher before encryption. and the reversal is done on the server side. Note that the parameters for generating the unique key are with the

server when a new user downloads the m-id application.

$$UniqueKey = (IMEIno) + DeviceID \tag{4.5}$$

Unique key converts the Plain text, let's say x, we execute a basic Caesar cipher. The equation 4.6 shows the encryption process.

$$Encryption(x) \equiv (x + UniqueKey)(\text{mod}26) \tag{4.6}$$

Equation 4.7 shows the decryption process after Encryption(x) is done.

$$Decryption(x) \equiv (x - UniqueKey)(\text{mod}26) \tag{4.7}$$

4.4 Time Complexity of Key Generation in ECC

In this section, we will delve into the time complexity analysis of key generation in ECC and explore optimization techniques to enhance computational efficiency.

4.4.1 Key Generation Procedure

1. **For Sender (A):**
   - Private Key SK_A $(SK_A < n)$
   - Calculate Public Key $PK_A = SK_A \cdot G$
2. **For Receiver (B):**
   - Select Private Key SK_B $(SK_B < n)$
   - Calculate Public Key $PK_B = SK_B \cdot G$
3. **Sender's Secret Key :**

   $k = SK_A \cdot PK_B$
4. **Receiver's Secret Key :**

   $k = SK_B \cdot PK_A$

4.4.2 Key Generation Code

```
def ECC_encryption_keys(publicKey):
ciphertextPrivateKey = secrets.randbelow(curve.field.n)
ciphertextPublicKey = ciphertextPrivateKey * curve.g
UniqueECCKey = publicKey * ciphertextPrivateKey
return (UniqueECCKey, ciphertextPublicKey)
```

4.4.3 Time Complexity Calculations

In the ECC key generation process, a PK (private key) and SK (public key) are derived, and shared secrets are computed. Below is a breakdown of the time complexity of each step:

1. **Private Key Generation:**

   Private key generation involves a random integer within a specified range. As the random function used for this purpose is independent of the value of n, the order of the elliptic curve group, it takes constant time $\mathcal{O}(1)$.

2. **Public Key Derivation:**

   Computing the SK from the PK involves scalar multiplication of a generator point (typically denoted as G) with the PK (d). This operation is typically performed through the Montgomery ladder and Double-and-add algorithms. These techniques significantly reduce the time complexity from quadratic to logarithmic, enhancing performance.

   (a) **Double-and-Add Algorithm:**

   Similar to the modular exponentiation method of square-and-multiply. It reduces the time complexity by adding a point at

a time along the elliptic curve to itself. to $\mathcal{O}(logn)$).

(b) **Montgomery Ladder:**

Computes point multiplication in a fixed number of operations, irrespective of the value of the multiplicand. It offers the same speed as the Double-and-Add approach but ensures constant time complexity and mitigates information leakage through branches or power consumption.

Algorithm 1 Montgomery Ladder Algorithm

```
S_0 ← 0
S_1 ← Q
for i from m downto 0 do
    if d_i = 0 then
        S_1 ← point_add(S_0, S_1)
        S_0 ← point_double(S_0)
    else
        S_0 ← point_add(S_0, S_1)
        S_1 ← point_double(S_1)
    end if
    // Invariant property to maintain correctness
    assert S_1 == point_add(S_0, Q)
end for
return S_0
```

3. **Shared Secret Computation:**

Deriving the shared secret between parties involves multiplying the receiver's 'SK' with the sender's 'PK'. Similar to 'SK' derivation, this operation also has a complexity of $\mathcal{O}(logn)$.

Total Time Complexity:

The total time complexity of key generation in ECC can be expressed as the sum of the time complexities of the individual steps:

$$Total\ Time\ Complexity = \mathcal{O}(1) + \mathcal{O}(logn) + \mathcal{O}(logn) = \mathcal{O}(logn)$$

The final time complexity of key generation in elliptic curve cryptography is logarithmic with efficient and scalable cryptography operations.

4.5 Time Complexity of Encryption in ECC

The encryption process in Elliptic Curve Cryptography (ECC) plays a critical role in securing communications by encoding messages and generating cipher texts. In this section, we will analyze each step of ECC time complexity, focusing on the individual operations involved and their computational efficiency.

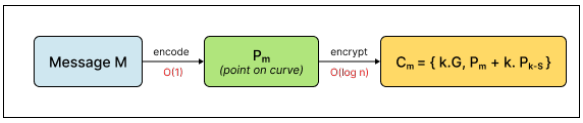


Figure 4.1: Encryption in ECC

ECC, the encryption process involves encoding a m message as a point P_m on the EC. Subsequently, this encoded message is encrypted by the receiver (public key) P_b, resulting in a ciphertext $C_m = (k \cdot G, P_m + k \cdot P_b)$, where G represents the generator point and k denotes the unique key with sender and receiver.

A breakdown of time complexity calculation for the steps involved is given below:

1. **Message Encoding as a Point (P_m):**
    - **Operation:** Encode the message m as a point on the EC.
    - **Time Complexity:** This operation is typically not computationally intensive and can be considered $\mathcal{O}(1)$ or very low in practice.
2. **Scalar Multiplication ($k \cdot G$):**
    - **Operation:** Multiply the generator point G by the shared unique key k.
    - **Time Complexity:** Scalar multiplication is a common operation in ECC, with a time complexity is $\mathcal{O}(\log n)$, where n is the base point.
3. **Point Addition ($P_m + k \cdot P_b$):**
    - **Operation:** Add the encoded message point P_m to the result of $k \cdot P_b$, where P_b is the recipient's public key.
    - **Time Complexity:** Time complexity in point addition in ECC is approximately $\mathcal{O}(\log n)$, where n is the base point.

The time complexity of the encryption step in ECC is dominated by scalar multiplication and point addition $\mathcal{O}(\log n)$, where n is the base point.

4.6 Time Complexity of Decryption in ECC

Decryption in ECC is a crucial process wherein the recipient retrieves the original message from the ciphertext encrypted by the sender. In this section, we will analyze ECC time complexity for the decryption step, focusing on the individual operations involved and their computational efficiency

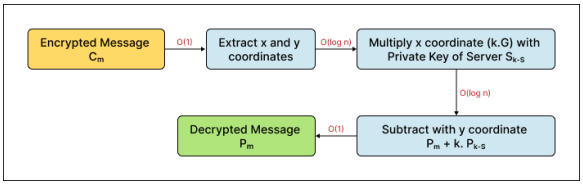


Figure 4.2: Decryption in ECC

In ECC, decryption involves receiving an encrypted message C_m consisting of two components: $k \cdot G$ and $P_m + k \cdot P_b$, where G represents the generator point, P_m is the encoded message point, P_b denotes the receiver (public key), k signifies the shared unique key, and n_b refers to the receiver (private key).

A breakdown of time complexity calculation for the steps involved is given below:

1. **Extracting y-coordinate of $P_m + k \cdot P_b$:**
    - **Operation:** Extract the y-coordinate of the second component $P_m + k \cdot P_b$.
    - **Time Complexity:** Extracting the y-coordinate of a point on an EC is typically considered a constant-time operation ($O(1)$).

2. **Multiplying x-coordinate with the private key n_b:**

   - **Operation:** Multiply the x-coordinate of the first component $k \cdot G$ by the recipient's private key n_b.
   - **Time Complexity:** Scalar multiplication is a common operation in ECC, with a time complexity of approximately $O(\log n)$, where n is the base point.

3. **Subtracting the result from the y-coordinate:**

   - **Operation:** Subtract the result of the multiplication from the y-coordinate obtained in the first step.
   - **Time Complexity:** Subtracting two numbers is a constant-time operation ($O(1)$).

Overall Time Complexity: The dominant factor in the time complexity of the decryption step in ECC is the scalar multiplication ($k \cdot G$) step, which has a time complexity of $O(\log n)$, where n is the base point. The remaining processes in decryption are constant-time and have no major impact on overall time complexity.

4.6.1 Detailed Work-flow

A PKI-based solution for mobile device-based identity has been described in this subsection (m-ID). This would give customers a secure authentication method to use their mobile devices whenever and wherever they wanted to do online banking, payments, and other services. The suggested fix will be based on Mobile PKI and won't require a SIM card or additional hardware. The user information, which includes demographics, SIM numbers, and IMEIs, will be safeguarded in

this manner. The confidential data will be kept in a secure area on the device itself. The suggested authentication and encryption approach for mobile device data consists of three steps: authentication, encryption, and decryption.

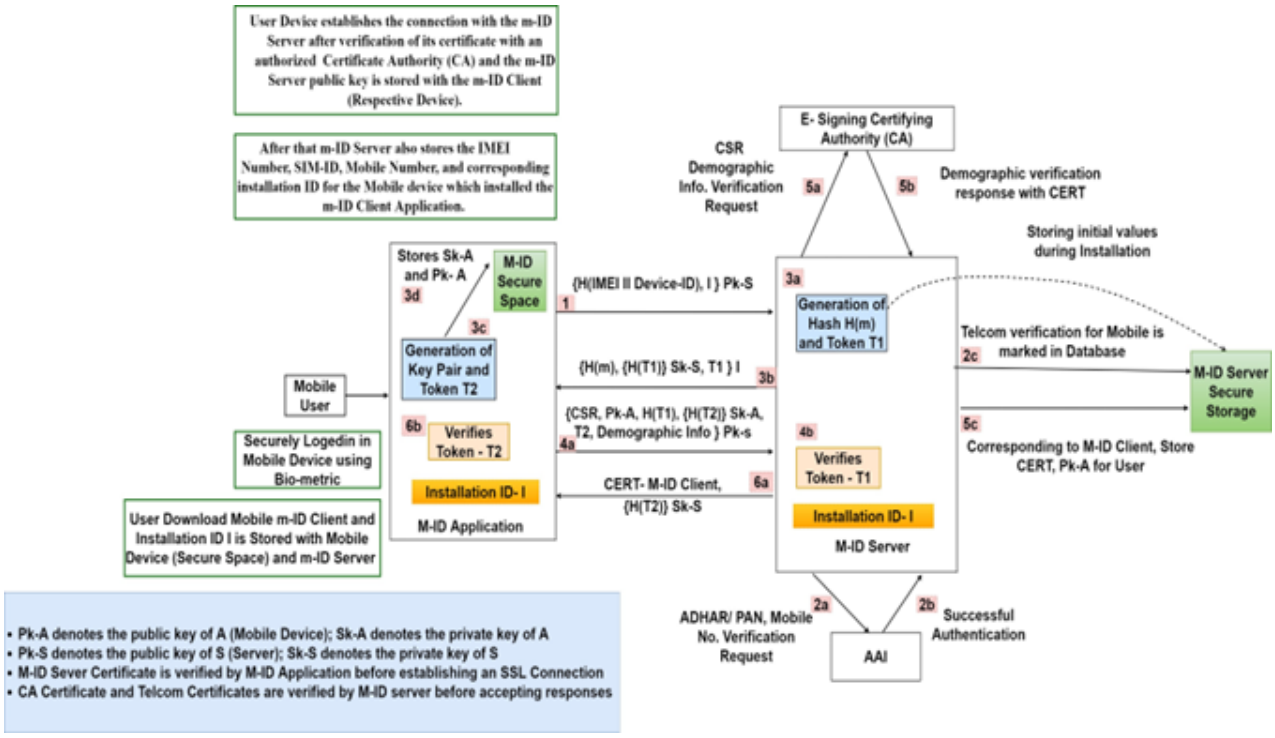


Figure 4.3: Authentication Process

There are two sections to the overall process:

(i) Registration

Essential demographic information is collected during registration, and IIA is used to verify user details. In addition to the demographic information logged, a biometric property, such as face detection, will be used for user authentication and app opening. All interactions with the m-ID app and server are through secure API, and after successful m-ID is issued and saved in the user's Mobile ID in the device secure area.

- Initially, when the user downloads and installs the m-ID Client App on his or her mobile, an installation ID 'I', gets downloaded and stored in its m-ID Client secure space, and kept with m-ID Server for the corresponding device. Mobile Device(A) has a function 'H' is also stored, which is used to calculate normal hashes; This 'H' is different than 'H(m)', which is later sent by m-ID Server to m-ID Client after the a certain verification as given below.

- The user device establishes the connection with m-ID Server after verification of its certificate with an authorized CA, and m-ID Server public key is stored with m-ID Client.

- At the same time, m-ID Server has been set up to store the Device ID, "IMEI Number", and corresponding unique installation ID for the Mobile device that installed the m-ID Client Application and the Mobile Number. Detailed steps for the authentication procedure are shown in Figure 9.

(ii) Authentication Phase Figure 4.3 shows the authentication process.

1. The m-ID app first sends the "hash of Device-ID and IMEI, " along with installation ID 'I'; The entire communication is encrypted using the m-ID Server public key.

2a. & 2b.m-ID Server duly verifies Mobile device details with Telecom Operator.

2c. After successful verification, it is marked in the database for the corresponding Mobile Number and Installation ID.

3a. m-ID Server then generates a hash function 'H(m)' and a token T1.

3b. m-ID Server then sends this hash function along with token T1 to the user's m-ID Client, where a signed hash value of token T1 is also sent along with the message. The entire message is encrypted by installation ID 'I', which is a shared secret between m-ID Client app and m-ID Server and is presumed secure. m-ID Client stores the H(m) and T1, after signature and hash verification.
3c. At m-ID Client, KGC generates a Key Pair and a token T2.
3d. Key Pair (sk-A, pk-A)is stored in the Mobile Device secure space.
4a. m-ID App then generates CSR (includes the public key of m-ID Client) and again sends it to m-ID Server along with demographic information (as per applicable GoI standards compliance): this message also contains a hash value of T1, and a signed hash value of token T2, along with token T2. The entire communication is encrypted using the m-ID Server's public key.
4b. m-ID Server verifies token T1 that it has previously sent to m-ID Client.
5a. After successful verification, the m-ID Server requests the Identity Verification Authority (CA) to validate the user's demographic information.
5b. Identity Verification Authority (CA) verifies the demographic of the user and sends a signed response to m-ID Server along with CERT for the user. Identity Verification Authority (CA) signature is verified by m-ID Server.
5c. If the response is valid, the m-ID Server stores the CERT of the m-ID Client user and its corresponding public key in a secure space.
6a. The m-ID Server then sends the CERT of the m-ID Client to the user, along with the signed hash of token T2, to the m-ID Client.

6b. m-ID App verifies m-ID Server signature and verifies token T2 w.r.t. value from signed hash. m-ID Client then stores CERT in its secure space.

4.7 Experimental Findings

In this study, we created a mobile identification framework system. The different components of the structure are interconnected. The complete set of tests is run in Android Studio on the Android OS. This simulation tool is connected to the CA. We put the CA into practice, utilising the free EJBCA CA.

For the mobile device on which the client application has been installed, this server stores the token together with the IMEI number, Device ID, and the specific related application installation ID. When a predefined period has passed since the token's creation, it is declared invalid. The active token received on the client application on the user's device is invalid if the token comes from a different source not from the m-id server. The token loses its validity immediately after being used. Only the m-ID application can utilize it. To implement the design, Java REST APIs are utilized to ensure a secure connection between the KGC server and the m-ID application. The recommended encryption methods were also measured and shown using an Android app.

The results of our proposal are compared to those of the RSA, ECC, and ECC+AES standards [36]. The comparison was made for the following mandatory activities: encryption and decryption, creating private keys, and creating public keys.

- Encryption and Decryption: Time is a crucial parameter of any encryption and decryption scheme to prove their applicability in the piratical scenario. Figure 4.4 shows the comparison of considered schemes. Here, RSA uses the most time in this situation. Comparing our approach to conventional ECC and ECC+AES, it performs well despite being lightweight. This occurs as a result of our small adjustments to the ECC scheme's default operation.

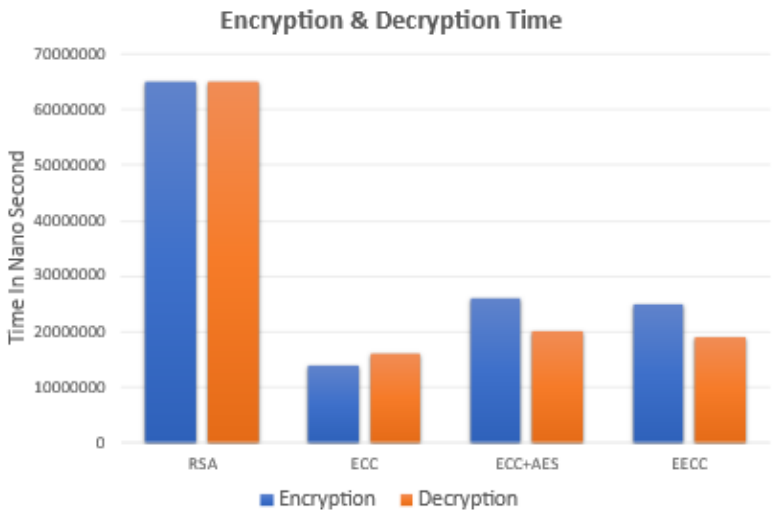


Figure 4.4: Encryption and Decryption Time

- Private Key Generation Time: The time required to produce a private key is highest with RSA. However, with all the other schemes, this time is significantly low. Compared to RSA methods, our approach has observed that the production of the private key takes about 70% less time. Please refer to figure 4.5.

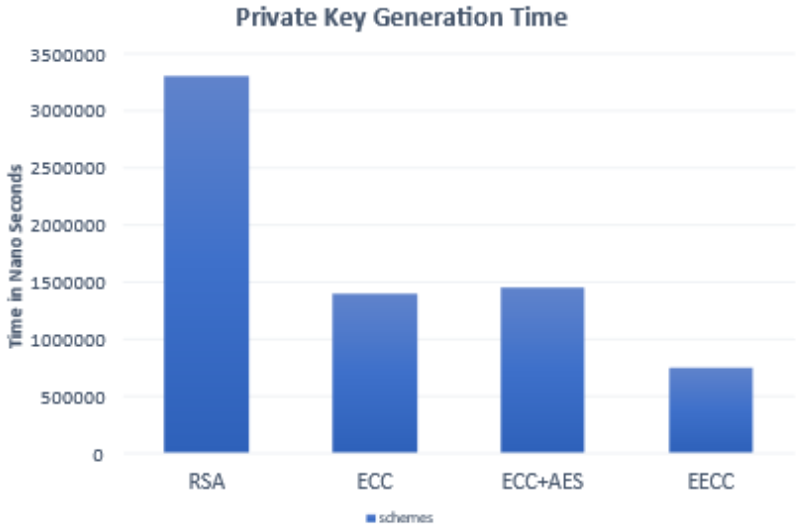


Figure 4.5: Private Key Generation Time

- Public Key Generation Time: In all the three considered schemes the ECC+AES hybrid scheme takes a much longer time because of its heavy and complex computations. However, our technique offers a slightly reduced optimum time to produce a public key when compared to RSA and ECC. However, the length of time required for key-pair production depends on the random selection of large numbers and points on curves. Figure 4.6 displays all the findings.

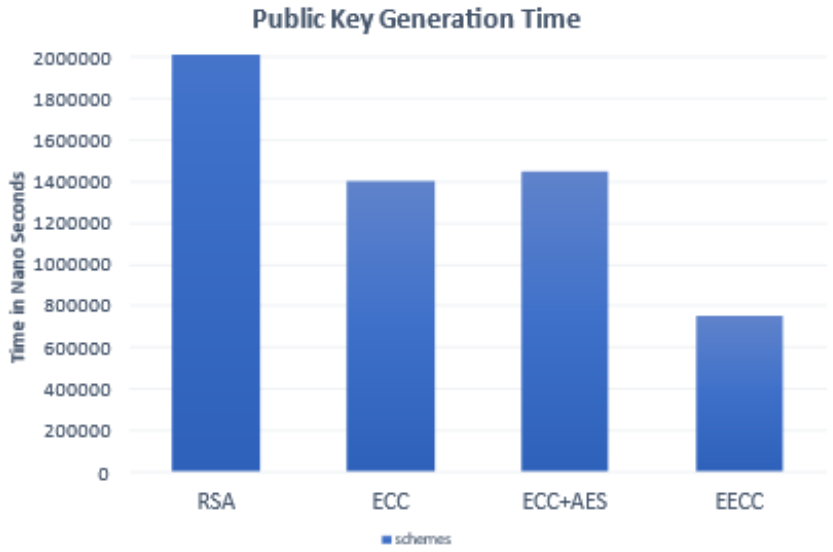


Figure 4.6: Public Key Generation Time

4.8 Comparative Analysis

4.8.1 Comparative analysis of various ECC Curves

This section provides an analysis of the experimental configuration and the conditions in which the experiments were conducted. **Android Studio**

The Android Studio is used for a development environment designed for app. The Android SDK supports native app development. It offers a selection of compilers for cross-compiling C programs to enable Android mobile device operation. Using .adb file (Android Debug Bridge) the apps were loaded onto the phones and executed.

Devices Used

Android devices are equipped with many CPU cores that have different processing rates. The table indicates the speed of the algorithm in the

different device's CPU. (screenshots included).

Running the Programs

Using the Android Debug Bridge, the apps were loaded onto the phones and executed (adb). To utilize it, phones had to have developer settings activated. A similar process is used to test the remaining three phones. The apps were then repeatedly run on the phones under examination to check how consistent the outcomes were. They were trustworthy most of the time. While not greatly, some runs varied more than others.

Results

This section presents and analyzes the findings obtained on various devices utilizing elliptic curves secp256r1. The results are summarized in Table 1. The processor was likely throttled to save battery life, which resulted in this oddity. The other metrics were taken on the phone when the battery was at least 10 percent full.

```
I/System.out: privateKeyEC Private Key
I/System.out:              S: b53fc4b1984efb388199fc391461699f50b55d55b99d957099e822fd998a5013
I/System.out: publicKeyEC Public Key
I/System.out:             X: b5b77ff6881fd58bc6a11667ea2bec2f8da3a28a053bf56136f032b6da6cdbe
I/System.out:             Y: 1562582511993569aec94701037329612008 3ed1c964173be4b7e5bae846f1f0
D/Key Generation: Execution Time: 343 ms
D/DeviceName: samsungSM-A217F
D/Android OS Version: 12
```

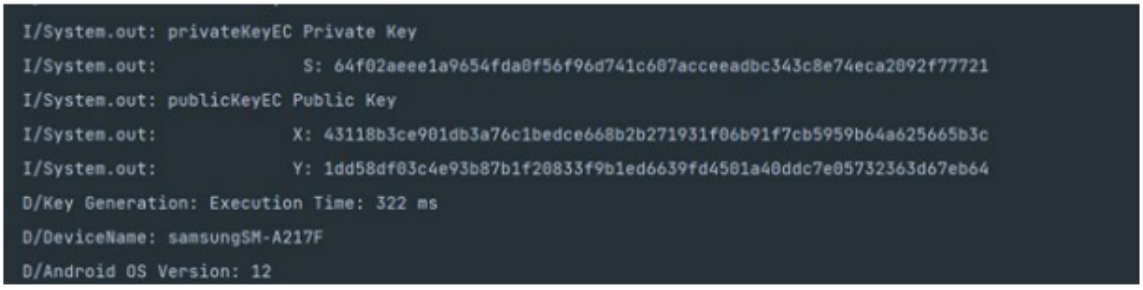
Android Version 12

```
I/System.out: privateKeyEC Private Key
I/System.out:              S: 6a538970d5aed2631b46d41cf445d1d377696c1bee7dc7d8d6e0ee6e87f800be
I/System.out: publicKeyEC Public Key
I/System.out:             X: 24d482e640fff9dbabc6fc732e98b7439defb0727c07e268a8e2f3f803979f4
I/System.out:             Y: d01636bf7261b23eca1b7d0db13042b6d14bf90f02aa5f7dd210a8ea965e575e
D/Key Generation: Execution Time: 117 ms
D/DeviceName: motorolamoto g82 5G
D/Android OS Version: 12
```

Android Version 10

Figure 4.7: Results of Android Versions

Time for secp256r1



Time for secp256k1

Time for Curve25519

Figure 4.8: Results of Different Curves

One PKC scheme that will satisfy these requirements is the ECC. Our research looks at the security and efficacy of ECC as well as its use in such a restricted environment. This study compares several implementation strategies in order to evaluate the efficacy of ECC. Finding the best strategy to provide digital signing, authentication, and user identity verification on mobile devices is the main goal. subsequently,

a mobile application evaluation of the effective strategy is conducted. ECC encryption and decryption are tested and implemented on the user module to see whether it can handle all restrictions and offer excellent security.

This section presents and analyzes the findings obtained on different devices utilizing elliptic curves secp256r1.

Chip Set	Processor Specification	Android Version	KG	CSR
exynos850	Octa-core (4x2.0 GHz Cortex-A55 & 4x2.0 GHz Cortex-A55)	Versions 12	343ms	124 ms
qcom	Octa-core (2x2.2 GHz Kryo 470 Gold & 6x1.8 GHz Kryo 470 Silver)	Versions 10	147ms	132ms
qcom	Octa-core (2x2.2 GHz Kryo 660 Gold & 6x1.7 GHz Kryo 660 Silver)	Versions 12	117ms	105ms

Table 4.1: Comparative Results

The results are summarized in the table 4.2 displays the experiment results. Using secp256r1 and secp256K1, KG required 323–325 ms. On the other hand, Curve25519 produces far faster results.

Table 4.2: Comparative Results

Chip Set)	Processor specification	Android Version	Execution Time
exynos850	Octa-core (4x2.0 GHz Cortex-A55 & 4x2.0 GHz Cortex-A55)	Version: 12	secp256r1- 325ms secp256k1-322ms Curve25519- 44ms

4.8.2 Comparative Analysis of the different solution

The table compares the existing mobile identity solution with the proposed solution based on (i) Key Generation, (ii) SE (Secure Element) used, and (iii) Telcom Service Provider (TSP) establishment. (iv) External hardware is dependent on, etc.

Table 4.3: Comparative Analysis of the Different Solution

Functionality	**Mobiilivarmenne (Finland)**	**Asan Imza (Azebaijan)**	**MobilImza (Turkey)**	**Our**
Key Generation on device	NO	NO	NO	YES
Key Storage on device	NO	NO	NO	YES
Signing	YES	YES	YES	YES
User Biometrics	NO	NO	NO	YES
User Control	No	NO	NO	YES
Without (TSP)	NO	NO	NO	YES
Secure Transaction	YES	YES	YES	YES
Bind with Phone	No	NO	NO	YES

4.8.3 Comparative Analysis of the different OS Versions

The table compares the algorithms run on different Android OS versions for our proposed mobile identity solution.

Table 4.4: Comparative Analysis of the Different Android OS Versions

OS	RSA		ECC	OUR	
	Key Generation	CSR Generation	Key	Key	CSR
10	702 ms	507 ms	475 ms	480 ms	80 ms
12	872 ms	697 ms	390 ms	330 ms	111 ms
13	539 ms	147 ms	681 ms	663 ms	170 ms
14	374 ms	97 ms	376 ms	357 ms	83 ms

4.9 Conclusion

This chapter describes a mobile PKI scheme to solve the problem of higher computation, communication, and storage overhead inside the mobile device. These numerical comparisons underscore each algorithm's varied strengths and weaknesses, emphasizing the importance of selecting the most suitable algorithm based on specific network characteristics and security objectives. Conclusively, based on the extensive testing and findings, the mobile identity Framework was designed. The computational cost of the EECC-based framework is greatly reduced because it generates only one key and only one encryption operation within the device itself. The computational cost

proposed framework based on EECC is significantly reduced because it generates keys inside the device for an encryption operation. The communication cost of the framework is also reduced since there are no dependencies on external hardware, and it's free of telecom operators and SIM manufacturer dependency, in contrast to conventional identity systems that mainly rely on third parties to generate keys on external devices without user control. The recommended framework is secure against various assaults and ensures the confidentiality of personal information. The theoretical and empirical analysis clearly shows that our framework is a better solution for mobile Identity. Also, storage for shared keys on user devices with full user control in the generation and revocation of keys.Overall, this chapter contributes to the field of mobile based user identity by proposing a robust framework that use for user authentication through mobile devices.The framework improves the efficiency and effectiveness of the authentication process, Securing user digital identity while accessing services on mobile devices with robust security measures.

Chapter 5

Mobile-based Signing (mCK) Solution for Mobile Devices

5.1 Introduction

Citizens with personal IDs frequently use cell phones for online benefits and transactions. A digital identity is required in electronic transactions for four reasons: authenticity, confidentiality, integrity, and non-repudiation. The physical ID was not intended to meet these requirements, but it does so in a straightforward and non-purist way.

Mobile-based user digital identity is crucial and incredibly convenient. It empowers users to verify their identities in online transactions, adding a layer of control and security. Given the widespread use of mobile phones in India, there is a clear demand for a mobile user identity solution. Mobile phones are more than just calling devices; they can also serve as gateways to digital identities, making transactions seamless and user-friendly.

Mobile ID can be integrated into modern authentication methods as a digital identity tool, and it can be used as a user ID with other IDs, depending on the sensitivity and assurance required for services. The primary goal of mobile identity is to provide digital user identification solutions that can protect personal information on handheld devices during authentication and transactions when using devices. Many apps rely on mobile phones for everyday transactions and communication, including gaming, banking, email, social media, photos, e-commerce payments, etc. Many companies,

institutions, and sectors use mobile devices to authenticate users. Providing a fake ID during online transactions is possible from a mobile device, leading to fraud and danger to mobile security.

mCK (Mobile certificate) for signing. We demonstrate that our proposed key generation and signing are hardware-independent, low-cost, and scalable. Furthermore, we present the theoretical analysis to establish its practicality. This proposed solution will help reduce dependence on SIM or HSM-based approaches.

The proposed solution is a mobile-originated PKI. The first advantage is that a mobile device will have an OS-dependent application, a soft key pair generator, and a digital signer. The second advantage of this invention is that it does not need extra hardware or devices to store private keys.

5.2 Motivations

The majority of available solutions are storing keys or certificates on servers or SIMs. All solutions rely on external hardware. The present approach has multiple challenges, some of which are given below:

- When doing transactions via a mobile device, there is a chance of fraud and giving a false identity.
- It is extremely difficult to generate key pairs on mobile handheld devices. To do this, lightweight cryptography algorithms must generate key pairs and certificates inside the device.
- The need for an alternative to key pair generation in the SIM card. Building on this highlighted gap, we suggest using a mobile device's memory rather than SIM cards to store user identity.
- Telecom service providers provide the SIM-based identity, and the entire security architecture depends on TSP, increasing reliance on telecommunications firms.
- In a SIM-based solution, the user has no control over keys and certificates because they are bound by TSP.

5.3 Contributions

We conducted an in-depth analysis of prevalent Mobile ID systems, identifying critical flaws such as vulnerabilities in authentication protocols, susceptibility to identity theft, and inadequate encryption mechanisms.

- We developed a PKI-Based solution offering a robust framework for mobile identification. This approach ensures heightened security, streamlined authentication processes, and enhanced user privacy compared to conventional systems.
- We proposed that a mobile device be used to obtain an x.509 certificate from a trusted Certificate Authority (CA) for signing transaction/data.
- We proposed adding additional Parameters in the x.509 certificate. The CA will issue the x.509 certificates with additional parameters (IMEI no and device ID) and user details.
- We provide validation through empirical experiments to rigorously test and validate the proposed PKI-based Mobile ID system. The experimental trials yielded tangible results, demonstrating the system's resilience to security threats, seamless integration with mobile platforms, and effective safeguarding of user identities across diverse operational environments.
- We presented concrete experimental findings to underscore the practical viability and efficacy of the solution. Empirical evidence showcased the system's potential to address shortcomings in Mobile ID systems, paving the way for enhanced security and reliability in mobile authentication practices.

5.4 Proposed Approach for Signing Process through Mobile Device

The proposed solution provides a secure authentication system for performing transactions such as banking, shopping, payments, and availing public services from mobile phones at any time and anywhere. The suggested solution is based on Smartphones with PKI technology and does not require any extra hardware or SIM cards. This technique stores the user's private keys and certificates in the SE area. A secure location will be created within the device to save confidential data. Due to the ingenious

design, it is cost-effective, secure, and interoperable solution. Figure 5.1 demonstrates the signing cycle in mobile devices. An authentication server that connects a CA, a mobile device, and a service provider (SP) enables data transmission between processing systems.

In the system, a mobile device as a soft key-pair generator generates the key-pair SK, PK using the mobile OS-dependent software and saves variables in the scope of the software program for the period of a complete cycle of the transaction.

The service provider (SP) requests the mobile device "d" to sign the data mSignREQ dk. "d" submits a request CertREQ PK to the trusted CA's for a valid certificate. Trusted CA's generate the X509Certificate in response to request CertREQ PK processed by "d". The "d" signs the data "dk" with SK and sends back response mSignRESP ds with signed data "ds." Then SP requests signAuthREQ ds to the authentication server to verify and authenticate the signed data "ds". The authentication server verifies the requested sign data "ds" and sends the response signAuthRESPds to the SP. SP responds according to the response received from signAuthRESP ds and finally sends acknowledgment signACK to "d".

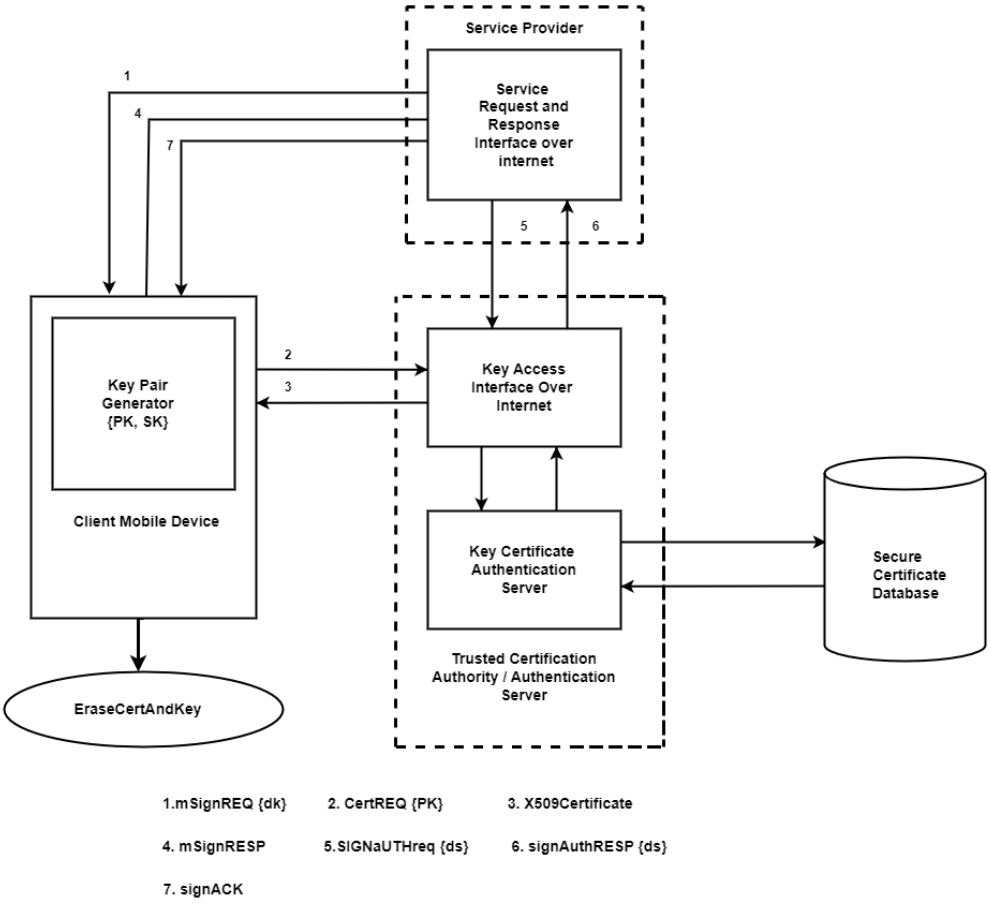


Figure 5.1: Authentication Process

Version
Serial number
Algorithm ID
Issuer
Validity
Subject
Public Key Info
Public Key Algorithm
Subject Public Key
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Mobile ID Unique Identifier
Device ID
Certificate Signature Algorithm
Certificate Signature

Table 5.1: Structure of m-ID X.509 Certificate

5.4.1 Mobile as a Signature Solution

The proposed solution can also be used for mobile signature (mSign) through mCK, for signing documents and transactions using mobile devices. The proposed approach differs from other approaches that use online signing, wherein our approach is key generation and signature creation on the user's mobile device after verifying by IIA. Figure 5.1 illustrates the architecture in public-key cryptosystems. The sender first signs the message using their private key before encrypting it with an ephemeral session key to ensure its confidentiality, integrity, authenticity, and non-repudiation.

- Public verifiability: Can verify the authenticity/validity of signcryption without private key, the original message.
- Ciphertext authentication: An external judge can use the components of the ciphertext and the receiver's intermediate decryption results to confirm the message's origin. The recipient does not need to reveal the secret key or the original communication to ensure non-repudiation.

- Public ciphertext authentication: An outside party can verify The message's origin without the recipient's involvement.
- Ciphertext anonymity: The encrypted elements cannot be utilized to derive valuable information about the sender. It is essential to understand that it is impossible to establish both public ciphertext authentication and anonymity simultaneously.

5.4.2 Certificate authority (CA)

A CA is an organization that uses a public key (PK) to verify and validate an identity. The CA produces digital identities through the issuance of certificates. Certification is a legally binding process that involves the creation of a signed data structure called a PK certificate. A CA, or Certification Authority, is an expert in the process of certifying anything. The issuing CA employs digital signatures to guarantee the integrity of certificates. The issuer and the certificate owner must possess confidence for this to occur. A CSP is required to offer security management services, encompassing the production and storage of cryptography keys.

Validation Authority (VA)

The term VA refers to the competent authority that automatically accepts or validates a certificate issued by the CA. A VA monitors the CRL provided by CAs and provides OSCP (online certificate status protocol) functionality.

Certificate Revocation

Certificate revocation refers to invalidating a certificate before its scheduled expiration date. If a certificate's private key is suspected to be compromised, it is imperative to cancel it promptly. A domain should be revoked when it becomes inactive. A certificate revocation list (CRL) is a comprehensive record of all certificates revoked by a CA.

Proposed Additional Parameters

The system initiates the process by generating a key pair. It then compiles the public key encryption data into a CSR. This CSR is then sent to a trusted CA to obtain x.509. The CA will issue the x.509 certificates with additional parameters (IMEI no and device ID) and user details.

5.4.3 Signing Steps

- A Proposed solution based on PKI originated from a mobile device for verification and authentication of the signed data, characterized by a solution for transmission of data securely over the world wide web, as also shown in Figure 5.1;

    a. Mobile devices are used as key pair generators (SK, PK).

    b. A mobile device is used as a digital signer for data signing.

    c. The mobile device is used for establishing internet communication with the Trusted Certification Authority (CA) as an x509 certificate provider and authentication server for decryption and authentication.

- A method for PKI originated from mobile devices for facilitating electronic certification for verification and authentication of the signed data using the proposed solution comprising the steps;

    a. Download the mobile ID application on a mobile device. The system will generate the key pair and send the CSR containing public key encryption data to the trusted CA to obtain a certificate.

    b. After creating a key pair and receiving a signing request from the SP, Sign the document by private key and complete the transaction.

    c. Generating the X509 certificate request to CA from a mobile device after successfully verifying the mobile device user credentials.

    d. Signing the request received from the service provider with the x509 certificate and sending the signed data back to the SP.

    e. Verifying the signed data by the SP for signature authentication request to a trusted certification authority;

    f. Receiving, on the service provider, a response from the trusted certification authority and sending acknowledgment on the mobile device;

    g. The user has the choice to erase it on the mobile device upon receiving acknowledgment from the service provider or not to erase it.

h. In our proposed solution, users fully control key pair generation and revocation. Based on the service request from SP, the user will perform the activity.

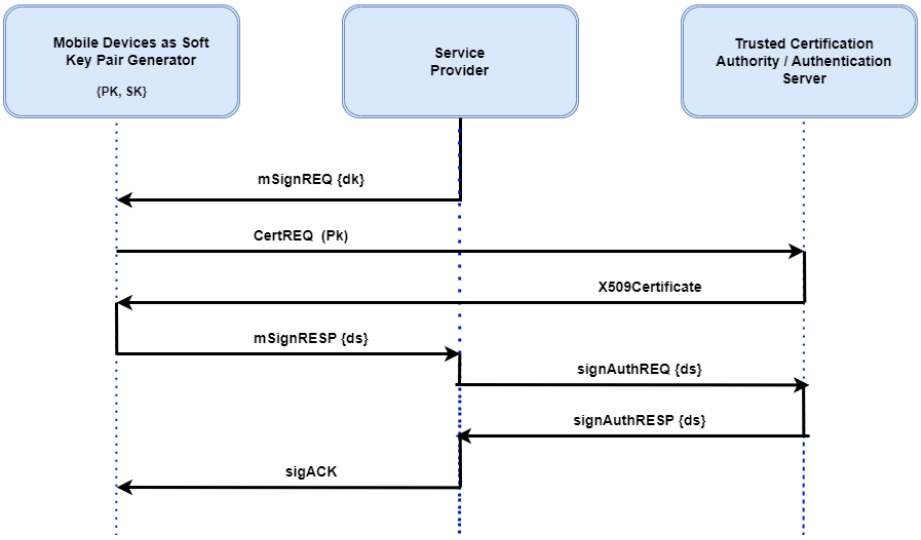


Figure 5.2: Flow of m-ID X.509 Certificate

5.4.4 Secure key stores in the device such in secure element

Our approach ensures a high level of security by limiting the use of keys for cryptography operations to those stored in the keystore. A TEE might restrict key usage to cryptography modes or require user identity. According to our method, hardware-level security measures, like a TEE, should safeguard the keystore. The user also has the option to erase their private key after the transaction is completed based on the service request, further enhancing the security of our system.

5.5 Security analysis

In this section, we present the security analysis of our proposed solution.

- Anonymity and privacy: Our approach involves users enrolling by providing their information through an Android device. In addition, users are verified by IIA. Convent details in hash encryption and storage on the server that minimize the possibility of fraudulently acquiring or disclosing user information. Furthermore, the EECC and TEE have been used to ensure high levels of security and privacy.

- Preventing replay attacks: We designed and devised a method to avoid and prevent replay attacks by collecting identity information during user registration. If the same request is made again, that user will be permanently prohibited with their unique ID, ensuring that our system remains secure the next time that user or attacker is barred.
- Protecting users from collusion attacks: if attackers and adversaries combine to attack our m-ID server and server compromise. Attackers will get the hash value of the user's identity, not any key (private or public), preventing them from generating and verifying the correct certificate. Furthermore, this step can safeguard our m-ID system.
- Protecting the m-ID server failure: In case our system fails, stops, is infected, or is compromised at any time. It will also notify the user to stop using the app for a repair time interval. Furthermore, the fundamental advantage of our solution is that key generation and storage are done on the device side therefore, any server attack will not affect keys.
- MITM attack: Our methods use a randomly generated nonce, a hash function, and ECC. An adversary cannot change the message to get past authentication methods since they require users and servers to have a secret key or private information. Thus, there is protection against a MITM attack for the suggested technique.
- Privileged insider attack: The insider attacker would find it difficult to figure out the password because of the discrete logarithm problem, biometric information prediction, and one-way hash function reversal.
- Known session key secrecy: The proposed technique has token generation for every request, and the token is valid for a limited time. An attacker can not access critical information because of token validity.
- User impersonation attack: The attacker required a PIN code to open the mobile app.
- Impersonation attack on server: The attacker requires the server's key to start a server impersonation attack. As a result, the proposed method protects against server impersonation attacks.
- Ephemeral Secret Key Leakage Attack: A session keys may be compromised under this attack if attacker A discloses a transient session method; however, adversary A

cannot do so.

5.6 Experimental Results

This section provides the details of the experiments and performance analysis.

5.6.1 Verifying mCK Signing and Mobile Based Identity through ECC algorithm

We implement the design using Java REST APIs to facilitate a secure connection between the m-ID app and the m-ID server. An Android application was also developed to demonstrate and evaluate the proposed encryption process. We will now describe and compare the results of our approach with those of RSA and ECC. We provide nanosecond time (ns) for encryption, decryption, and key pair generation. Paar et al., 2010, Koblitz, 2000, Rehman et al., 2021.

E(m) = EndTime - StartTime

Where,

E(m) = encryption of user information.

StartTime= The starting time of encryption.

EndTime = The ending time encryption.

ECC algorithms have significantly improved the existing techniques.

Figure 5.3: Signing and verification time

Figure 5.3 shows our solution reduces encryption and decryption time as compared with traditional RSA, ECC algorithms, etc.

Figure 4.6 shows our solution compares the public key generation time with RSA, ECC and Fugjeaw et. al. Paar et al., 2010 It also shows the minor improvement of the ECC-based encryption technique.

Figure 4.5 shows our solution compares the private key generation time with RSA, ECC, and another algorithm. The time was reduced, and performance increased.

5.7 Conclusion

In conclusion, the proposed Mobile ID framework offers a promising avenue for implementing secure online signatures through Mobile Sign (mSign) via Mobile mCK certificate while prioritizing security, usability, confidentiality, integrity, authenticity, and non-repudiation. By leveraging the widespread usage of smartphones, this solution eliminates the need for cryptography tokens, making it more accessible and convenient for users. The mSign architecture, with its centralized Mobile identity repository, streamlines processes such as user authentication, registration, and verification, providing a cohesive and efficient framework for digital signing using mobile devices. This approach stands out from others in its minimal user involvement, particularly in key generation and signature creation processes, thus enhancing the overall user experience. The core concept of Mobile ID revolves around developing after successful user authentication from the IIA, users are seamlessly registered and empowered to create their key pair. Subsequently, they undergo certification through a CA. This streamlined process ensures the security and reliability of digital signatures. The CA will append additional parameters related to the user's mobile device in the x. 509 certificate. In summary, the proposed approach offers substantial development in online signature techniques. It provides a secure and user-friendly alternative that takes advantage of the widespread availability of mobile devices. Combining security and usability improves digital signing experiences in our increasingly interconnected society.

Chapter 6

Trusted and Secure Storage on the Mobile Device for Securing Private Keys

6.1 Introduction

A mobile ID solution built on PKI does not require any additional hardware. In this strategy, user information such as demographics will be stored in a secure space (SSS), which will be incorporated into the device to contain the secret data. This novel design technique ensures the proposed solution is autonomous, cost-effective, secure, and interoperable. The suggested mobile ID solution requires the user to first register on a mobile app, which collects and verifies the essential demographic data using an IIA.

The user's control over their private key is a crucial aspect of this solution. The second step involves generating a key pair on the user's device, which includes their information and device ID. The keys are stored within the device, such as the Secure Element, and the user retains the private key, ensuring utmost secrecy. The SSS module's primary objective is to establish a protected and impregnable zone within the mobile memory, accessible by the m-ID app only with the user's explicit consent.

The secure space, a robust security measure, stores all data in an encrypted form in a file system that is completely inaccessible to regular OS file traversal techniques. This ensures the user's confidentiality and privacy are safeguarded. The data is only

accessible to the m-ID app, as shown in figure 6.1. After the key pairs are generated, a certificate can be saved in the device's Keystore. Access to a key imported from the Keystore is restricted to authorized users only. The user's biometric information captured will be used for mobile authentication, and the user can revoke keys and generate a new key at any time.

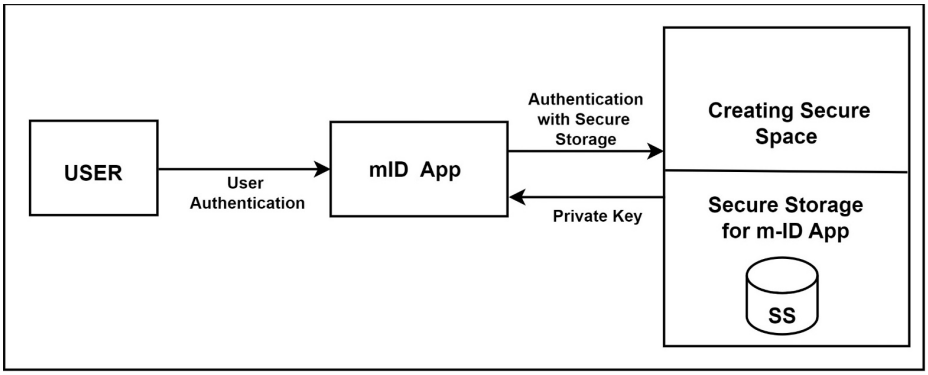


Figure 6.1: Flow of Secure Storage Inside the Mobile Device

Once the key pairs have been established on a mobile device, the next step is to send a Certificate Signing Request (CSR) to the Certificate Authority (CA) to acquire an X.509 certificate to perform online transactions and signing. The figure illustrates the processes of the proposed mobile ID solution for saving the key and certificate SE. 3.3.

1. Registration request submitted by m-ID app to the m-ID server.
2. The m-ID server transfers the user information to send the token to the user app.
3. The m-ID server forwards the PK and CW to the m-ID user.
4. The m-ID user app produces a key pair SK from PK and CW.
5. The generated private key 'SK' is stored in the secure area.
6. After generating the key, the m-ID user can use the portal to access the services.
7. An authentication request is approved by the m-ID server.

List of use cases for which our proposal can be utilized are listed below:

(a) Documents signing.

(b) Signing transactions.

(c) Bank transactions.

(d) e-commerce purchases

(e) Access to health records.

(f) e-government portal.

(g) Tax declarations and payment.

Over the last decade, mobile platforms such as smartphones have been increasing rapidly. Users are growing comfortable using their phone devices for online transactions, entertainment, games, watching movies, and shopping through mobile apps. Consequently, mobile phone manufacturers (OEM) and app developers have been enhancing the security of their applications and platforms. The TEE module of the mobile device provides a context for the execution of critical apps like mobile payment and authentication management, which can be executed independently of the Operating System. TEEs must meet the two conditions listed below:

1. Memory Isolation: Memory isolation protects TEE areas from a kernel. Virtual Ghost is a software-only technique, while ARM TrustZone and Intel SGX are hardware approaches. Software approaches, such as kernel deprivileging or compiler instrumentation, should separate kernel memory from TEE memory.
2. Attestation: Attestation is a way to determine whether an application is executing in a TEE environment. A chain of trust must be established within the system to offer confirmation of attestation.

6.2 Contributions

- A review and evaluation of TEE's design and architecture, as well as a static and dynamic analysis, and a detailed explanation of the links and integration between TEE and the Android operating system.
- Verification mechanisms that load trusted applications. Furthermore, these flaws are a benefit in recovering plain text data from the device. This study analyzes the TEE's runtime Trusted Application loader.

- TEE offers trusted apps with security features such as mobile-based authentication and other trusted applications that provide services to the outside world. Cryptography procedures such as encrypting, signing, and hashing provide (Trusted Authorities) TAs with these capabilities. Sensitive information on mobile phones is encrypted using cryptography to verify the device's identity.
- A prototype for a Mobile ID solution that enables all hardware-backed cryptography to store keys has been developed.

6.3 Preliminaries

6.3.1 Android OS

A widely used, open-source Android operating system is available for smartphones, tablets, and other smart devices. Built on the Linux kernel, it is developed by Google. The source code of Android and its packages is available as open source. Android apps can be developed using Java, Kotlin, and C++. The Android operating system is available to any smartphone user, suggesting that many manufacturers produce Android phones.

1. All apps and services data is saved under the directory "/data".
2. Appstore data is saved in the directory "/data/data".
3. The SD card is data saved in the directory "/sdcard for additional storage.

6.3.2 Hardware-Enforced Isolation

A suggestion has been made to utilize TEE to create space to run cryptography functions and co-processors. The TEE leverages the platform's hardware capabilities to create a secure environment where code execution is isolated from the rest of the system. This isolation ensures that data saved in the TEE is protected from any assaults by external software. Unlike the remaining code in the system, which is labeled as untrusted code, the code residing in the TEE is regarded as reliable.

6.3.3 ARM TrustZone Technology

ARM TrustZone technology divides the global physical memory into secure and insecure sections For example, this can be implemented at each subordinate level, in the memory controller, or a global module. It enables the device to use secure world mode and normal world mode. Every mode provides a distinct execution environment. The hardware and software resources can be classified into memory and peripherals. The crucial software components for security are frequently executed in a Secure World environment. It can access all hardware and software, including those commonly seen daily. Only resources from the hardware and software of the normal world can be accessed. TrustZone technology enables the ARM processor to monitor and control the transition between secure and non-secure environments through its monitor mode. Monitor mode is a component of the secure world.

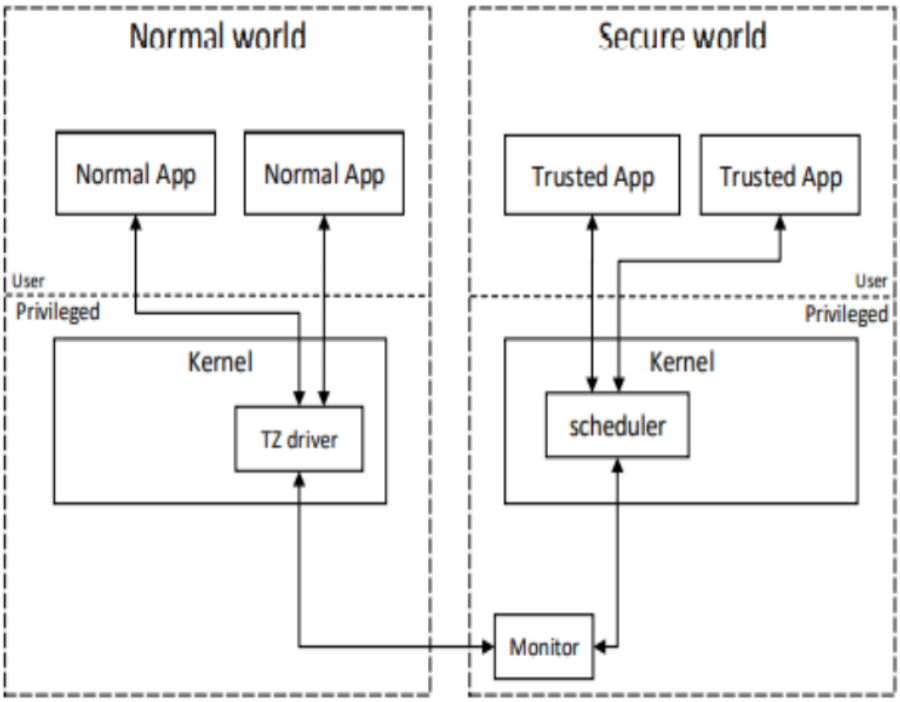


Figure 6.2: ARM TrustZone

6.4 Attacker Model

1. Malicious Application Attacker – The attacker tries to use private keys from another app on the same device. The app is downloaded from an AppStore/marketplace. The attacker is presumed to have full access and full permissions are given at the time of app installation.
2. Root Attacker - The attacker gets access to the root permission and runs the program on the device. our application can not be installed and run on a rooted device.
3. Intercepting Root Attacker - Any instance in which a hacker secretly intercepts and modifies communication between two parties is considered an interception attack.

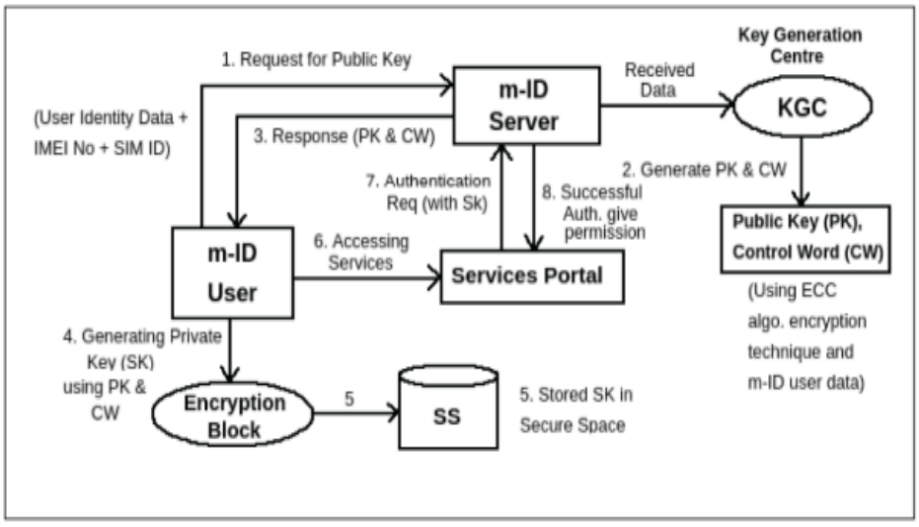


Figure 6.3: Secure Element of TEE

6.5 Security Level

Application security level

The proposed framework encompasses the subsequent security at the application level.

1. Protect user data on application: When building a mobile app, prioritize security, usability, and performance. A secure mobile application can significantly increase client satisfaction.

2. Application permissions granted by user: Mobile app permissions based on security features can support user privacy goals by minimizing data, controlling access, and ensuring transparency.
3. Application Verification: Verifying apps installed on the mobile device originate from a reputable and reliable source/AppStore. Mobile OS will verify programs' digital signatures at the system level.
4. Anti-debugging: To protect app reverse engineering using code obfuscation and hooking techniques.

Framework-level Security

1. To protect system resources (e.g. Network, SMS, GPS, call, photo camera, GPS), apps permission to use. To allow a third-party application to access devices, users must first provide access authorization via Mobile OS Permissions.
2. Key management involves enforcing system security and managing keys appropriately.
3. Secure Containers: A multi-user design allows for the safe separation of professional and personal profiles on a single, unified interface.
4. Device Resource Management refers to effectively managing the constrained hardware and software resources to maximize performance, prolong battery life, guarantee security, and offer a seamless user experience.

6.6 Performance Evaluation

6.6.1 Verifying Trusted and Secure Storage on the Mobile Device for Securing Private Keys

We tested a prototype on the Android device. The m-ID server is built on top of an Apache web server and a MySQL database. We built a secure TEE space on Android, uploading and downloading basic application text in the.txt file; that approach is shown in Figure.

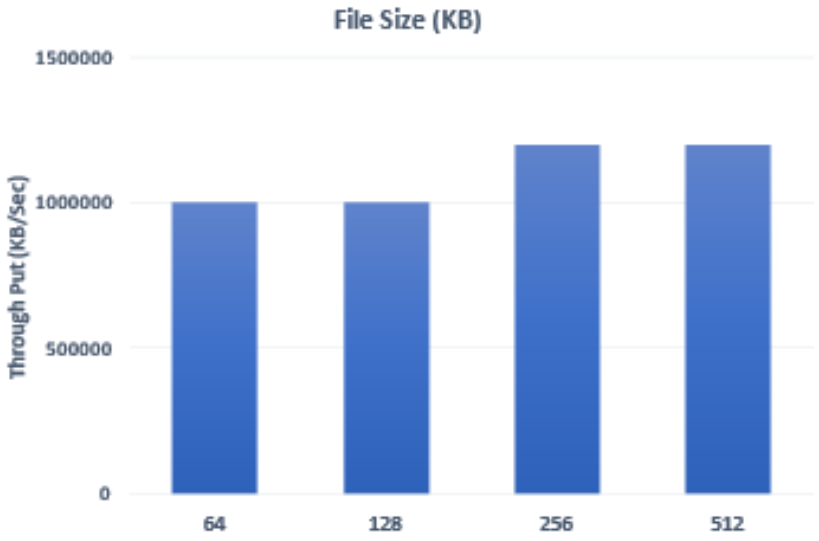


Figure 6.4: Throughput of Secure Element of TEE

We discovered that our idea has a reasonable and stable throughput. However, when file sizes grow, the throughput reduces marginally.

6.7 Conclusion

In conclusion, the rapid advancement of technology in the last decade has witnessed a significant surge in the utilization of mobile technology, particularly phones, for various online activities such as transactions, gaming, entertainment, and shopping through mobile applications. This heightened usage has necessitated a corresponding focus on bolstering the security measures of these platforms and applications by mobile platform manufacturers and developers.

This study examines Trusted Execution Environments (TEEs) have emerged as a critical element in the security architecture of mobile devices. They provide a segregated context for executing security-sensitive applications independently from the rich operating system. This separation facilitates the secure execution of tasks such as mobile payment processing and user authentication management and enhancing overall system security.

As the leading mobile operating systems, Android and iOS have prioritized security measures, with TEEs playing a pivotal role in their security frameworks. These environments are fortified by hardware primitives, ensuring isolation from the rest of the system and safeguarding against potential security breaches.

The advent of Secure Application Execution (SAE) further underscores the importance of robust security architecture within TEEs. Leveraging technologies such as the ARM TrustZone, which provides trusted execution environments for security-sensitive services, mobile device security continues to evolve and strengthen.

In essence, as mobile devices continue to play an increasingly integral role in our daily lives, ensuring their security remains paramount. The utilization of TEEs, backed by hardware primitives and advanced security architectures, represents a significant step forward in safeguarding user data and maintaining trust in mobile platforms and applications amidst the ever-evolving threat landscape.

Chapter 7

Conclusion & Future Scope

7.1 Summary of Findings

This research explores the demand for a novel Mobile Identity framework based on PKI to protect the user's sensitive information and proposes the framework.

This thesis contributions are:

1. Proposed a Framework for Mobile Identity on a Handheld device;
2. Proposed key generation inside a Handheld device;
3. User identity binds with the device.
4. User has full control of Key generation and revocation.
5. Mobile-based Signing (mCK) solution for mobile devices with additional parameters in x.509 certificate
6. Trusted Environment framework for storing private keys on mobile devices;

Globally, various mobile ID solutions are available. All these solutions depend on Hardware Security Module (HSM) systems, SIMs, and card-based solutions involving telecom service providers for key generation and storage. To address this issue, we proposed a secure ECC-based Mobile ID user authentication framework independent of any external hardware or third-party provider with user control and privacy for their information needed.

The design had challenges to overcome, which was

- Identification of cryptography algorithms that will run on handheld devices without compromising the performance.
- Identifying the secure place on handheld devices to store the keys.
- Binds user identity with device
- Performing digital signing on handheld devices.

This thesis aims to present a coherent proposition emphasizing identity and privacy management for mobile devices within user onus and privacy. This framework encompasses methods and processes for procedural handling, security measures as well as privacy aspects, thus allowing people to control their own digital identity without outsourcing to any service providers. To validate the framework, prototypes were developed through which they were able to demonstrate the scope and applicability of the design. The prototypes were also subjected to functional, security, and privacy analysis to ensure the mobile identity management solution met the required functional, security, and privacy requirements.

The work done under this thesis is summarized below:

- Under this research, a demand survey for a need for a Mobile Identity framework based on PKI to safeguard user-sensitive information on mobile devices was done through publications and market available products.
- Identifies limitations of existing mobile ID choices, primarily relying on Hardware Security Modules (HSM), SIM cards, and external hardware, emphasizing the necessity for more flexible solutions.
- In this research work, a comprehensive framework for Mobile Identity on handheld devices is proposed. The approach also includes the generation of key pairs within the device and a Trusted Environment framework for securely storing private keys on the device.
- Proposes a Mobile-based Signing (mCK) solution for mobile devices, addressing the need for adaptable solutions that respect user privacy in online transactions and ensure security.
- Explores possible enhancements in the ECC algorithm to enhance security and implement an identity and privacy management framework for mobile devices with user control over their identity. It addresses trust-related issues and proposes alternative frameworks for key pair generation and storage inside handheld devices.

- Perform informal analysis to demonstrate that our framework can prevent various security attacks.
- We demonstrated that our framework can ensure user authentication and key security using the BAN logic and Scyther verification tool.

7.2 Contributions to the Field

The contributions outlined in the research can significantly impact future developments in mobile identity management and security. Here are some potential contributions to the future:

1. *Enhanced Security Measures*: By exploring novel approaches such as biometric data and Elliptic Curve Cryptography (ECC), the research lays the groundwork for future advancements in mobile device security, especially for storing confidential data. Implementing these measures could lead to more robust protection of user-sensitive information against evolving cyber threats.
2. *Improved Privacy Protocols*: The proposed identity and privacy management framework offers users greater control over their data, paving the way for future privacy-conscious mobile applications and services. This could foster increased trust between users and service providers in the digital ecosystem.
3. *Adaptable Mobile Identity Solutions*: The research highlights the need for adaptable mobile identity solutions that will not rely on external hardware or telecom service providers. Future developments inspired by this research may lead to more flexible and user-centric approaches to identity management, accommodating diverse user needs and preferences.
4. *Shift Towards Device-Centric Solutions*: By suggesting frameworks for generating and storing key pairs within handheld devices, the research challenges the reliance on SIM-based identity management systems. Future implementations of device-centric solutions could empower users with greater autonomy over their digital identities while reducing dependencies on third-party providers.
5. Advancements in Trust Models: The examination of trust-related issues and underlying assumptions contributes to a deeper understanding of trust dynamics in mobile

identity management. Future research may build upon these insights to develop more robust trust models that adapt to evolving threats and user behaviors.
Overall, the research's contributions have the potential to shape the future landscape of mobile identity management by prioritizing security, privacy, and user empowerment. These advancements could lead to more resilient, user-friendly, and privacy-preserving mobile identity solutions.

7.3 Significance and Prospects for Future Research

The significance and prospects for future research stemming from this work are multifaceted and impact. Here are some key points:

1. Enhanced Mobile Security: This research addresses critical issues in mobile security by proposing novel approaches to mobile identity management. Future research could delve deeper into refining these approaches, exploring additional security measures, and testing them in real-world scenarios.
2. Enhanced cryptography algorithms: Future research could focus on designing cryptography algorithms that are immune to attacks by quantum computers.
3. Exploring blockchain-based and other decentralized approaches to manage mobile identity.
4. Investigating the role of 5G technology in enhancing mobile identity solutions.
5. Standardization and Interoperability: As mobile identity management evolves, standardization and interoperability become increasingly important. Future research could focus on creating mobile identity solutions that work across different platforms, ecosystems, and international borders.
6. Investigating the role of mobile identity in the Internet of Things (IoT) devices.

Mobile identity research continues to offer exciting and critical opportunities for technological advancement. Future work must address both the evolving landscape of cyber threats and user needs for security, privacy, and convenience. Emerging technologies like AI, blockchain, and quantum computing will have a profound impact on how mobile identities are managed and secured.

Bibliography

Albakri, A., Harn, L., & Maddumala, M. (2019). Polynomial-based lightweight key management in a permissioned blockchain. *2019 IEEE Conference on Communications and Network Security (CNS)*, 1–9.

Al-Khouri, A. M. (2011). Pki in government identity management systems. *arXiv preprint arXiv:1105.6357*.

Al-Mahmud, A., & Morogan, M. C. (2012). Identity-based authentication and access control in wireless sensor networks. *International Journal of Computer Applications*, *41*(13).

AlMajed, H., & AlMogren, A. (2020). A secure and efficient ecc-based scheme for edge computing and internet of things. *Sensors*, *20*(21), 6158.

Aminuddin, A. (2020). Android assets protection using rsa and aes cryptography to prevent app piracy. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 461–465.

Arabo, A., Shi, Q., & Merabti, M. (2009). Context-aware identity management in pervasive ad-hoc environments. *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, *1*(4), 29–42.

Assefa, S., Porth, B. W., & Shank, S. M. (2016). Integrated photodetector waveguide structure with alignment tolerance [U.S. Patent Application US 10/090422 B2]. *U.S. Patent*, (US10090422B2).

Batina, L., Örs, S. B., Preneel, B., & Vandewalle, J. (2003). Hardware architectures for public key cryptography. *Integration*, *34*(1-2), 1–64.

Bhargav-Spantzel, A., Camenisch, J., Gross, T., & Sommer, D. (2006). User centricity: A taxonomy and open issues. *Proceedings of the second ACM workshop on Digital identity management*, 1–10.

Bicakci, K., Unal, D., Ascioglu, N., & Adalier, O. (2014). Mobile authentication secure against man-in-the-middle attacks. *Procedia Computer Science*, *34*, 323–329.

Böger, D., Barreto, L., Fraga, J., Urien, P., Aissaoui, H., Santos, A., & Pujolle, G. (2014). User-centric identity management based on secure elements. *2014 IEEE Symposium on Computers and Communications (ISCC)*, 1–6.

Boontaetae, P., Sangpetch, A., & Sangpetch, O. (2018). Rdi: Real digital identity based on decentralized pki. *2018 22nd International Computer Science and Engineering Conference (ICSEC)*, 1–6.

Cai, L., Yang, X., & Chen, C. (2005). Design and implementation of a server-aided pki service (sapki). *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)*, *1*, 859–864.

Chandrashekhara, J., Anu, V., Prabhavathi, H., & Ramya, B. (2021). A comprehensive study on digital signature. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN*, 2347–5552.

Chudamani, M., & Tatini, N. (2019). An improved cryptographic key generation and data transmission technique using images. *Int J Innov Technol Explor Eng*, *8*(6), 1797–1800.

Cooijmans, T., de Ruiter, J., & Poll, E. (2014). Analysis of secure key storage solutions on android. *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, 11–20.

Dunphy, P., Garratt, L., & Petitcolas, F. (2018). Decentralizing digital identity: Open challenges for distributed ledgers. *2018 IEEE European symposium on security and privacy workshops (EuroS&PW)*, 75–78.

e-Estonia. (n.d.). E-Estonia - We have built a digital society & we can show you how — e-estonia.com. https://e-estonia.com.

El Haddouti, S., & El Kettani, M. D. E.-C. (2019). Analysis of identity management systems using blockchain technology. *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 1–7.

Enck, W., Ongtang, M., & McDaniel, P. (2009). Understanding android security. *IEEE security & privacy*, *7*(1), 50–57.

En-Nasry, B., & Dafir Ech-Cherif El Kettani, M. (2011). Towards an open framework for mobile digital identity management through strong authentication methods. *Secure and Trust Computing, Data Management, and Applications: STA 2011 Workshops: IWCS 2011 and STAVE 2011, Loutraki, Greece, June 28-30, 2011. Proceedings 8*, 56–63.

Fioravanti, F., & Nardelli, E. (2008). Identity management for e-government services. *Digital Government: E-Government Research, Case Studies, and Implementation*, 331–352.

Gong, S., Tao, X., Li, N., Wang, H., & Han, Z. (2020). Secure secret key and private key generation in source-type model with a trusted helper. *IEEE Access*, *8*, 34611–34628.

Gope, P., & Hwang, T. (2016). A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks. *IEEE Transactions on industrial electronics*, *63*(11), 7124–7132.

Group, T. (n.d.-a). Delivering a mobile eid for the whole nation case study finland. https://www.thalesgroup.com/sites/default/files/gemalto/tel-cs-mobile-id-finland.pdf.

Group, W. B. (n.d.-b). Moldova Mobile Id Case Study. https://documents1.worldbank. Mobile-ID-Case-Study.pdf.

Gupta, H., Mondal, S., Giri, B., Majumdar, R., Ghosh, N. S., & Mishra, V. P. (2019). An authentication model for secure electronic transaction. *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, 283–286.

Hiner, D., Kelly, M., & Huemoeller, R. (2019). Semiconductor package and fabricating method thereof [U.S. Patent Application US 10/497674 B2]. *U.S. Patent*, (US10497674B2).

Husni, E. (2016). Digital signing using national identity as a mobile id. *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 261–264.

Hussain, M., Zaidan, A., Zidan, B., Iqbal, S., Ahmed, M., Albahri, O. S., & Albahri, A. S. (2018). Conceptual framework for the security of mobile health applications on android platform. *Telematics and Informatics*, *35*(5), 1335–1354.

(ITU), I. T. G. (n.d.). Mobile ID: Implementations, challenges and opportunities. https://www.itu.int/en/ITU-D/Regional-Presence/Europe/Documents/Publications/2f_MID_Report_415270-BAT3.pdf.

Ji, D., Zhang, Q., Zhao, S., Shi, Z., & Guan, Y. (2019). Microtee: Designing tee os based on the microkernel architecture. *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th*

IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 26–33.

Jia, X., He, D., Kumar, N., & Choo, K.-K. R. (2019). A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Systems Journal*, *14*(1), 560–571.

Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. (2023). Secure mobile id architecture on android devices based on trust zone. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 1060–1064.

Kamal, K. K., Kapoor, M., & Joshi, P. (2022). Secure and flexible key protected identity framework for mobile devices. *International Journal of Information Security and Privacy (IJISP)*, *16*(1), 1–17.

Kavitha, S., Alphonse, P., & Reddy, Y. V. (2019). An improved authentication and security on efficient generalized group key agreement using hyper elliptic curve based public key cryptography for iot health care system. *Journal of medical systems*, *43*(8), 260.

Kerttula, E. (2015). A novel federated strong mobile signature service—the finnish case. *Journal of Network and Computer Applications*, *56*, 101–114.

Khan, M. A., Quasim, M. T., Alghamdi, N. S., & Khan, M. Y. (2020). A secure framework for authentication and encryption using improved ecc for iot-based medical sensor data. *IEEE Access*, *8*, 52018–52027.

Kim, W. C., Shim, J. H., & Chae, S. H. (2019). Semiconductor package [U.S. Patent Application US 10/741510 B2], (US10741510B2).

Koblitz, N. (2000). Towards a quarter-century of public key cryptography. *A Special Issue of DESIGNS, CODES AND CRYPTOGRAPHY An International Journal*.

Kotwal, V., Parsheera, S., & Kak, A. (2017). Open data & digital identity: Lessons for aadhaar. *2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, 1–8.

Lai, L., & Ho, S.-W. (2012). Simultaneously generating multiple keys and multicommodity flow in networks. *2012 IEEE Information Theory Workshop*, 627–631.

Lambrinoudakis, C., Gritzalis, S., Dridi, F., & Pernul, G. (2003). Security requirements for e-government services: A methodological approach for developing

a common pki-based security policy. *Comput. Commun.*, *26*(16), 1873–1883. https://doi.org/10.1016/S0140-3664(03)00082-3

Lee, U., & Park, C. (2020). Softee: Software-based trusted execution environment for user applications. *IEEE access*, *8*, 121874–121888.

Lee, Y., Lee, J., & Song, J. (2007). Design and implementation of wireless pki technology suitable for mobile phone in mobile-commerce. *Computer communications*, *30*(4), 893–903.

Lee, Y. S., Lim, H., & Lee, H. (2010). A study on efficient otp generation using stream cipher with random digit. *2010 the 12th international conference on advanced communication technology (ICACT)*, *2*, 1670–1675.

Martens, T. (2010). Electronic identity management in estonia between market and state governance. *Identity in the Information Society*, *3*(1), 213–233.

Naik, N., & Jenkins, P. (2016). A secure mobile cloud identity: Criteria for effective identity and access management standards. *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 89–90.

Nishimura, H., Omori, Y., Yamashita, T., & Furukawa, S. (2018). Secure authentication key sharing between mobile devices based on owner identity. *2018 Fourth International Conference on Mobile and Secure Services (MobiSecServ)*, 1–6. https://doi.org/10.1109/MOBISECSERV.2018.8311436

Otterbein, F., Ohlendorf, T., & Margraf, M. (2017). The german eid as an authentication token on android devices. *arXiv preprint arXiv:1701.04013*.

Paar, C., Pelzl, J., Paar, C., & Pelzl, J. (2010). Introduction to public-key cryptography. *Understanding Cryptography: A Textbook for Students and Practitioners*, 149–171.

Prabu, M., & Shanmugalakshmi, R. (2010). A study of elliptic curve cryptography and its application. *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, 425–427.

Ramadan, M., Liao, Y., Li, F., & Zhou, S. (2020). Identity-based signature with server-aided verification scheme for 5g mobile systems. *IEEE Access*, *8*, 51810–51820.

Ray, S., & Biswas, G. (2011). Design of mobile-pki for using mobile phones in various applications. *2011 international conference on recent trends in information systems*, 297–302.

Rehman, S., Talat Bajwa, N., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). Hybrid aes-ecc model for the security of data over cloud storage. *Electronics*, *10*(21), 2673.

Rumbao, F., Manuel, J., & Rodriguez Rubio, F. (2011). Digital signature platform on mobile devices. *MOBILITY 2011: The First International Conference on Mobile Services, Resources, and Users*, 151–157.

Sandberg, L., & Rodberg-Larsen, K. (2004). Method for enabling PKI functions in a smart card [U.S. Patent Application US 7024226 B2], (US7024226B2).

Saquib, Z., Kumar, M., Kamal, K. K., & Varyani, B. (2017). Secure solution: One time mobile originated pki. *2017 Annual IEEE International Systems Conference (SysCon)*, 1–6.

Selvakumaraswamy, S., & Govindaswamy, U. (2016). Efficient transmission of pki certificates using elliptic curve cryptography and its variants. *International Arab Journal of Information Technology (IAJIT)*, *13*(1).

Shepherd, C., Arfaoui, G., Gurulian, I., Lee, R. P., Markantonakis, K., Akram, R. N., Sauveron, D., & Conchon, E. (2016). Secure and trusted execution: Past, present, and future-a critical review in the context of the internet of things and cyber-physical systems. *2016 IEEE Trustcom/BigDataSE/ISPA*, 168–177.

Silasai, O., & Khowfa, W. (2020). The study on using biometric authentication on mobile device. *NU Int. J. Sci*, *17*, 90–110.

Singh, A. K., Solanki, A., Nayyar, A., & Qureshi, B. (2020). Elliptic curve signcryption-based mutual authentication protocol for smart cards. *Applied Sciences*, *10*(22), 8291.

Tavangaran, N., Schaefer, R. F., Poor, H. V., & Boche, H. (2018). Secret-key generation and convexity of the rate region using infinite compound sources. *IEEE Transactions on Information Forensics and Security*, *13*(8), 2075–2086.

Thant, M., & Zaw, T. M. (2018). Authentication protocols and authentication on the base of pki and id-based. *2018 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF)*, 1–8.

Theuermann, K., Tauber, A., & Lenz, T. (2019). Mobile-only solution for server-based qualified electronic signatures. *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 1–7.

Toth, K. C., & Anderson-Priddy, A. (2019). Self-sovereign digital identity: A paradigm shift for identity. *IEEE Security & Privacy*, *17*(3), 17–27.

Tseng, Y.-M., Huang, S.-S., & You, M.-L. (2017). Strongly secure id-based authenticated key agreement protocol for mobile multi-server environments. *International Journal of Communication Systems*, *30*(11), e3251.

Turkcel. (n.d.). Mobil imza — turkcell.com.tr. https://www.turkcell.com.tr/servisler/turkcell-mobil-imza.

Verzeletti, G. M., de Mello, E. R., & Wangham, M. S. (2018). A national mobile identity management strategy for electronic government services. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 668–673.

Wang, D., Cheng, H., He, D., & Wang, P. (2016). On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices. *IEEE Systems Journal*, *12*(1), 916–925.

Wang, H., Sheng, B., & Li, Q. (2006). Elliptic curve cryptography-based access control in sensor networks. *International Journal of Security and Networks*, *1*(3-4), 127–137.

Wu, L., Wang, J., Choo, K.-K. R., & He, D. (2018). Secure key agreement and key protection for mobile device user authentication. *IEEE Transactions on Information Forensics and Security*, *14*(2), 319–330.

Yalew, S. D., Maguire, G. Q., Haridi, S., & Correia, M. (2017). T2droid: A trustzone-based dynamic analyser for android applications. *2017 ieee trustcom/bigdatase/icess*, 240–247.

Zhang, H., Liang, Y., Lai, L., & Shitz, S. S. (2017). Multi-key generation over a cellular model with a helper. *IEEE Transactions on Information Theory*, *63*(6), 3804–3822.

Zhang, P., Liu, Z., Ma, C., Zhang, L., & Han, D. (2019). Kpam: A key protection framework for mobile devices based on two-party computation. *2019 IEEE Symposium on Computers and Communications (ISCC)*, 1–6.

Zhao, S., Zhang, Q., Qin, Y., Feng, W., & Feng, D. (2019). Sectee: A software-based approach to secure enclave architecture using tee. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1723–1740.

List of Publications

- Journals:

  1. Kamal, K. K., Kapoor, M., & Joshi, P. (2022). Secure and flexible key protected identity framework for mobile devices. In 2022 International Journal of Information Security and Privacy (IJISP), (1), 1-17. https://doi.org/10.4018/IJISP.2022010117

  2. Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. (2023). An efficient mCK signing and mobile based identity solution for authentication. In 2023 International Journal of Information Technology, 1-10. https://doi.org/10.1007/s41870-023-01189-8

  3. Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. A Secure and Efficient Mobile ID Framework for Authentication with Enhanced ECC. In International Journal of System of Systems Engineering, https://doi.org/10.1504/IJSSE.2025.10059079

  4. Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. (In press). Verification and Analysis of Solution Based on Mobile PKI for Signing and User Identity. In International Journal of Critical Infrastructures.

- International Conference:

  1. Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. (2023). Secure Mobile ID Architecture on Android Devices based on Trust Zone. In 2023 IEEE 7th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1060-1064). https://doi.org/10.1109/ICCMC56507.2023.1008390

  2. Kamal, K. K., Gupta, S., Joshi, P., & Kapoor, M. (2023). Comparative Analysis of Various Elliptic Curve Cryptography Algorithms for Handheld Devices. In IEEE 7th International Conference on Electronics, Communication and Aerospace Technology 2023. https://doi.org/10.1109/ICECA58529.2023.10395705

  3. Kamal, K. K., Gupta, S., & Joshi, P. (2024). A Framework for Identity Management on Mobile Devices for mGovernance Application. In 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS). IEEE, Raichur, India, pp. 1-8. https://doi.org/10.1109/ICICACS60521.2024.10498337

Kamal

ORIGINALITY REPORT

9%	2%	7%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Kapil Kant Kamal, Monit Kapoor, Padmaja Joshi. "Secure and Flexible Key Protected Identity Framework for Mobile Devices", International Journal of Information Security and Privacy, 2022 Publication	2%
2	Kapil Kant Kamal, Sunil Gupta, Padmaja Joshi, Monit Kapoor. "Secure Mobile ID Architecture on Android Devices based on Trust Zone", 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), 2023 Publication	1%
3	Kapil Kant Kamal, Sunil Gupta, Padmaja Joshi, Monit Kapoor. "An efficient mCK signing and mobile based identity solution for authentication", International Journal of Information Technology, 2023 Publication	1%
4	Zia Saquib, Manish Kumar, Kapil Kant Kamal, Bharat Varyani. "Secure solution: One time	<1%