

# A Smart Card Based Security Extension for the Bitcoin Wallets



By

**Majid Amjad Hussain**

**NUST201464148MSEECS63114F**

Supervisor

**Dr. Shahzad Saleem**

**Department of Computing (DoC)**

A thesis submitted in partial fulfillment of the requirements for the degree  
of Master of Science in Information Security (MSIS)

In

Department of Computing (Doc)

School of Electrical Engineering and Computer Science

National University of Sciences and Technology (NUST)

Islamabad, Pakistan.

(July 9, 2018)

# Approval

It is certified that the contents and form of the thesis entitled “**A Smart Card Based Security Extension for the Bitcoin Wallets**” submitted by **Majid Amjad Hussain** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Shahzad Saleem**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

Committee Member 1: **Dr. Muhammad Muddassir Malik**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

Committee Member 2: **Ms. Hira Anwar**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

Committee Member 3: **Ms. Ayesha Kanwal**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

# Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis written by Mr. Majid Amjad Hussain, (Registration No NUST201464148MSEEC63114F), of SEEC (School/College/Institute) has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Name of Supervisor: **Dr. Shahzad Saleem**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

Signature (HOD): \_\_\_\_\_

Date: \_\_\_\_\_

Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_

# Dedication

This work is dedicated to my parents, teachers and friends.

# Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Majid Amjad Hussain

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

# Acknowledgment

I am thankful to Allah Almighty for granting me patience to do my Master's degree. I would like to thank my parents and family who always motivated me. Special thanks to my supervisor Dr. Shahzad Saleem and external supervisor Dr. Awais Shibli and Prof Dr. Sead Muftic who always helped and guided me during my thesis phase.

I would like to thank my committee members, who always guided me. I would like to express my gratitude to my friends (Amna Riaz, Kanwal Qayyum, Sara Khurshid and Zohaib Shahid), colleague and especially KTH-AIS Lab members.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Problem Statement . . . . .	2
1.3	Aim and Scope . . . . .	2
1.3.1	Objective 1: . . . . .	3
1.3.2	Objective 2: . . . . .	3
1.4	Research Methodology . . . . .	3
1.4.1	Design Science and Our Research . . . . .	3
1.5	Audience . . . . .	9
1.6	Limitations . . . . .	9
1.7	Research Contribution . . . . .	9
1.8	Organization of Thesis . . . . .	10
<b>2</b>	<b>Literature Review</b>	<b>12</b>
2.1	Bitcoin Wallet . . . . .	12
2.2	Access Management . . . . .	12
2.3	Management and Protection of User Security Profiles . . . . .	13
2.4	Types of Bitcoin Wallet . . . . .	14
2.4.1	Software Based Wallets . . . . .	14
2.4.2	Local Wallet . . . . .	15
2.4.3	Online/ Hosted Wallet . . . . .	15
2.4.4	Paper Wallet . . . . .	15
2.4.5	Hardware Wallet . . . . .	15
2.5	Bitcoin Wallet Security . . . . .	16
2.5.1	Password Protected Wallet . . . . .	16
2.5.2	Litecoin Wallet Application . . . . .	16
2.5.3	Malware Attacks . . . . .	16
2.6	Existing Schemes in Literature . . . . .	17
2.6.1	Bitcoin Clients Comparison on the Basis of Usability and Security . . . . .	17
2.6.2	A First Look at the Usability of Bitcoin Key Management	18

2.6.3	Investigation of Cryptocurrency Wallets on iOS and Android Mobile Devices for Potential Forensic Artefacts	18
2.6.4	Realizing Two-Factor Authentication for the Bitcoin Protocol . . . . .	18
2.6.5	Securing Bitcoin Wallets via Threshold Signatures . . .	19
2.6.6	Multi-Signatures . . . . .	20
2.7	Analysis . . . . .	21
<b>3</b>	<b>Motivation with Case Studies and Findings</b>	<b>23</b>
3.1	Motivation . . . . .	23
3.2	Case Study . . . . .	24
3.2.1	Coinbase Wallet Authentication Bypass . . . . .	24
3.2.2	Bitcoin Wallet Security Bypass . . . . .	25
3.3	Findings . . . . .	26
<b>4</b>	<b>Proposed Solution</b>	<b>28</b>
4.1	Introduction . . . . .	28
4.2	Proposed Architecture Design . . . . .	28
4.2.1	Architecture Design Components . . . . .	29
4.3	Identified Issues and Proposed Solutions . . . . .	30
4.3.1	Poor/ Weak or No Authentication . . . . .	31
4.3.2	Encryption of Bitcoin Wallet with Weak Key . . . . .	33
4.3.3	Encryption of Bitcoin Wallet's backup with Weak Key	34
4.3.4	Malware Attacks . . . . .	34
4.3.5	Social Engineering Attacks . . . . .	34
<b>5</b>	<b>Implementation</b>	<b>36</b>
5.1	Introduction . . . . .	36
5.2	Detailed Architecture . . . . .	36
5.3	Tools and Technologies Used . . . . .	37
5.4	Entities and their Roles . . . . .	38
5.5	Data Model . . . . .	39
5.5.1	Functions . . . . .	39
5.5.2	Data Objects . . . . .	40
5.6	Implementation . . . . .	40
5.7	Application Setup Usage and Evaluation . . . . .	41
5.7.1	Initial Setup and Activation . . . . .	41
5.7.2	Setup for Transactions . . . . .	41
5.7.3	Getting Smart Card . . . . .	42
5.7.4	Bitcoin Wallet Legitimate User Authentication . . . . .	42
5.7.5	Bitcoin Wallet Illegitimate User Authentication . . . . .	43



<b>6</b>	<b>Evaluation of the Proposed Solution</b>	<b>45</b>
6.1	Scope . . . . .	45
6.2	Economy . . . . .	45
6.3	Usability . . . . .	45
6.4	Performance . . . . .	45
6.5	Objectives . . . . .	46
6.6	Security . . . . .	46
<b>7</b>	<b>Conclusion and Future Work</b>	<b>47</b>
7.1	Conclusion . . . . .	47
7.2	Future Work . . . . .	48
<b>A</b>	<b>Smartcard (Javacard) Applet Code for Bitcoin Wallet</b>	<b>49</b>
<b>B</b>	<b>APDU Command/ Response of Applet</b>	<b>54</b>
B.1	Select Wallet Applet . . . . .	54
B.2	Verify Wallet Applet PIN . . . . .	54
B.3	Change Wallet Applet PIN . . . . .	55
B.4	Secret Key Generation . . . . .	55
B.5	Getting Secret Key from Wallet Applet . . . . .	56
<b>C</b>	<b>Screenshots of Application</b>	<b>57</b>

# List of Figures

1.1	Design Cycle [1] . . . . .	6
1.2	Reasoning in the Design Cycle [2] . . . . .	7
2.1	Litecoin .93 File Showing Private Key . . . . .	16
2.2	Multi-Signatures . . . . .	21
3.1	Coinbase Bitcoin Wallet Authentication Bypass . . . . .	25
3.2	Bitcoin Wallet . . . . .	26
4.1	Smart Card based Bitcoin Application's Architecture . . . . .	29
4.2	Smart Card based Authentication . . . . .	33
4.3	Random Keyboard . . . . .	35
5.1	Detailed Architecture . . . . .	37
5.2	Customized Secure Bitcoin Application . . . . .	41
5.3	Getting Smart Card . . . . .	42
5.4	Legitimate User Authentication . . . . .	43
5.5	Illegitimate User Authentication . . . . .	44
C.1	Screenshots of Application . . . . .	57
C.2	Screenshots of Application . . . . .	58

# List of Tables

3.1	Existing Issues/ Challenge . . . . .	27
4.1	Issues/ Challenges and their Solutions . . . . .	31
5.1	Tool and Technologies Used . . . . .	38
5.2	Entities and their Roles . . . . .	39

# Abstract

Virtual currencies are being increasingly used around the globe. For this reason, the security of the Bitcoin wallets is also becoming a major concern for Bitcoin community. The security experts and developers are constantly at work to come up with concrete solutions for ensuring the security of Bitcoin wallets, so that their vulnerability exploitation could be reduced to zero. This is so, because every single day, a large number of attacks targeted towards Bitcoin, are being launched leading to monetary losses to a large number of Bitcoin users. In this regard, this research work gives a security analysis of existing Android Bitcoin wallets. It includes bypassing techniques used by malicious entities for the implemented security practices. Such techniques have also caused financial losses to Bitcoin users. As a countermeasure, we have proposed a smart card based authentication scheme that can protect the users against all of the identified attacks.

**Key Words:** *Bitcoin Wallet, Smart Card, Authentication, Android, Security, Java, Virtual Currency, Cryptography, Shared Preference, SQLite, Security Architecture, Software Development, Mobile Development.*

# Chapter 1

## Introduction

Bitcoin is the world's first decentralized peer-to-peer crypto currency, presented by Satoshi Nakamoto in 2009 [3]. Because of its decentralized structure, it is not controlled by any government, bank or any other entity. The whole system is based on peer-to-peer communication and on an open source protocol. The security of this protocol is based on cryptographic algorithm (Elliptic Curve and SHA256) which is already proven to be very secure and used world widely for public-key cryptography and integrity check. People are adopting Bitcoin rapidly because it enables one to make instant payment with very low transaction fee across the world. The transfer of money, using Bitcoin, is cheaper than other available solutions.

Big companies, such as WordPress, PayPal and Ebay gave it considerable attention and also accepted Bitcoins as a medium for their payment [4]. Bitcoin is being used and accepted by many other big companies. Whose number is increasing every day. At the time this thesis is being written, 16475363 Bitcoins are in transmission [5] and each Bitcoin is worth \$ 2713.45 USD [6]. Crypto-currencies like Bitcoin have redefined the meaning of financial transaction by providing a virtual platform for creation, circulation and transformation through anonymous and irrevocable transactions. In 2016, more than 100,000 merchants were accepting Bitcoin [7] and adopting it for transfer of money due to its convenience and lesser transaction fees. At the time of writing this thesis, the market capitalization of Bitcoin is \$5.7 billion [8]. However, with this increase in market acceptance, the Bitcoin has to face a lot of security challenges.

Nowadays, the mobile phone is becoming more and more popular due to its communication and other features. Different research studies show that mobile phones are replacing personal computers. As mobile phones are be-

coming smart, most of the people have switched to smart phones. Smart phones are just like personal computers which people can carry with them anywhere. Due to the portable nature of smart phone, people are most likely to use different applications on smart phone, including payment and banking applications for online payment and transactions. Hence, most of the time, people are using smart phones for Bitcoin transaction through some mobile-based Bitcoin application. In 2016, Android market share was 88% [9], as majority of the people are using Android. There are many security vulnerabilities and violations of Android security [10–13]. Therefore, we chose smart card. In this research we focused on security of Android based Bitcoin wallet applications.

## 1.1 Motivation

Bitcoin wallets are very important entities in Bitcoin architecture. Just like physical wallets, they are used for storing cash and are responsible for keeping a record of all transactions. The security of Bitcoin wallet is also critical and associated with the security of the complete Bitcoin architecture. This is so because the Bitcoin wallet creates the Bitcoin addresses along with the storing private keys that are associated with these Bitcoin addresses. Compromise of private key results in loss of all Bitcoins associated with a specific Bitcoin address. For this purpose, a lot of emphasis is placed on securing the Bitcoin wallets for securing the private keys within these wallets.

## 1.2 Problem Statement

Once an adversary gains physical access to the device running Android based Bitcoin wallets, they can bypass the authentication mechanism and can steal Bitcoins by carrying out transactions on users' behalf. This creates a need for an efficient and secure solution which can ensure the security of the wallet, even when the device is not in the possession of actual user.

## 1.3 Aim and Scope

This research addressed the security concerns of Android based Bitcoin wallets. The first aspect of this research was to study existing security schemes available for Bitcoin wallet thoroughly and identify their issues. An analysis of their security and usability problems was done. For this, security audit

and penetrating testing of existing famous open source and proprietary application was also performed. The second aspect of this research was to propose a valid security solution for Android based Bitcoin wallets. This research did not only focus on how an application stores the user's credentials and private key securely but also how it evaluates the security of its authentication mechanism, especially in the case when the cellphone is being operated by some person, other than the true owner. The scope of this research does not include covering all security and usability aspects in this research rather it is limited to the following objectives.

### **1.3.1 Objective 1:**

The top five existing security schemes (for Android based Bitcoin wallets), different types of wallets and their security/ usability issues were studied. The penetration testing of two existing wallet applications, that is, one open source (Bitcoin Wallet [14]) and one proprietary (CoinBase Bitcoin Wallet [15]) was carried out. The penetration testing was done from the perspective of authentication and secure storage, using OWASP standards [16].

### **1.3.2 Objective 2:**

A secure Android application (Bitcoin Wallet) was developed, in order to address the security issues like authentication and secure storage of user's private key, was developed. An open source application was modified for sake of implementation of different security features and extensions, as a proof of concept.

## **1.4 Research Methodology**

This research work falls under problem solving domain, as this research is going to address the security concerns of Android based Bitcoin wallets. As this research that is being carried out, aims at solving a particular problem and providing a corresponding solution, therefore it can be thoroughly studied better through the "design science research approach [17]."

### **1.4.1 Design Science and Our Research**

Design science is a research methodology in the field of information technology, which suggests detailed procedures and guidance for assessment and

repetition within the research projects. Design science research not only focuses on developing designed artefact and their performance but also there is a clear intention of improving the functional performance of the artefact is involved. According to Van Aken, design science research aims at developing knowledge that can be used by professionals to design solutions for the problems they face in their respective fields.

We built solution concepts, which will address the security issues like authentication and secure storage of user's private key, in Bitcoin Android wallets. In short, aim is to improve the security of Android Bitcoin wallets, which is an "improvement solution" for security savvy users. Therefore, it lies and fits well in the domain of design science research, as argued by Aken [18].

### Activities and Output

The main elements of this research were Development and later its evaluation. The outputs included models, techniques and best practices. For development of these models and methods, we investigated in this area in order to gather information and knowledge of constructs and concepts being used in the domain of Android and information security. According to the concept of design science by Mark and Smith [19], there are two design processes: developing and evaluating; and four types of outputs: constructs, models, methods and instantiations, where:-

1. **Constructs:** Constructs are about forming a conceptual vocabulary of a particular domain. Constructs are utilized to explain the problem as well as to propose its corresponding solution. The theoretical bases, existing methodologies and techniques used for authentication and to protect private key were studied and evaluated to build our concepts.
2. **Models:** Models are set of statements expressing relationship between constructs. In much simpler words, it describes and represents how things actually are and how they should be like. After formulation of concepts, we utilized them to form models to: Improve the authentication process of Android based Bitcoin wallets. Improve the security of Bitcoin wallet's private key.
3. **Methods:** Methods are the guidelines/ steps to perform a particular task to solve the problem. The proposed solution was developed, by selecting the smart card and following the best practices, which was the key to achieve both strong authentication and safe storage of Bitcoin wallet private key.



4. **Instantiations:** According to March and Smith [19], the final output is Instantiation, which “operationalizes constructs, models and methods.” It is actually the completion of realization of the artefact in an environment. The developed solutions are in the form of models, methods and guide for realizing best practices. These solutions fulfil the requirements of strong authentication and safe storage of Bitcoin wallet private key. We have thoroughly described the crux of the research methodology that we plan to use and the ways that our research is related to it. Our research goals can be understood with the help of design science approach, as we want to create reality by:-

- (a) Exploring the domain.
- (b) Elaborating the problem.
- (c) Proposing its solutions.

### **Research Process**

Now, at this stage, we will map our research process steps to the steps of design science research. According to Takeda et al. [1], the design research cycle comprised of five sub-steps:-

1. Awareness of problem
2. Suggestion
3. Development
4. Evaluation
5. Conclusion

The whole steps of research activity based on design science research approach, is presented in figures 1.1,1.2.

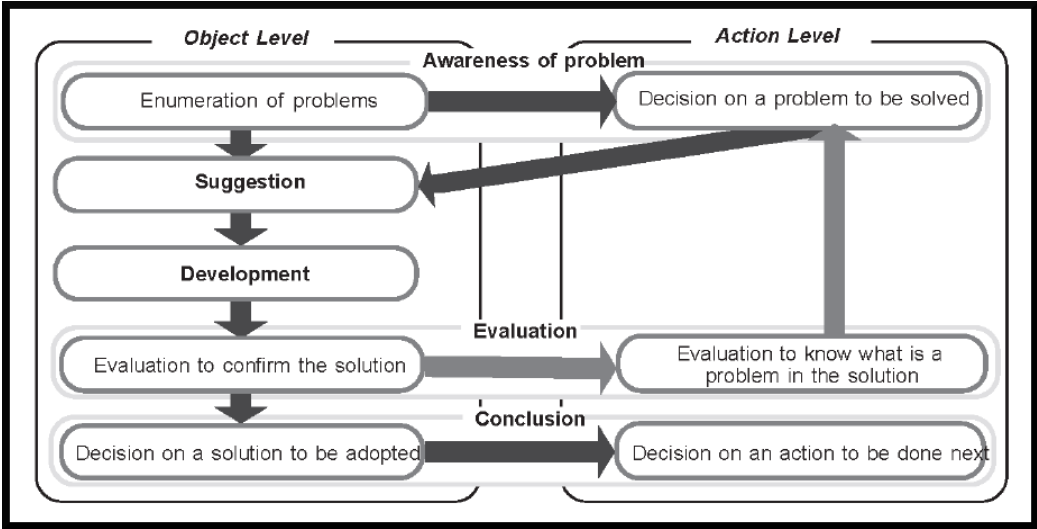


Figure 1.1: Design Cycle [1]

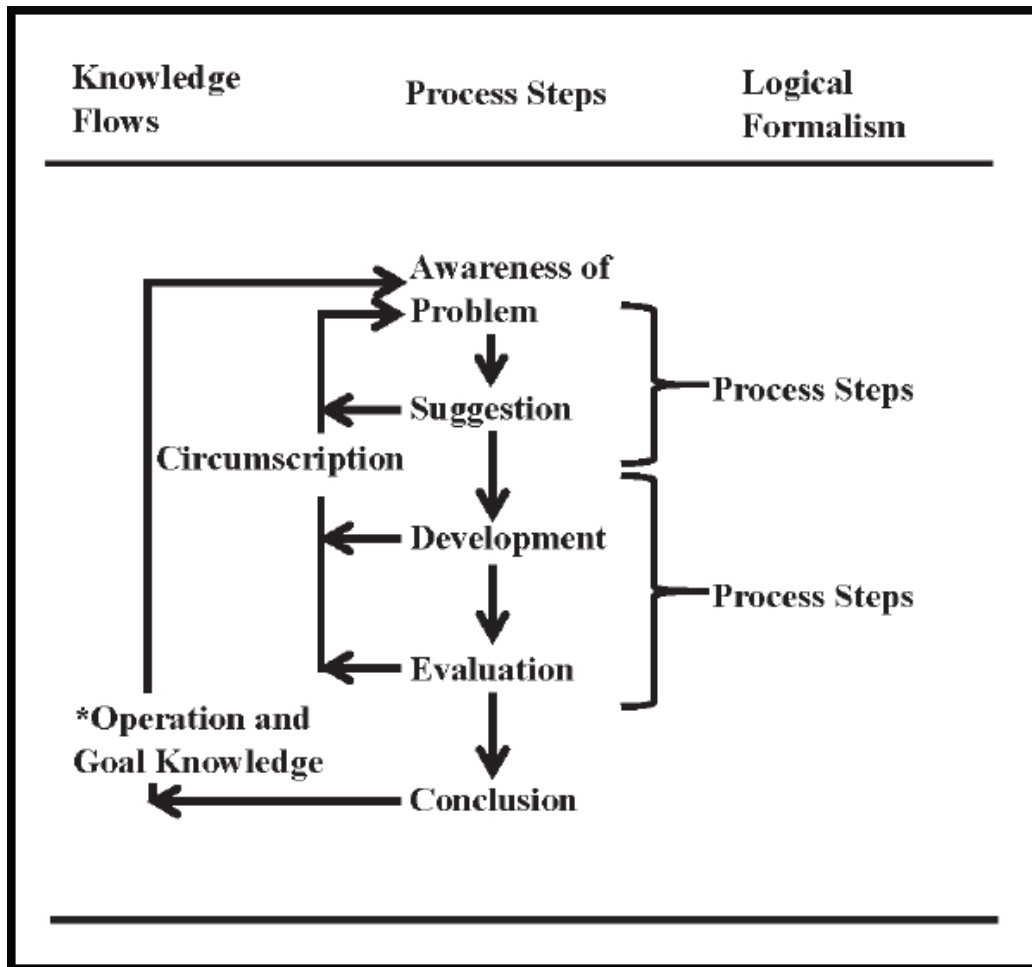


Figure 1.2: Reasoning in the Design Cycle [2]

1. **Awareness of the Problem:** Getting familiar with the problem and its related domain is acquired during this phase. The output of this phase is a Proposal, formal or informal, for a new research effort. Awareness of the problem is required in order to suggest a solution. During this phase problems are identified by exploring existing security schemes available for Bitcoin wallet and identifying their issues. Awareness was the first phase in our research. During this phase we performed a detail analysis of existing security schemes with respect to their security and usability. For this, security audit and penetrating testing of existing famous open source and proprietary application was also done.
2. **Suggestion:** After gathering enough knowledge about the domain and

gathering enough awareness of the problem in the initial phase. During this phase, we proposed a solution to the problem being encountered. The key concepts necessary for solving the problem were recommended [Table No. 4.1]. The domain understanding as enhanced in phase one, in which the concerned knowledge was gathered. During this phase, we suggested a solution for strong authentication and safe storage of Bitcoin wallet private key.

3. **Development:** In this step, an artefact is further developed and implemented under the guidance of the suggestions. Following the suggestions, a prototype is developed to secure Android Bitcoin wallet application. An open source application is modified to implement different security features and extensions, as a proof of concept.
4. **Evaluation:** After developing the artefact, it is evaluated for different criteria of performance and improvement. Any deviations from expectations, both quantitative and qualitative are carefully noted. A new cycle is vital, if a problem is catered during the phase of evaluation. Development, evaluation and suggestion phases are frequently performed over and over again in a classic design research approach [2]. The solution that was proposed was evaluated. It was established that, a secure chip (smart card) is introduced in existing system by following the suggested solution. On the positive side, the solution provide strong authentication, safe storage of Bitcoin wallet private key, prevention from malwares and phishing attacks.
5. **Conclusion:** In this phase the knowledge attained as a product of our research activity was shared. Specific solutions were recommended in the form of models, methods and best practices. During the research different Bitcoin wallets were examined from security prospective, finally it is concluded that Bitcoin wallets and similar application requires more security than other general application, as these application are dealing with money. In this research, we developed a secure Bitcoin wallet application for this purpose. This application provides strong authentication and secure storage of user's credentials inside smart microSD card. This solution is developed using modularity approach. Everybody can take these security extensions and integrate with their own application.

## 1.5 Audience

As the competition in the mobile application development in online market has increased considerably over the past few years, there are many applications specifically focusing on financial sector services provisioning lack comprehensive security in their systems that have been deployed.

This research focused on providing secure solution to financial sector for building applications with added facility of making Bitcoin based transactions. This solution will help to cater the threats with considerable ease without affecting performance. The proposed design of the solution as per this research will act as an important contribution in this domain and will provide guidance for building secure solutions for mobile application involving online payments.

The approach selected for designing the proposed solution was scalable as well as inter operable among various platforms. It was modular and capable of being easily integrated with any application for providing necessary security features. The security mechanism that was proposed is transparent to users.

## 1.6 Limitations

In order to define the scope of our thesis, we limited our work to Android based Bitcoin wallet application authentication and the security of key storage. Other security aspects like secure communication were not covered in this research. The proposed solution was only implemented and tested in simulators, not on the real device, that is, the smart card. This is because, in Pakistan, programmable smart cards and smart card readers are currently not available.

## 1.7 Research Contribution

In this research, the following contributions were made:

1. A paper titled “A Smart Card Based Security Extension for the Bitcoin Wallets”, written by Majid Amjad Hussain, Sadia Khalil and Shahzad Saleem, was presented at the First International Conference on Cyber Security and Digital Forensics, arranged at National University of Sciences and Technology in 2017. Later, this article got published in

NUST Journal of Engineering Sciences (NJES). In this paper, we discussed the already proposed security mechanisms for Android Bitcoin wallet and their security/ usability flaws. We also proposed a solution for the security of Android based Bitcoin wallet applications. The solution focused on the authentication and secure storage of sensitive information.

2. We developed a prototype by modifying existing open source Bitcoin application to fix the current security flaws. This included implementation of different security mechanisms within this application, including secure authentication using smart card and secure storage of private key inside the smart card.

## 1.8 Organization of Thesis

The thesis document is organized into seven different chapters. Each chapter explains a particular aspect of research.

**Chapter 1**, titled **Introduction**, includes introduction of Bitcoin, its evolution and the role of the Bitcoin wallet in Bitcoin ecosystem. It explains the problem statement and the motivation behind the research. This chapter describes, briefly, the aim and scope of the thesis with its objectives and limitations of the research. It also gives details about the research contribution and the methodology used to carry out this research.

**Chapter 2**, titled **Literature Review**, provides knowledge about different types of wallets, their functioning and their security and usability flaws. This chapter also provides a detailed survey of the existing schemes which are used to protect Android based Bitcoin wallets. The highlights of the security and usability problems of these wallets are also given.

**Chapter 3**, titled **Motivation with Case Studies and Findings**, explains security flaws of the existing open source and proprietary Android based Bitcoin wallet applications, found by the author, while conducting this research. These flaws created a need to provide a solution to overcome these security challenges.

**Chapter 4**, titled **Proposed Solution**, describes, in detail, the solution proposed by the author, in order to make a secure application for using Bitcoin on Android platform. This chapter also explains the architecture of the

proposed solution.

**Chapter 5**, titled **Implementation**, provides the technical details about implementation of prototype to secure Bitcoin wallets. In addition to this, it also explains the implementation flow and code snippets.

**Chapter 6**, with the title of **Evaluation of the Proposed Solution**, provides knowledge about evaluation results of implemented solution.

**Chapter 7**, with the title of **Conclusion and Future Work**, details conclusion and highlights the future research directions.

# Chapter 2

## Literature Review

In this chapter, first of all, we are going to understand the Bitcoin wallet and its significance in Bitcoin architecture. We will go through different types of Bitcoin wallet, based upon their functioning, and look into their security aspect, critically. As the last part of this chapter, the existing schemes, which are found in literature, are discussed with security pros and cons.

### 2.1 Bitcoin Wallet

Bitcoin wallets are digital wallets, just like the real wallets. A Bitcoin Wallet is basically software which makes it easier for users to make worldwide transactions and payments for free. With the Bitcoin wallet, you can send or receive payments and do other transactions, based on the Bitcoin. The Bitcoin wallet is responsible for creating and managing user profile, Bitcoin address, public key and private key. Bitcoin wallets carry necessary information including user credentials and a public and private key pair. Like real-life, it is important to ensure security of Bitcoin Wallet and also that it is not used by any person who is not authorized to its use. There are many aspects which should be taken into consideration, as follows:

### 2.2 Access Management

The protection of user's Bitcoin Wallet needs to be ensured and should be made accessible only to those who are authorized. Most commonly, authorized access is ensured using popular mechanisms such as passwords. Passwords, although not the best solution for providing high level security, are still good enough to act as easiest and cheapest way to ensure security. The major issue leading to unauthorized access is weak security passwords set



by the users. One can only assume that users are security savvy enough to choose complex passwords that are difficult to guess by hackers or unauthorized users.

## 2.3 Management and Protection of User Security Profiles

This is related to the user profile stored on the device storage system. In general, Android system offers five types of storage options [20]:-

1. **Shared Preferences:** Store private data in key-value pairs.
2. **Internal Storages:** Store private data on the device memory.
3. **External Storage:** Store public data on the shared external storage.
4. **SQLite Databases:** Store structured data in a private database.
5. **Network Storage:** Network storage can be used to store and retrieve data on the web-based services of the application.

The “Shared Preferences” option is used to store and fetch persistent key-value pairs of primitive data types. SQLite, on the other hand is a lightweight Android database, having full support of Android OS. It is generally used for storing objects including user contacts and user specific content. It uses external storage for this purpose which is not that secure.

The internal storage should be preferred to store the SQLite database. The internal storage is only accessible to particular user applications that are installed in the phone. This storage is not even accessible by user who owns that phone, therefore provides basic security to application based data. Even that being the case, the Android operating system’s vulnerabilities have been violated way too many times. Therefore, it is advised that the system should not be trusted and used cautiously and carefully.

The significant part of a user’s profile is the password, PIN and private key that are sensitive information of the user and needs to be stored, fetched and used safely and securely. Passwords are made secure using special characters that make them difficult to crack and guess. The four digit PIN generally used for additional features, such as blocking of account on more than three

attempts etc. In general, possibilities of online and offline attacks are minimized in this way.

There are different types of Bitcoin wallet, based on their functionality. Some of them are software-based and some of them are hardware-based. The following section describes in detail the different types of Bitcoin wallets along with their security and usability issues.

## 2.4 Types of Bitcoin Wallet

The different types of Bitcoin wallets are as follows:-

### 2.4.1 Software Based Wallets

Software-based wallets are those which can be installed on PCs or on smart phones. Mainly, there are two types of software-based Bitcoin wallets:-

1. Heavyweight (full application) Wallets
2. Lightweight (GUI/ communications only) Wallets

The two main characteristics of these types of Wallets are:-

#### Full Wallet

Full wallet stores all security credentials and block-chain transactions locally and has a direct connection with the Bitcoin network.

#### Lightweight Wallet

Lightweight wallet provides only a GUI-based communication with users and a connection to the Wallet server. The security credentials and block-chain transactions are all stored in the Wallet server. The Wallet Server also performs communication with the Bitcoin networks. Other types of wallets are given below:-

1. Local Wallet
2. Online/ Hosted Wallet
3. Paper Wallet
4. Hardware Wallet

### **2.4.2 Local Wallet**

The Local wallet can be installed on the mobile or desktop computer. In this type of wallet, all the keys and passphrases are stored locally on the device. In the case of being hacked or loss of the phone or laptop, the Bitcoins can be stolen easily.

### **2.4.3 Online/ Hosted Wallet**

The Hosted wallets are the wallet services provided by third party vendor via a web application. In these types of wallets, the user does not need to manage or secure the private key as key management is handled by the service provider. In this case, the user is dependent on the security of the service provider.

### **2.4.4 Paper Wallet**

Paper wallets are considered as the safest way to store Bitcoins. In a paper wallet, the Bitcoin address and private keys are stored on a piece of paper instead of storing on some digital device or some device which is connected to internet. As everything is on paper and not connected to the internet, there is no chance to lose the Bitcoin through some hacking or a malware attack. It can only be possible if someone gets that piece of paper. It is the safest way to make a transaction but at the same time, this wallet creates the problem of usability and availability. You need that particular piece of paper every time whenever you want to make a transaction. It is also difficult to write down 1024 or 2048-bit long key on a paper and later use it in the wallet to make a transaction.

### **2.4.5 Hardware Wallet**

The Hardware wallet stores the user profile, Bitcoin address and private key on some hardware, like a USB device. Physical access to the hardware, which stores information and user credentials, is needed at the time of transaction. This provides security but, at the same time, creates usability and availability issues because the offline media has to be available all the time in order to carry out the transactions.

## 2.5 Bitcoin Wallet Security

The Bitcoin wallet is an integral part of the overall Bitcoin architecture. The security of the Bitcoin is very necessary. If the wallet gets compromised by any means, the attacker can steal all the Bitcoin without the knowledge of the owner. Following are some techniques which are used to protect the wallet.

### 2.5.1 Password Protected Wallet

Password protected (encrypted) wallets provide much more security than any other wallet. A key is derived from the password, chosen by the user, and the whole wallet is encrypted with that key. The security of the wallet is dependent on the key, which is in turn, dependent on the password chosen by the user. Normally, users don't use best password practices and choose passwords which have a very low entropy. This motivates the attackers to perform brute force or dictionary attacks.

### 2.5.2 Litecoin Wallet Application

Litecoin is a virtual currency just like Bitcoin. It is found that, .93 file of this wallet contains the private key as well as the date and time stamp, as depicted in figure 2.1 . Any entity who has access to this file can steal the money easily. This file is accessible to everyone after rooting the device. With the help of the public key found in log file, one can search the address of the public ledger to check all previous transactions, in order, to connect to the address.

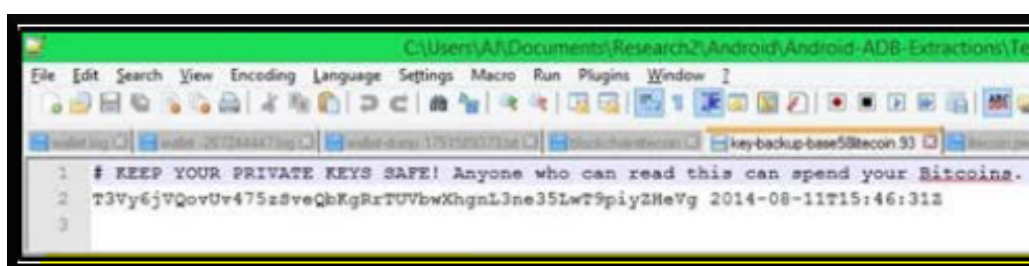


Figure 2.1: Litecoin .93 File Showing Private Key

### 2.5.3 Malware Attacks

Nowadays, attackers write malwares and spread it to the botnet. The malware steals the private key and wallet data from the local storage of a user's

device and sends it to the attacker, which leads to the loss of Bitcoins. In the case of device theft, an attacker can simply carry out a transaction, as there is no security mechanism implemented. Another threat to the wallets is spyware or keyloggers, these softwares can record keys strokes while user is entering PIN or password in the Bitcoin application. Later on these malicious software can send all this information to the attacker. The attacker can carry out different types of attack with this information resulting in loss of Bitcoin, for true owner.

## 2.6 Existing Schemes in Literature

Following are the schemes presented by different authors in order to protect the Bitcoin wallet so the Bitcoins cannot be stolen.

### 2.6.1 Bitcoin Clients Comparison on the Basis of Usability and Security

Rostislav Skudnov et al. [21] describe different types of Bitcoin clients, with respect to their usability and security. Full clients are those who download the whole block-chain and require excessive storage and more network bandwidth. Mobile based clients need to download block headers and hence, require less storage space and network bandwidth. Bitcoinj, a Java library, is used for this purpose. Thin clients are the browser-based clients, where no computation is done on the client side and most of the tasks are being done on the server side. In this type, the user doesn't need to secure the private key. As most of the things are managed on the server side, therefore, if the server is compromised, then all the system is compromised. Mining clients provide high CPU or GPU-based computation in order to run the Bitcoin system. Some other clients include signing-only clients, deterministic wallets and brain wallets. The Bitcoin release 0.4.0 provides the facility to encrypt the wallet with a passphrase to prevent the wallet from an attacker. The release 0.4.0 enables the Bitcoin wallets to read the encrypted wallet and decrypt it, on run time, if correct passphrase is provided. Paper backup is a technique where the user takes the backup of his/ her private key on paper instead of digital media. For this purpose, the QR-code is being used in order to avoid the typing mistake. The printed paper, with the correct QR-code having private key, can be accepted as the payment. The receiver can send a transaction by scanning the QR-code and signing it with the scanned private key.

### **2.6.2 A First Look at the Usability of Bitcoin Key Management**

Shayan Eskandari et al. [22] proposed a scheme in which the keys can be stored in the local storage of the phone, accessible by the application at any time, in order to carry out the transactions. Despite this benefit of offline or local storage, any other application which has access to the physical storage can steal the private keys. Such a scenario is prone to malware attack whereby the malware can extract and send private keys to any malicious entity.

### **2.6.3 Investigation of Cryptocurrency Wallets on iOS and Android Mobile Devices for Potential Forensic Artefacts**

Angelica Montanez et al. [23] examines different Bitcoin wallets with respect to the aspect of forensics. The paper discusses different techniques which can be used to get useful information about any installed Bitcoin wallet. This information can be the date of installation, updating or deletion. Log files can be manipulated to find transaction information, IP addresses of peers as well as a number of Bitcoins in the wallets. During the examination of Litecoin wallet application, it was found out that .93 part of the file contained the private key as well as date and time stamp. Any entity which has access to this file can steal the money easily. This file is accessible to everyone after rooting the device. With the help of the public key found in the log file, one can search the address of the public ledger to check all previous transactions to connect to the address.

### **2.6.4 Realizing Two-Factor Authentication for the Bitcoin Protocol**

Christopher Mann et al. [24] presented a threshold scheme, as a solution, to device theft if unencrypted wallets are installed on that device. In order for a transaction to take place, more than one signature is required. To prevent a single point of failure, 'n' out of 'm' signatures are needed to proceed with a transaction. This resolves the problem of theft in such a way that, in the case of theft of one device, the attacker cannot spend Bitcoins. However, this solution causes a problem of the size of increased transactions which ultimately leads to increase in the transaction fees. Another solution provided is to split the key and store it in two devices, that is, the mobile

phone and the desktop. The user uses a desktop application to initiate the transaction. Then the application generates a QR-code, which is scanned by the mobile application for the purpose of authenticity. A secure TLS connection is built and part of the key is exchanged to proceed with the transaction. This solution uses less internet bandwidth and file system, as well as less transaction fee because the size of the transaction is small. The drawback of this solution is that both of the devices, which have the parts of the key, should be available at the time of the transaction.

### Features

1. Split the key and store it in two devices (mobile, desktop).
2. Avoid single point failure.
3. Resolved theft issue.
4. No additional bandwidth and space required.

### Limitations

1. Usability issue (no ease of use).
2. Availability.
3. No security mechanism.

## 2.6.5 Securing Bitcoin Wallets via Threshold Signatures

Steven Goldfeder et al. [25] provided a mechanism to secure the Bitcoin wallet via threshold digital signatures. This technique is based on a threshold value, ‘t’, which is actually predefined in the system. First, we have to provide a private signing key to each entity in the system. If we want to carry out a Bitcoin transaction, we need the digital signatures of at least ‘t’ entities in order to carry out the transaction. The mechanism implements the principle of “separation of privileges” where multiple privileges are required to perform a task. This solution is feasible for office environments where the consent of more than one person is required for a transaction to be made. Multiple signatures also create a problem of increased size of the transaction and transaction fees.

**Features**

1. Requires 'n' out 'm' signatures for transactions.
2. Resolve theft issue.
3. Suitable for office environment.
4. Separation of privileges.

**Limitations**

1. Increase in transaction size.
2. Increase in transaction fee.
3. Not suitable for a single individual.
4. Additional bandwidth and space required.

**2.6.6 Multi-Signatures**

In this scheme, 3 keys are required to secure the wallet. A trusted third party holds one key while the users control the remaining two keys. Two keys are required to unlock the wallet. As two keys are needed to access the Bitcoins, the wallet is virtually hack-proof, as depicted in figure 2.2.

**Limitations**

1. Spoofing
2. Masquerading
3. Phishing attack



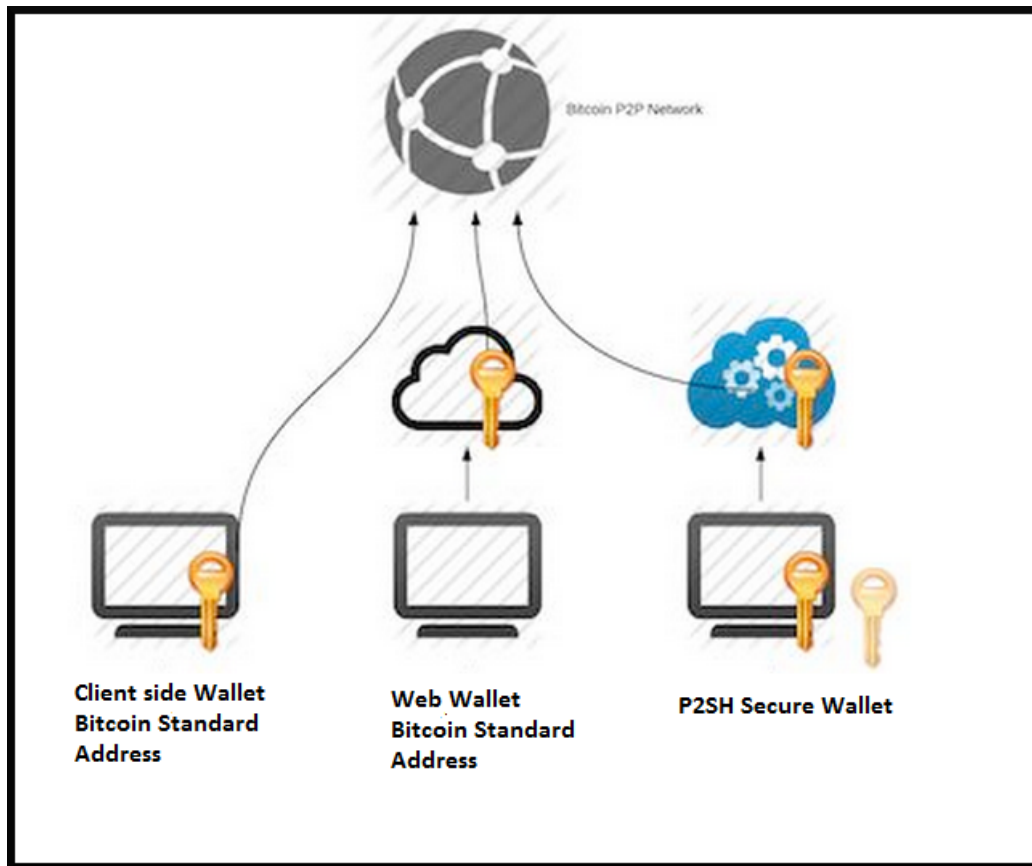


Figure 2.2: Multi-Signatures

## 2.7 Analysis

Bitcoin Wallets contains public and private crypto keys, other passcodes, and PIN. This makes them prone to a large number of theft attacks. If an adversary gets hold of the phone, they can easily bypass authentication and then, can carry out transactions on user's behalf.

After analysing the existing schemes for Bitcoin wallet security, it was seen that currently in every scheme, the way of storing credentials and the implementation of the security mechanism is prone to various attacks. For the sake of proof, in the following chapter, we analysed two renowned applications, an open source (Bitcoin wallet) and a proprietary (CoinBase wallet) Android-based Bitcoin wallet application and performed security penetration testing.

We observed that the authentication process, used by the Bitcoin Wallets, to make sure that only the authorized person is using the application, can be bypassed. The existing authentication scheme stores passcodes and PIN in internal storage (in “Shared Preferences” as XML file or as SQL database file) and believes that the stored information cannot be accessed by another application, not even by the owner of the phone. However, the files can easily be accessed by anyone who can get root access to the device.

For this purpose, we proposed a solution; a security scheme for Android based wallets based on the smart card, which keeps the keys and other necessary data out of the Android device.

# Chapter 3

## Motivation with Case Studies and Findings

In order to prove our hypothesis that existing applications are not implementing enough security, we considered two applications, an open source (Bitcoin wallet) and a proprietary (Coinbase wallet) Android-based Bitcoin wallet application for security penetration testing.

### 3.1 Motivation

The existing authentication scheme used by Bitcoin wallets can be analysed by using the application logic written in the code. This can be exploited by simply decompiling the Android .APK file. If Bitcoin Wallet store hash of passcode and PIN using MD5 or SHA1 algorithm, then this can be easily cracked within seconds with the help of web-based cloud services, which already have the computed hash chains [26].

Sometimes, the application developer relies on a cryptographic solution (encryption, decryption). With this approach, the developer accepts PIN code from the user and generates a key from this PIN code and then, encrypts all data with it and stores nothing in the internal memory of the phone except the encrypted file. At the time of decryption, the application requests and obtains the PIN, generates the key, and decrypts the file. If the file is decrypted into a correct format, this means that the provided PIN is correct; otherwise, the PIN is considered incorrect.

Normally, the length of the PIN is 4 digits, which means that an adversary only needs  $10^4$  tries to log into the system successfully. To overcome

this issue, the applications put a limit of 2-5 unsuccessful tries for the PIN. An attacker can still bypass this mechanism by simply rooting the device, extracting the encrypted file and cracking the encryption on their system, by using brute force, without the limit on the number of tries. They can achieve results, within a few seconds, on a simple laptop.

## 3.2 Case Study

To test the security level of the existing application, we penetrated Bitcoin wallet application and Coinbase application from the authentication prospective. Following are two case studies, showing how easily an attacker can bypass the authentication mechanism if true owner loses the device or some other person takes the device from the true owner temporarily.

### 3.2.1 Coinbase Wallet Authentication Bypass

Coinbase is an Android-based Wallet which contains a design flaw which leads to a severe vulnerability whereby the adversary can bypass PIN authentication. This allows the adversary to open an Android Wallet without knowing the PIN. With this design flaw, the adversary can steal and manipulate user information, for example, username and email. Apart from that, this allows the adversary to steal all the Bitcoins present in the wallet. Android applications can store data in either, a SQLite3 database or an XML file called “Shared Preferences.” Coinbase Android Wallet uses XML file to store information regarding the PIN.

By making certain changes in the Shared\_Preferences.XML file, we were able to bypass PIN authentication. If the attacker deletes the XML statement used for storing PIN related information, he can mislead the application into launching with the home page instead of the PIN authentication page. The home page offers the ability to steal Bitcoins from the wallet. The steps used for exploitation are explained below and are shown in Figure No. 3.1. Set PIN for authentication.

1. User logs into the system.
  - (a) User data is stored in com.coinbase.android\_preference.xml.
2. Login is successful and the home screen is prompted.
3. The attacker edits com.coinbase.android\_preferences.XML accordingly and replaces the actual file with the edited one.

4. Attacker opens the app and sees the home page instead of the authentication screen.

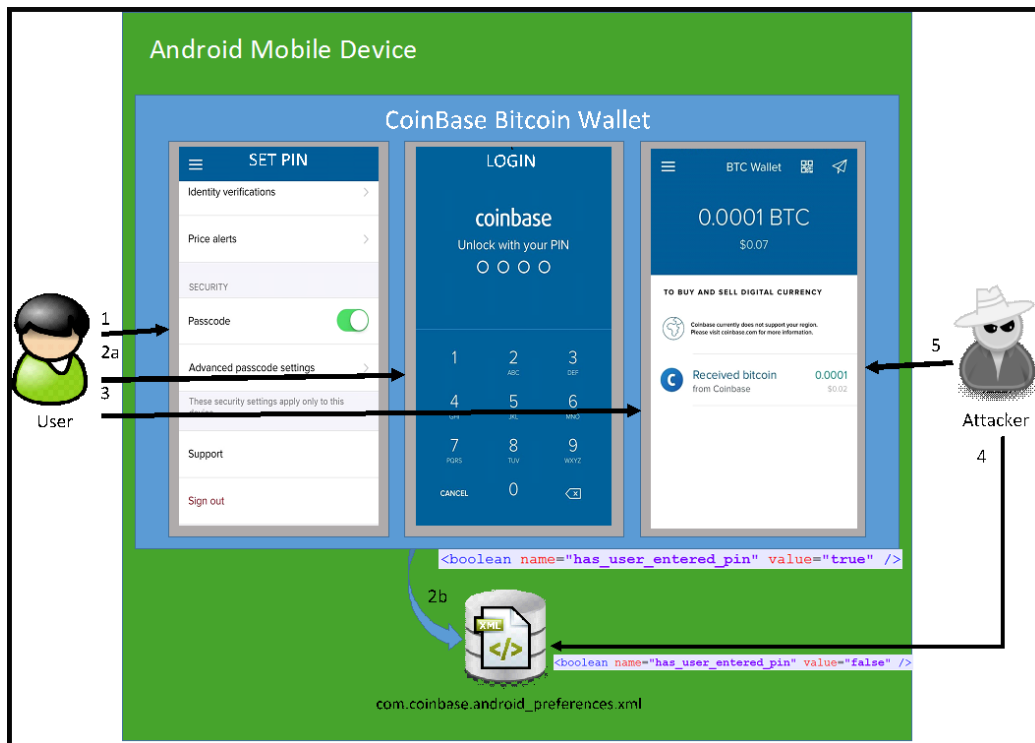


Figure 3.1: Coinbase Bitcoin Wallet Authentication Bypass

### 3.2.2 Bitcoin Wallet Security Bypass

“Bitcoin Wallet” is an Android-based open-source Wallet application, which can be downloaded from the official site of Bitcoin as well as from Android Play Store. On Android Play Store, it has 500,000 –1,000,000 downloads which mean this Wallet is quite popular among the users of Bitcoins. For the purpose of safety, this wallet uses a PIN code to protect unauthorized spending of Bitcoins. The wallet only accepts numeric values which means that we can only enter digits ranging from zero to nine. Figure No. 3.2 shows that the application marks the PIN “strong” even when the user uses a guessable PIN (8 digits). This application also implements the policy of limiting the tries which means that a user cannot enter a PIN code after a certain number of tries. With this PIN, the application generates a key and encrypts the whole wallet and saves it to internal storage. An adversary can

root the device, pull the encrypted file and apply brute force by using  $10^8$  combinations.

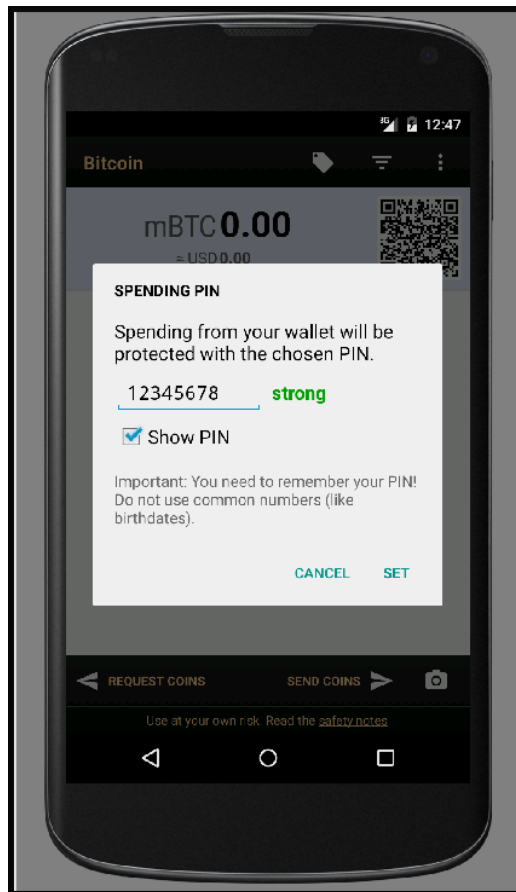


Figure 3.2: Bitcoin Wallet

The above discussion and case studies clarify that the existing mechanisms are not good enough to secure Bitcoin wallets. In the coming chapter, we proposed a suitable solution for this problem.

### 3.3 Findings

Table No. 3.1 shows the issues and challenges which exist in the existing Bitcoin wallet application as per the case studies discussed above.

Table 3.1: Existing Issues/ Challenge

<b>Sr. No.</b>	<b>Issues/ Challenges</b>
1	Poor/ weak or no Authentication while accessing the Bitcoin Wallet
2	Encryption of Bitcoin Wallet with weak key, which can be derived from Password/ PIN
3	Encryption of Bitcoin Wallet's backup with weak key, which can be derived from Password/ PIN
4	Prone to Malware Attacks
5	Prone to Social Engineering Attacks

# Chapter 4

## Proposed Solution

This chapter explains the proposed solution.

### 4.1 Introduction

We noted that the Android device is not safe in perspective of storing users' credentials and important keys. We proposed a solution which keeps the keys and other necessary data out of the Android device. For this, we proposed a security scheme for Android based wallets based on the smart card.

### 4.2 Proposed Architecture Design

Figure No. 4.1 shows the architecture of proposed scheme, for secure authentication for Android-based Bitcoin Wallets.



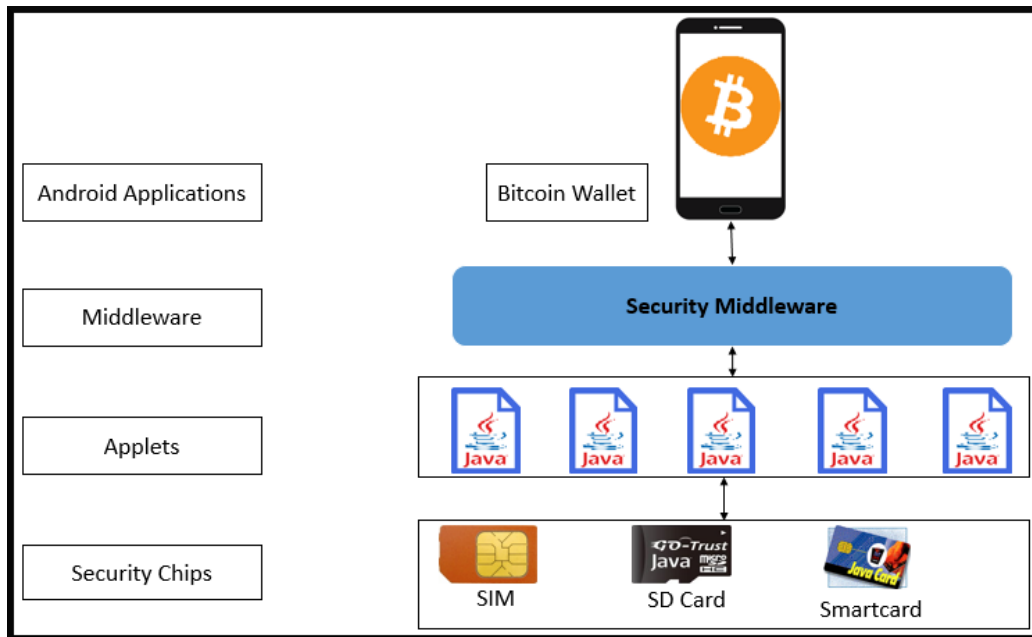


Figure 4.1: Smart Card based Bitcoin Application's Architecture

### 4.2.1 Architecture Design Components

Following are the core design components of the proposed solution.

1. Android application
2. Middleware
3. Applet
4. Security chip

#### Android Application

In this architecture, an Android Bitcoin wallet application is a specially design application which enables a user to authenticate via PIN verification and store private key on the smart card. It enables the user to perform Bitcoin transactions securely. User interacts with this application to make transactions while this application interacts with the smart card by using middleware to provide security features.

### **Middleware**

Middleware is the API which enables a Bitcoin wallet application to interact with the smart card. It provides different functions, depending upon the type of the smart card.

### **Applet**

Applet is a piece of software which resides in the smart card. In this architecture, the applet provides security service like PIN verification, storage of private key, encryption and decryption.

### **Security Chip**

A plastic chip with an embedded microprocessor and memory is used to achieve security. Most of the time, this chip is used in financial transactions. There are different types of security chips in the market. Such sophisticated smart cards do exist, which can be placed in the memory card slot of the mobile phone. You can use them as a memory card and as a smart card at the same time. Go-Trust Technologies launched microSD secure memory card, which combines advanced features of smart card technology and USB mass storage into a convenient microSD form. This is a very smart solution to secure portable devices. Go-Trust launched the SDK for accessing smart card chip through a file-system based interface for different mobile platforms. Its extended benefits include application security in JavaCard-based solution for PKI support, that is, digital certificates, cryptographic keys storage, strong authentication, digital signature and data encryption for a perfect PKI integration and standard mass storage facility (i.e. 2 GB to 8 GB) [27].

## **4.3 Identified Issues and Proposed Solutions**

Table No. 4.1 shows the issues and challenges identified in existing Bitcoin wallet application and their proposed solution which will be explained in detail later. For the problems identified in Table No. 2, we have proposed a smart card-based solution derived from FIPS 196. This standard specifies challenge-response protocols. Using this protocol, different entities authenticate themselves to each other in a computer system. A paper titled “Smart Card Authentication for Mobile Devices” [28] and its report [29] were published by NIST.

Table 4.1: Issues/ Challenges and their Solutions

Sr. No.	Issues/ Challenges	Solution
1	Poor/ weak or no Authentication while accessing the Bitcoin Wallet	Smart Card Based Authentication
2	Encryption of Bitcoin Wallet with weak key, which can be derived from Password/ PIN	Encrypt wallet with smart card's key
3	Encryption of Bitcoin Wallet's backup with weak key, which can be derived from Password/ PIN	Encrypt backup with Smart card's key
4	Prone to Malware Attacks	Custom keyboard
5	Prone to Social Engineering Attacks	Random keyboard every time

### 4.3.1 Poor/ Weak or No Authentication

Existing Bitcoin applications either use no authentication or use weak or poor authentication. By poor or weak authentication, we mean, a simple PIN code, which can be found easily either in "preferences.XML" file or in the database file. This PIN can also be brute forced easily. To overcome this problem, we proposed a smart card-based authentication. Following are the steps:-

1. The first user downloads the Android Bitcoin application from Android Play Store. This application has a CA certificate.
2. The user requests the service provider to provide a smart card. The service provider provides the smart card, which has a private key (KR) as well as a user certificate.
3. The application retrieves the certificate from the smart card and stores it. After the retrieval of the certificate from the smart card, it is verified first. As the application has a CA certificate, it matches the CA signature on the certificate provided by the smart card.
4. The user enters the default PIN code to authenticate with smart card and sets a new PIN code.
5. The registration process ends here. Now it's the turn of the authentication process.

6. At first, the user enters the PIN code. If the PIN code is correct, the smart card will allow the user to proceed. Otherwise, it will terminate the authentication process and lock the smart card after three unsuccessful attempts.
7. If the PIN code is correct, the application generates a random number “A” and sends it to the smart card.
8. The smart card generates a random value “B”, signs “A—B” (‘—’ denotes concatenation) with the private key that is stored on the card and returns “B” and the respective signature to the application.
9. The application retrieves the user certificate, from which it extracts the public key “KU”, then verifies it and then, verifies the card’s signature over; “A—B” using the public key contained in the certificate.
10. If everything is verified successfully, the authentication is successful. Otherwise, the attempt of authentication fails.

Figure No. 4.2 explains the flow of different activities.

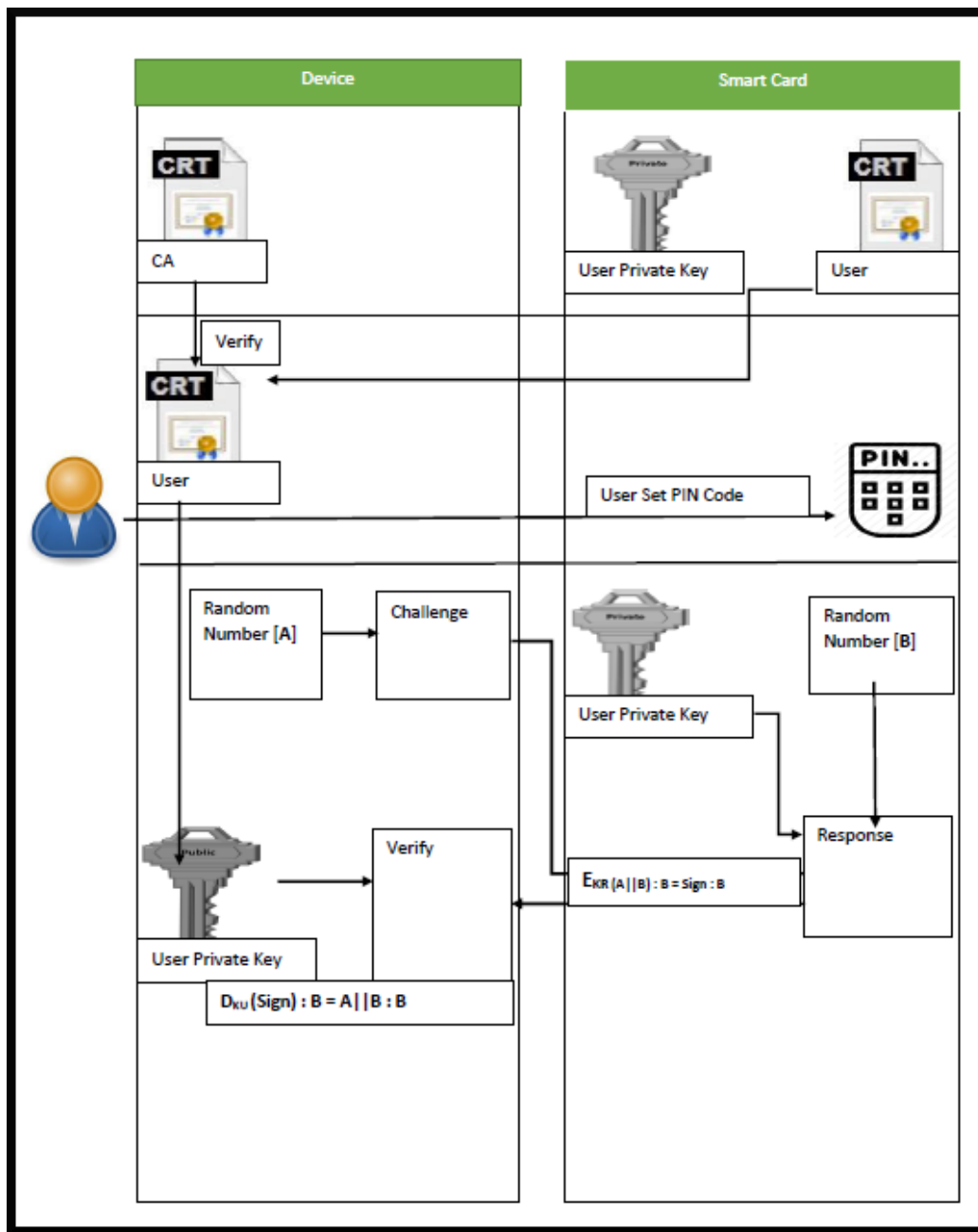


Figure 4.2: Smart Card based Authentication

### 4.3.2 Encryption of Bitcoin Wallet with Weak Key

Bitcoin Wallet provides protection to the wallet by encrypting it. Bitcoin Wallet encrypts the wallet with the PIN provided by the user. It is very easy

to brute force only numeric values.

In our proposed solution, first the smart card takes the hash of the user's PIN and encrypts it with the private key which is stored on the smart card. The newly generated key (New Key = ENCKR (HASH (PIN))) is sent back to the application for wallet encryption.

Another solution is that by using a smart card with good processing power and memory, we can send the wallet file to the smart card and the smart card would encrypt it with its own private key, which is, ENCKR (Wallet File).

### 4.3.3 Encryption of Bitcoin Wallet's backup with Weak Key

Bitcoin Wallet provides the mechanism for backup in the case, the phone is lost or some malware corrupts the data in the wallet or its private payment key. Bitcoin Wallet, at the time of backup, asks for a password to encrypt the Wallet so that only the true owner can access the data. The wallet is useless for others as it is encrypted but it is observed that people don't often use strong passwords. As users use low entropy passwords, the attacker takes advantage of this fact and figures out the passwords easily in most of the cases. To provide enhanced security, we proposed the encryption of the Wallet, at the time of backup, with the private key of the smart card.

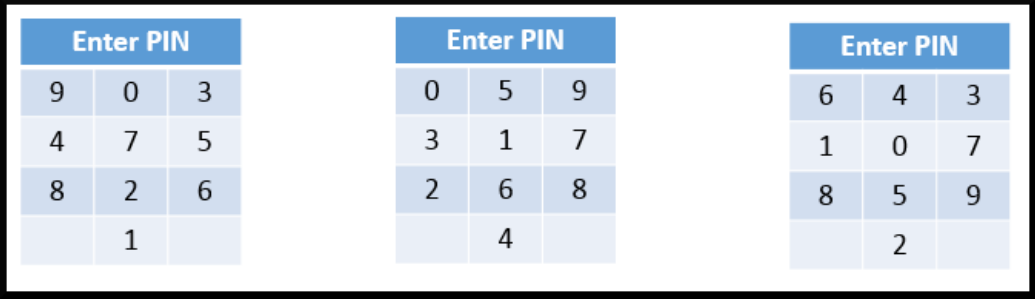
### 4.3.4 Malware Attacks

Android has approximately 88% market share of smartphones. Attackers are writing more malware for Android as compared to any other mobile operating systems. Nowadays, a number of malwares and rootkits are infecting the Android devices. Most of the malware installs a keylogger on the device and tries to capture credentials. Keylogger gets the root-level privilege and listens to system calls made by the application and captures keystrokes. The solution to this problem is to write your own custom keyboard.

### 4.3.5 Social Engineering Attacks

Humans are known to be the weakest link in security. For example, imagine a person sitting in front of you and observing you while you enter the PIN in the Bitcoin Wallet application. By carefully observing the movement of

fingers, one can detect the correct PIN. To avoid this, we proposed a custom keyboard which shows a random placement of keys every time.



The figure displays three different random keyboard layouts for PIN entry, each enclosed in a blue header box labeled "Enter PIN". The layouts are arranged horizontally within a larger black-bordered frame.

Enter PIN		
9	0	3
4	7	5
8	2	6
	1	

Enter PIN		
0	5	9
3	1	7
2	6	8
	4	

Enter PIN		
6	4	3
1	0	7
8	5	9
	2	

Figure 4.3: Random Keyboard

# Chapter 5

## Implementation

### 5.1 Introduction

In order to make the Android Bitcoin wallet application secure, we took an open source Android Bitcoin application and modified its code to implement our security extensions, which are mentioned in the previous chapter. Apart from the modification of the application, a piece of code for the smart card, technically called an applet, has also been written. To test the whole scenario we used the smart card and Android emulator.

### 5.2 Detailed Architecture

Figure No. 5.1 shows the detailed architecture and the flow of proposed and implemented solution. The user interacts with the Bitcoin application. The Bitcoin application is implemented on the model view presenter design pattern whereby the user interacts with the view. The view is responsible to pass the user's request to the model and then, the model communicates with the drivers and middleware APIs to pass information to the smart card. Before using the smart card, the applet is installed on the smart card, which is responsible for receiving different requests from the application, process them accordingly and pass back the results to the application.



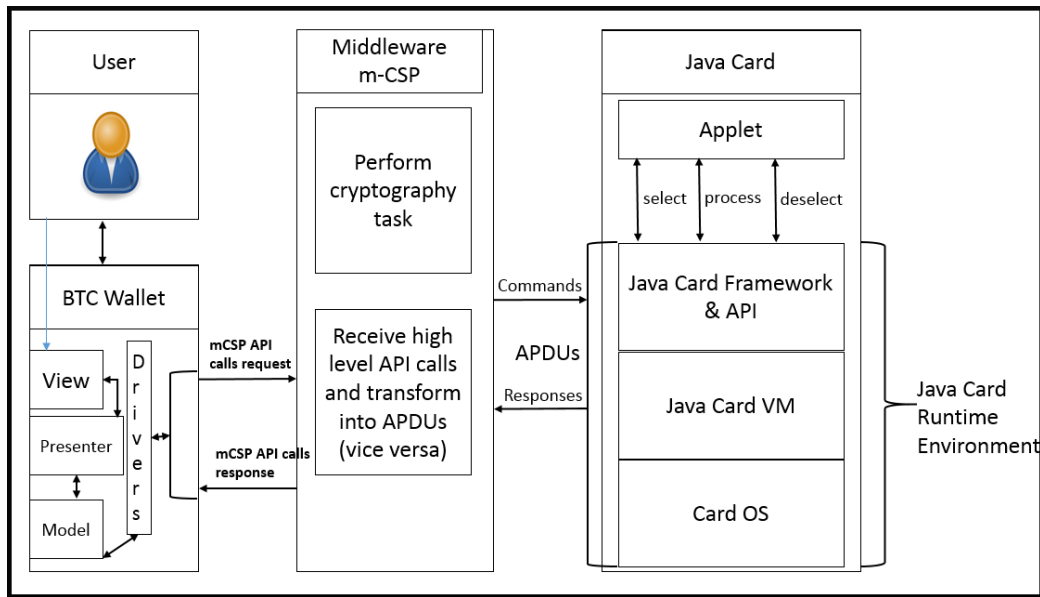


Figure 5.1: Detailed Architecture

### 5.3 Tools and Technologies Used

While extending the existing open source application and writing code for the smart card applet, we used different tools and technologies which are mentioned in Table No. 5.1 along with their explanation. To test the application and smart card applet, we used the Android and smart card simulator.






Table 5.1: Tool and Technologies Used

<b>Sr. No.</b>	<b>Name</b>	<b>Explanation</b>
1	Java Card OpenPlatform	Smart card Operating System for the Java Card Platform, developed by IBM
2	Java Card (SKD)	Smart (Java) Card Development Kit 2.2.1
3	Eclipse	IDE
4	Android Studio	IDE
5	Java	Programming Language (1.7 +)
6	Android (SDK)	Mobile OS (2.0 8.0)
7	Bitcoin Wallet	Android Bitcoin Wallet
8	JCIDE	Java card Development and Simulation Environment
9	Android Emulator	To emulate the Android applications
10	Algorithm	RSA

## 5.4 Entities and their Roles

The different entities involved and their roles are given in Table No. 5.2. These entities are the part of our new secure Bitcoin application system. We will see how these entities interact with each other and their function in the system.

Table 5.2: Entities and their Roles

Sr. No.	Entity	Image	Role
1	Smart Card		The Smart card is used for authentication, which holds the certificates which will be used for the task of cryptography.
2	Legit User		A user who wants to use the Android Bitcoin wallet for Bitcoin transactions.
3	Android Bitcoin Application		Android Bitcoin application used for the Bitcoin transactions.
4	Service Provider		The service provider, who provides and manages the smart card and develops the Bitcoin wallet.
5	Attacker		A malicious entity who wants to steal Bitcoins by accessing the legit user's Android Bitcoin application and smart card.

## 5.5 Data Model

Following is the data model of our developed system. We implemented the following methods and data objects to implement the proposed security measures, in order to make the existing Bitcoin application more secure.

### 5.5.1 Functions

1. Select Applet
2. Set PIN
3. Verify PIN
4. Update PIN
5. Block PIN/ User
6. Generate RSA Key pair
7. Set/ Get Private Key

8. Set/ Get Public Key
9. Generate X509 User Cert
10. Generate X509 User Cert signing request
11. Store X509 User Cert
12. Get X509 User Cert
13. Encrypt Data
14. Decrypt Data
15. External Authentication

### **5.5.2 Data Objects**

1. User Certificate
2. Public Key
3. Private Key
4. User PIN

## **5.6 Implementation**

The applet code is attached in appendix A, while the APDUs and their format is specified in appendix B.

## 5.7 Application Setup Usage and Evaluation

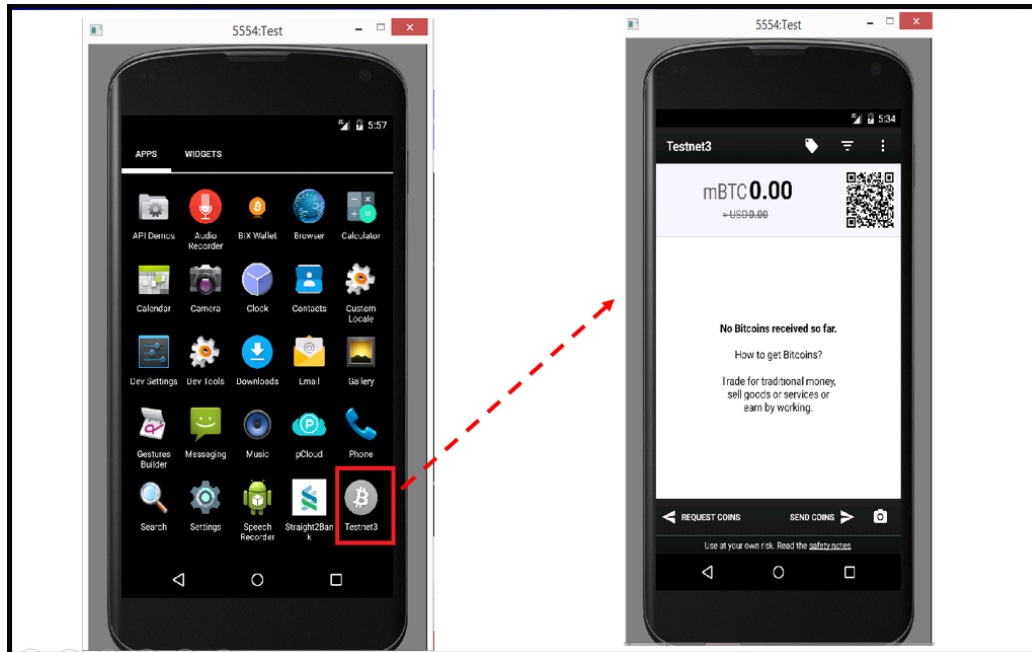


Figure 5.2: Customized Secure Bitcoin Application

### 5.7.1 Initial Setup and Activation

1. Application installation.
2. When the user activates the application for the first time, it generates a public and private key pair and a Wallet object.
3. The application stores the private key and wallet object in the internal space. You can find these as files if you have root level privilege on the phone. These files can be found in the folder named “/data/-data/de.schildbach.wallet/files.”

### 5.7.2 Setup for Transactions

1. For the purpose of receiving and making payments, Bitcoin Wallet connects to the Bitcoin network and downloads a small part of the blockchain.

2. After that, the application accesses the wallet object and public and private keys and makes payment transactions by directly connecting to the Bitcoin network.

### 5.7.3 Getting Smart Card

1. The user requests a smart card.
2. The service provider provides smart card.
3. Download the Bitcoin wallet from Play Store.
4. User possesses smart card and Bitcoin application and is ready to make transactions.

The Figure No. 5.3 shows the diagram of this part.

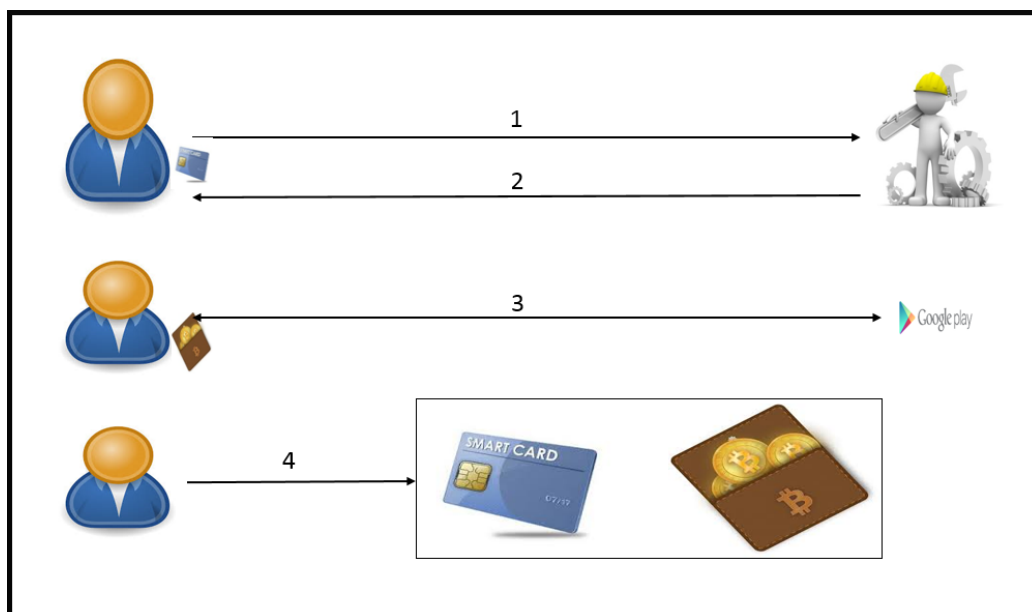


Figure 5.3: Getting Smart Card

### 5.7.4 Bitcoin Wallet Legitimate User Authentication

1. The user enters PIN/ password to authenticate locally to the Wallet.
2. The phone receives PIN/ password and sends it to the smart card.
3. The smart card processes and sends back results (authenticated or not).

4. The Bitcoin Wallet shows the message to user about whether he or she has been given the access or not.

In Figure No. 5.4, the steps of authentication have been shown.

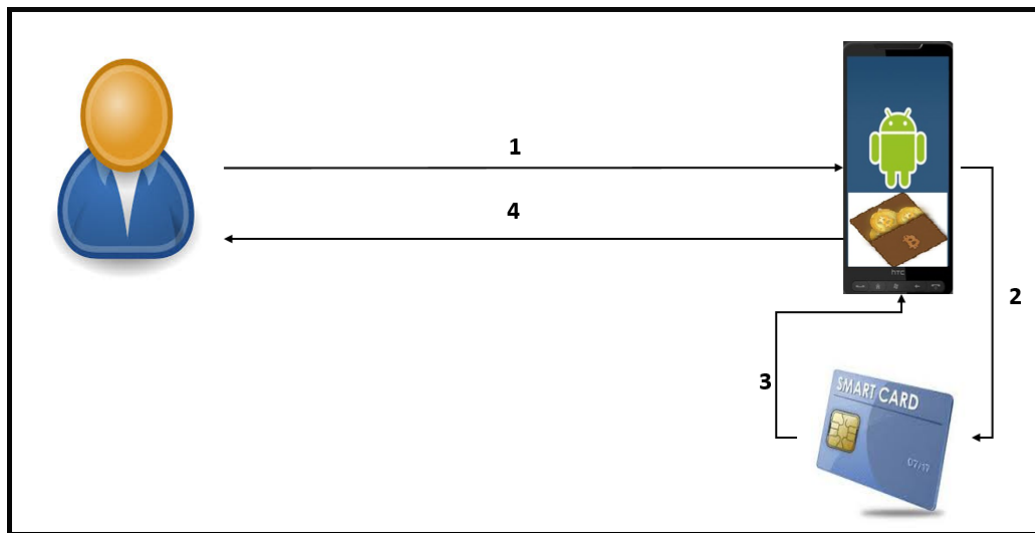


Figure 5.4: Legitimate User Authentication

### 5.7.5 Bitcoin Wallet Illegitimate User Authentication

In Figure No. 5.5, the procedure, in the case an illegitimate user tries to access the Bitcoin Wallet, has been shown.

1. Attacker enters wrong or random PIN/ password to authenticate.
2. In 2a, the attacker use an actual smart card and tries to access the Bitcoin Wallet. In 2b, the attacker use a fake smart card and tries to access Wallet.
3. In the result of 2a application and smart card will block after the specified number of attempts. In 2b application will not work with a fake smart card.
4. Rejection message is sent to the attacker.

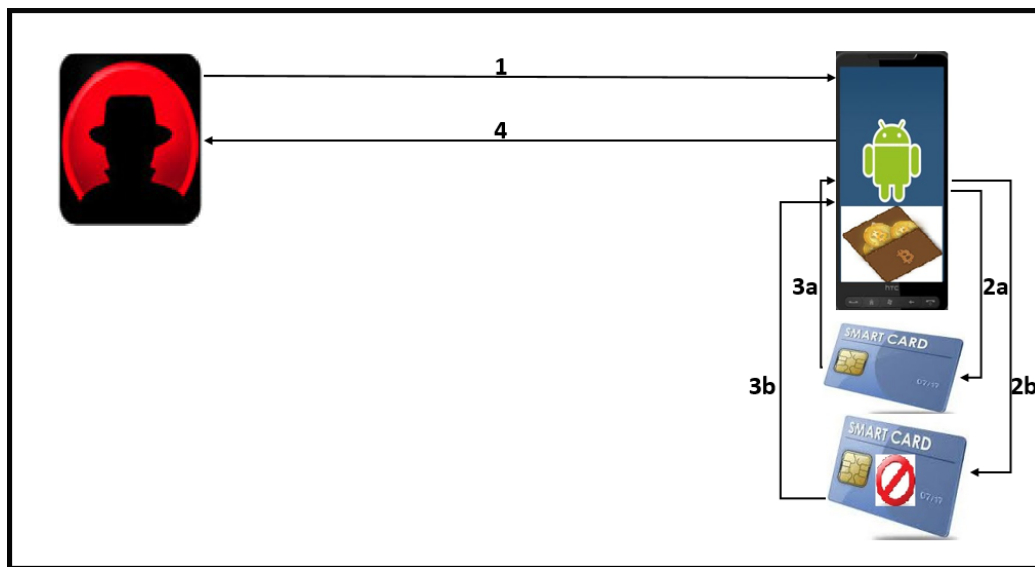


Figure 5.5: Illegitimate User Authentication



## **Chapter 6**

# **Evaluation of the Proposed Solution**

In this chapter, the proposed solution “Bitcoin Wallet”, is evaluated using different standards. Security of the Wallet was the main aim of this research.

### **6.1 Scope**

The scope of the Bitcoin Wallet includes all the users of Android mobile phones who prefer to use Bitcoins as a way of making computerized payments, in a secure and reliable way, with high usability.

### **6.2 Economy**

This project is Java based and is open source. It is freely available on internet and can be utilized and enhanced by other Android developers.

### **6.3 Usability**

This application is user-friendly. There is no special knowledge required to operate this application.

### **6.4 Performance**

The performance of the wallet application depends on the device hardware and the speed of internet connection. The application is well responsive.

## 6.5 Objectives

The application achieves PIN authentication, encryption and decryption of all sensitive information of the user. It guarantees the secrecy and security of the user's data.

## 6.6 Security

One of the major aims of this research was enhancement of security in Bitcoin mobile applications for facilitating the users. For that it is imperative that the user authentication mechanism cannot be forged and it is ensured that the digital currency must not be hacked by malicious entity. For this purpose, smart card is used to provide authentication and other security features. The security of smart card is already proven as it is widely used by banking industry as ATM card. However, 100% security is never possible and hence cannot be promised by any technology available so far.

# Chapter 7

## Conclusion and Future Work

### 7.1 Conclusion

In this thesis, a precise overview of Android mobile operating system security mechanism, Bitcoin ecosystem, Bitcoin wallets, their types and threats to Bitcoin wallet are provided. During the research, different Bitcoin wallets were examined from security prospective, finally it was concluded that Bitcoin wallets and similar application requires more security than other general application, as these applications are dealing with monetary assets of the users.

Existing Bitcoin wallet applications store their critical data normally in mobile phones internal memory which are always under threat. In this research, we developed a secure Bitcoin wallet application for this purpose. This application provides strong authentication and secure storage of user's credentials inside smart microSD card. Nowadays smartphones are becoming more powerful and have the capability of processing security operations, but the Bitcoin wallet application's data and user's credentials are not secure in internal memory, as PIN, password, and other important information saved in internal memory of smartphones which are on the hit list of adversaries all the time. Therefore, without using tamper-proof smart card, we can't ensure that all important information is safe against threats. In our solution we achieved the required security features by using smart microSD based smart card. We took an open source Bitcoin wallet application and add security extensions to it, to make to secure. The developed application was tested with the help of Android and smartcard emulator. User will use his/her Android phone where microSD memory slot is generally available. Nowadays almost every person owns Android devices, so that there is no need to get

any hardware except smart microSD memory card, which is a cheap device.

In case the mobile is lost or stolen, the saved information is still protected from an adversary, as the protected information cannot be retrieved from the PIN protected smart card.

Our developed solution provides strong PIN authentication mechanism and protection of important information by using secure microSD card. We have assessed our solution by using comprehensive threat model, and verified that it's been useful for Android Bitcoin wallet applications. The most important thing about developed solution is that, this solution is developed using modularity approach. Everybody can take these security extensions and integrate them with their own application. The provided solution is totally transparent to user and no interaction with the network operator is required.

## 7.2 Future Work

This research work addresses most of the security issues faced by the Android Bitcoin application. Due to time limitation and resource limitation our work was limited. During this research, problem of authentication, social engineering and malware attacks were countered, while on the other hand full wallet encryption and full back up encryption is not yet done inside the smart card. These tasks are still being perform inside the phone. There is a need to perform these task with the help smart card. As there is no secure microSD card available in Pakistan market so there is a need to work on it. Whole system is tested on emulator. There is a need to test whole system on actual devices (secure microSD card and Android phone).

# Appendix A

## Smartcard (Javacard) Applet Code for Bitcoin Wallet

---

```
package com.nust.wallet;

import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISO7816;
import javacard.framework.ISOException;
import javacard.framework.JCSystem;
import javacard.framework.OwnerPIN;
import javacard.framework.Util;
import javacard.security.DESKey;
import javacard.security.KeyBuilder;
import javacard.security.RandomData;

public class wallet extends Applet {

    private final static byte Wallet_CLA = (byte) 0x80;
    private final static byte VERIFY_PIN = (byte) 0x20;
    private final static byte UPDATE_PIN = (byte) 0x21;
    private final static byte GENERATE_KEY = (byte) 0x30;
    private final static byte GET_KEY = (byte) 0x31;

    // Maximum PIN Try Limit
    final static byte PIN_TRY_LIMIT = (byte) 0x03;

    // Maximum PIN Size
    final static byte MAX_PIN_SIZE = (byte) 0x04;
```

## APPENDIX A. SMARTCARD (JAVACARD) APPLLET CODE FOR BITCOIN WALLET50

```
// Signal=> PIN Verification Failed
final static short SW_VERIFICATION_FAILED = 0x6300;

// Signal=> PIN Validation is Required
final static short SW_PIN_VERIFICATION_REQUIRED = 0x6301;

private OwnerPIN pin;
private byte[] keyBytes;
RandomData rng;
DESKey key;

private wallet(byte bArray[], short bOffset, byte bLength) {
    pin = new OwnerPIN(PIN_TRY_LIMIT, MAX_PIN_SIZE);

    byte iLen = bArray[bOffset]; // AID Length
    bOffset = (short) (bOffset + iLen + 1);
    byte cLen = bArray[bOffset]; // Info Length
    bOffset = (short) (bOffset + cLen + 1);
    byte aLen = bArray[bOffset]; // Applet Data Length
    bOffset = (short) (bOffset + 1);
    byte pinLen = bArray[bOffset];

    // The Installation Parameters Contain the PIN
    // Initialization Value
    byte defaultPin[] = { 0x01, 0x02, 0x03, 0x04 };
    pin.update(defaultPin, (short) 0, (byte) 4);

    // Key Generation Pre_Work
    keyBytes = JCSysytem.makeTransientByteArray((short)
        16, JCSysytem.CLEAR_ON_DESELECT);
    rng = RandomData.getInstance(RandomData.ALG_SECURE_RANDOM);
    key = (DESKey)
        KeyBuilder.buildKey(KeyBuilder.TYPE_DES, KeyBuilder.LENGTH_DES3_2KEY,
            false);

    register();
}

public static void install(byte[] bArray, short bOffset, byte
    bLength) {
    // GP-compliant JavaCard Applet Registration
    new wallet(bArray, bOffset, bLength);
}
```

APPENDIX A. SMARTCARD (JAVACARD) APPLLET CODE FOR BITCOIN WALLET51

```
public boolean select() {
    return super.select();
}

public void deselect() {
    pin.reset();
    super.deselect();
}

public void process(APDU apdu) {
    // Good practice: Return 9000 on SELECT
    if (selectingApplet()) {
        return;
    }
    byte[] buf = apdu.getBuffer();

    if (buf[ISO7816.OFFSET_CLA] == Wallet_CLA) {
        switch (buf[ISO7816.OFFSET_INS]) {

            case VERIFY_PIN:
                verify(apdu);
                return;
            case UPDATE_PIN:
                updatePin(apdu);
                return;
            case GENERATE_KEY:
                generateKey();
                return;
            case GET_KEY:
                getKey(apdu);
                return;
            default:
                ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
        }
    } else
        ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
}

private void updatePin(APDU apdu) {
    if (!pin.isValidated()) {
        ISOException.throwIt(SW_PIN_VERIFICATION_REQUIRED);
    }
}
```

## APPENDIX A. SMARTCARD (JAVACARD) APPLLET CODE FOR BITCOIN WALLET52

```
byte[] buffer = apdu.getBuffer();
// Retrieve the PIN Data for Validation.
byte byteRead = (byte) (apdu.setIncomingAndReceive());
pin.update(buffer, (short) ISO7816.OFFSET_CDATA, byteRead);

}

private void verify(APDU apdu) {
    byte[] buffer = apdu.getBuffer();
    // Retrieve the PIN Data for Validation.
    byte byteRead = (byte) (apdu.setIncomingAndReceive());

    // Check PIN
    // The PIN Data is Read into the APDU Buffer
    // At the Offset=> ISO7816.OFFSET_CDATA
    // PIN Data Length = byteRead
    if (pin.check(buffer, ISO7816.OFFSET_CDATA, byteRead) ==
        false) {
        ISOException.throwIt(SW_VERIFICATION_FAILED);
    }
}

private void generateKey() {
    if (!pin.isValidated()) {
        ISOException.throwIt(SW_PIN_VERIFICATION_REQUIRED);
    }

    try {
        rng.generateData(keyBytes, (short) 0, (short) 16);

        key.setKey(keyBytes, (short) 0);
    }
    catch(Exception e){

    } finally {
        Util.arrayFillNonAtomic(keyBytes, (short) 0,(short)
            keyBytes.length, (byte) 0x00);
    }
}

private void getKey(APDU apdu) {
    if (!pin.isValidated()) {
```



APPENDIX A. SMARTCARD (JAVACARD) APPLET CODE FOR BITCOIN WALLET53

```
        ISOException.throwIt(SW_PIN_VERIFICATION_REQUIRED);
    }

    byte[] buffer = apdu.getBuffer();
    apdu.setIncomingAndReceive();

    key.getKey(buffer, (short)0);

    apdu.setOutgoingAndSend((short)0,
        (short)(key.getSize()/(short)8));
    }
}
```

---

# Appendix B

## APDU Command/ Response of Applet

### B.1 Select Wallet Applet

CLA	00
INS	A4
P1	04
P2	00
Lc	05
Data Field	12 34 56 78 90
Le	0F

Example:

APDU Command: 00 A4 04 00 05 12 34 56 78 90

APDU Response: 90 00

### B.2 Verify Wallet Applet PIN

CLA	80
INS	20
P1	00
P2	00
Lc	04
Data Field	01 02 03 04
Le	00

Example:

APDU Command: 80 20 00 00 04 01 02 03 04

APDU Response: 90 00

### B.3 Change Wallet Applet PIN

CLA	80
INS	21
P1	00
P2	00
Lc	04
Data Field	04 03 02 01
Le	00

Example:

APDU Command: 80 21 00 00 04 00 00 00 00

APDU Response: 90 00

### B.4 Secret Key Generation

CLA	80
INS	30
P1	00
P2	00
Lc	01
Data Field	00
Le	00

Example:

APDU Command: 80 30 00 00 01 00

APDU Response: 90 00

## B.5 Getting Secret Key from Wallet Applet

CLA	80
INS	31
P1	00
P2	00
Lc	01
Data Field	00
Le	00

Example:

APDU Command: 80 31 00 00 01 00

APDU Response: 31 1B 49 BC 8E 11 4F 65 B8 D7 A5 28 F9 2F 57 DC  
90 00

# Appendix C

## Screenshots of Application



Figure C.1: Screenshots of Application



Figure C.2: Screenshots of Application

# Bibliography

- [1] H. Takeda, P. Veerkamp, and H. Yoshikawa, "Modeling design process," *AI magazine*, vol. 11, no. 4, p. 37, 1990.
- [2] S. P. Vijay Vaishnavi, Bill Kuechler. (2017, Sep.) Design science research in information systems. Desrist.org. [Online]. Available: <http://desrist.org/design-research-in-information-systems/>
- [3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [4] diamondmac. (2017, Jul.) 100 companies that accept bitcoins as payment. Ebay. [Online]. Available: <https://www.ebay.com/gds/100-Companies-That-Accept-Bitcoins-As-Payment-/10000000206483242/g.html>
- [5] Blockchain. (2017, Jul.) Bitcoins in circulation - blockchain. [Online]. Available: <https://blockchain.info/charts/total-bitcoins>
- [6] CoinDesk. (2017, Jul.) Bitcoin price index - real-time bitcoin price charts. [Online]. Available: <https://www.coindesk.com/price/>
- [7] ibtimes. (2017, Jul.) Bitcoin now accepted by 100,000 merchants worldwide. [Online]. Available: <http://www.ibtimes.co.uk/bitcoin-now-accepted-by-100000-merchants-worldwide-1486613>
- [8] CoinMarketCap. (2016, Aug.) Crypto-currency market capitalizations. [Online]. Available: <https://coinmarketcap.com/>
- [9] Quartz. (2017, Jul.) Android just hit a record 88market share of all smartphones. [Online]. Available: <https://qz.com/826672/android-goog-just-hit-a-record-88-market-share-of-all-smartphones/>
- [10] Android. (2017, Aug.) Android security bulletins. [Online]. Available: <https://source.android.com/security/bulletin/>

- [11] A. Security. (2017, Sep.) Android security acknowledgements. [Online]. Available: <https://source.android.com/security/overview/acknowledgements>
- [12] C. Details. (2017, Aug.) Google android : List of security vulnerabilities. [Online]. Available: [https://www.cvedetails.com/vulnerability-list/vendor\\_id-1224/product\\_id-19997/Google-Android.html](https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html)
- [13] A. V. Org. (2017, Sep.) Android vulnerabilities. [Online]. Available: <http://androidvulnerabilities.org/>
- [14] Github. (2017, Sep.) Bitcoin wallet. [Online]. Available: <https://github.com/bitcoin-wallet/bitcoin-wallet>
- [15] Coinbase. (2017, Sep.) Buy/sell digital currency - coinbase. [Online]. Available: <https://www.coinbase.com>
- [16] O. W. A. S. Project. (2017, Sep.) Android testing cheat sheet - owasp. [Online]. Available: [https://www.owasp.org/index.php/Android\\_Testing\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Android_Testing_Cheat_Sheet)
- [17] K. Piirainen, R. A. Gonzalez, and G. Kolfshoten, “Quo vadis, design science?—a survey of literature,” in *International Conference on Design Science Research in Information Systems*. Springer, 2010, pp. 93–108.
- [18] J. E. v. Aken, “Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules,” *Journal of management studies*, vol. 41, no. 2, pp. 219–246, 2004.
- [19] S. T. March and G. F. Smith, “Design and natural science research on information technology,” *Decision support systems*, vol. 15, no. 4, pp. 251–266, 1995.
- [20] Android. (2017, Aug.) Storage options — android developers. [Online]. Available: <https://developer.android.com/guide/topics/data/data-storage.html>
- [21] R. Skudnov, “Bitcoin clients,” *Instructor*, no. 3, p. 32, 2012.
- [22] S. Eskandari, J. Clark, D. Barrera, and E. Stobert, “A first look at the usability of bitcoin key management,” *arXiv preprint arXiv:1802.04351*, 2018.



- [23] A. Montanez, "Investigation of cryptocurrency wallets on ios and android mobile devices for potential forensic artifacts," *Dept. Forensic Sci., Marshall Univ., Huntington, WV, USA, Tech. Rep*, 2014.
- [24] C. Mann and D. Loebenberger, "Realizing two-factor authentication for the bitcoin protocol." *IACR Cryptology ePrint Archive*, vol. 2014, p. 629, 2014.
- [25] S. Goldfeder, J. Bonneau, J. Kroll, and E. Felten, "Securing bitcoin wallets via threshold signatures," 2014.
- [26] C. Station. (2017, Aug.) Crackstation - online password hash cracking - md5, sha1, linux, rainbow tables, etc. [Online]. Available: <https://crackstation.net/>
- [27] G.-T. E. Family. (2017, Jun.) Go-trust encrypter family. [Online]. Available: <http://www.go-trust.com/en/go-trust-encrypter-family/>
- [28] J. Wayne. (2017, Aug.) Smart card authentication for mobile devices. NIST. [Online]. Available: [http://csrc.nist.gov/groups/SNS/mobile\\_security/documents/mobile\\_devices/pp-btScardAuthentication-final.pdf](http://csrc.nist.gov/groups/SNS/mobile_security/documents/mobile_devices/pp-btScardAuthentication-final.pdf)
- [29] W. Jansen, S. Gavrila, C. Sveillac, and V. Korolev. (2017, Sep.) Nistir 7206, smart cards and mobile device authentication: an overview and implementation. NIST. [Online]. Available: <http://csrc.nist.gov/publications/nistir/nist-IR-7206.pdf>