



University of Nairobi
School of Computing and Informatics

**Mobile – Based Multi-Factor Authentication Scheme for Mobile
Banking**

By

Ombiro Zablon B. Handson (P53/79466/2015)

Supervisor

Dr. Christopher Chepken

November 2016

**Submitted in partial fulfillment of the requirement of the Master of Science Degree
in Distribute Computing Technology of the University of Nairobi**

Declaration

I, **Ombiro Zablon B. Handson**, hereby declare that this research project and the work presented in it, is my original work and that it has not been presented for any other University award.

Signature: _____ Date: _____

Name: **Ombiro Zablon B. Handson**

Reg. No.: **P53/79466/2015**

This project has been submitted in partial fulfillment of the requirement of the Master of Science Degree in Distribute Computing Technology of the University of Nairobi with the approval of the University Supervisor.

Signature: _____ Date: _____

Name: **Dr. Christopher Chepken**

School of Computing and Informatics

University of Nairobi

Dedication

To my dear wife Damaris, my lovely daughters Siena Chloe and Alma Ciel for your love,
patience, and understanding.

Acknowledgment

Much thanks to our Lord God, for the love, life and knowledge He has given me. My parents Jerome and Alice, *merci beaucoup* for readying me for scholarship.

Secondly, I acknowledge and appreciate the support and guidance of my project supervisor, Dr. Christopher Chepken; the panelists: Professor William Okello, Dr. Evans Miriti, and Dr. Stephen Mburu whose corrections and insights shaped my academic pursuits; and my classmates for their criticism during the research period.

Finally, I appreciate the moral support from my wife, Damaris Kerubo, my daughters Siena Chloe and Alma Ciel. Life won't have been any easier without your love.

Abstract

Authentication for all mobile initiated financial transaction is a mandatory requirement. USSD applications authenticate using PIN and Phone number while native applications can have further authentication inbuilt or provided for by third a party. The level of security for a given authentication scheme depends on attribute combination, authentication channel, credential storage, and encryption. A number of researches have been conducted on mobile based authentication and their level of security. However, there is limited research on authentication schemes that combines attributes asynchronously, securely and efficiently.

Mobile payment transactions are vulnerable when using single and two-factor authentication schemes. This research project proposes a combination of multiple factors – PIN, One-time password (OTP), flash call interception, device specific soft tokenization using IMEI, and encryption these attributes using AES 256 bit in mobile banking applications. The solution uses one user-supplied attribute while the rest are authenticated asynchronously in the background. The storage of credentials is in distributed locations. This architecture provides increased security from identity theft, sniffing attacks, dictionary attacks, and man in the middle attacks.

A software solution was developed using prototyping in a waterfall model. Authentication time delays, delivery mechanism were measured and analyzed. Using Kernel Density Estimation, the results showed that combination of PIN and OTP had shorter time delays followed by PIN and phone call combination and OTP and phone call combination in that order. In the background, credentials were encrypted and the mobile device was identified and authenticated.

Table of Contents

Declaration	i
Dedication	ii
Acknowledgment	iii
Abstract	iv
List of Figures	ix
List of Tables	x
List of Abbreviations	xi
Definition of Terms	xii
Chapter One: Introduction	1
1.1 Background of the Study	1
1.2 Problem Statement	2
1.3 Research Objectives	3
1.4 Significance of the Study	4
1.5 Project Justification.....	4
Chapter Two: Literature Review	6
2.1 Introduction.....	6
2.2 A Review of Theoretical Literature	6
2.2.1 Single-Factor Authentication (SFA)	6
2.2.2 Two-Factor Authentication (2FA)	7
2.3 Multi -Factor Authentication (MFA)	8
2.3.1 Mobile Device-Based Authentication.....	9
2.3.2 Computer Simulated Software Tokens	10
2.3.3 One Time Pad	10

2.3.4 Voice Biometric Authentication	11
2.4 A Review of Empirical Literature.....	11
2.5 Mobile Banking in Kenya.....	13
2.6 Mobile Banking Fraud	14
2.7 Research Gaps.....	15
2.8 Conceptual Framework.....	15
2.9 Summary	16
Chapter Three: Research Methodology and System Development	17
3.1 Introduction.....	17
3.2 Choice of Development Model.....	17
3.2.1 Waterfall Model	17
3.2.2 Prototyping.....	18
3.3 Data Collection and Measurement.....	19
3.4 System Design and Development	20
3.4.1 Development Tools	20
3.4.2 Testing Tools	20
3.4.3 Application Modules.....	21
3.5 User Interface Design	22
3.6 Database Design.....	25
3.7 Process Flow	27
3.8 Actual Application User Interface	33
3.9 System Performance Results.....	35
Chapter Four: Research Results and Analysis	36

4.1 Introduction.....	36
4.2 Testing Scripts	36
4.3 PIN and OTP Authentication Report	37
PIN and OTP Authentication Pass Results	37
PIN and OTP Authentication Pass Kernel Density Plot	38
PIN and OTP Authentication Report Interpretation	38
4.4 PIN and Phone Authentication Report.....	38
PIN and Phone Authentication Pass Results.....	38
PIN and Phone Authentication Pass Kernel Density Plot.....	40
PIN and Phone Authentication Report Interpretation.....	40
4.5 OTP and Phone Authentication Report.....	40
OTP and Phone Authentication Pass Results.....	40
OTP and Phone Authentication Pass Kernel Density Plot.....	42
OTP and Phone Authentication Report Interpretation.....	42
4.6 Authentication Failed Report.....	42
Authentication Failed Kernel Density Plot.....	43
Authentication Failed Report Interpretation	44
4.7 Summary.....	44
Chapter Five: Summary and Recommendations.....	45
5.1 Introduction.....	45
5.2 Summary of the Study	45
5.3 Further Developments.....	46
5.4 Limitations of the Study.....	46

5.5 Future Research areas	46
5.6 Conclusion	46
References	48

List of Figures

Figure 1 Conceptual Framework Representation	16
Figure 2 Water Fall Software Development Model.....	18
Figure 3 Prototyping Model.....	19
Figure 4 Multi-Factor Authentication Process Flow Chart.....	22
Figure 5 Step 1: Phone Registration	23
Figure 6 Step 3: PIN Registration.....	23
Figure 7 Step 4: Choosing Authentication Scheme	24
Figure 8 Step 5: Authenticating Based on Chosen Scheme Type	24
Figure 9 Step 6: Mobile Banking Deposit	24
Figure 10 Step 7: Mobile Banking Withdrawal.....	25
Figure 11 Database Scheme Showing Table Relationships.....	26
Figure 12 Sinch API Validation Process	27
Figure 13 Process Flow Diagram - Entity Relationship Diagram	28
Figure 14 Application Download, Phone and PIN Registration.....	33
Figure 15 Login Preference, PIN, and OTP Authentication	34
Figure 16 OTP Logs, OTP and Phone Call Authentication and Banking Interface	34
Figure 17 PIN and OTP Authentication Pass Kernel Density Plot.....	38
Figure 18 PIN and Phone Authentication Pass Kernel Density Plot	40
Figure 19 OTP and Phone Authentication Pass Kernel Density Plot	42
Figure 20 Authentication Failed Kernel Density Plot.....	43

List of Tables

Table 1 Mobile Data Main Table.....	25
Table 2 Mobile Data Main Offline Table	26
Table 3 Metrics Log Table.....	26
Table 4 Test Scripts Table	36
Table 5 PIN and OTP Authentication Pass Results	37
Table 6 PIN and Phone Authentication Pass Results.....	39
Table 7 OTP and Phone Authentication Pass Results	41
Table 8 Authentication Failed Results.....	43

List of Abbreviations

2FA	Two Factor Authentication
API	Application Programming Interface
CAK	Communications Authority of Kenya
CER	Crossover Error Rate
EER	Equal error rate
GSM	Global System for Mobile Communications
IMEI	International Mobile Equipment Identifier
JSON	JavaScript Object Notation
KDE	Kernel Density Estimation
MFA	Multi-Factor Authentication
MPS	Mobile Payment System
OOBA	Out-of-Band Authentication
OTP	One Time Password (Pad)
PIN	Personal Identification Number
REST	Representational State Transfer
SFA	Single-Factor Authentication
SMPP	Short Message Peer-to-Peer
SSH	Secure Shell

Definition of Terms

M-PESA: (**M** for mobile, **Pesa**, *Swahili* for money) is a mobile phone-based money transfer and microfinancing service, launched in 2007 by Vodafone for Safaricom and Vodacom, the largest mobile network operators in Kenya and Tanzania.

USSD (Unstructured Supplementary Service Data) is a technology used by mobile network providers in sending messages using shortcodes from a mobile station to an application for querying and returning results. It is used for querying and returning airtime balances, mobile payments and chatting.

Chapter One: Introduction

1.1 Background of the Study

Business financial applications that enable user's ability to transact from their mobile devices to the traditional bank account require that the user supplies given entities that uniquely identify them and which can be validated against already existing entities held by a trusted third party. That is authentication which will give or deny access to transactional services to the user. In multi-factor authentication (MFA), a digital access control is used to allow a user access after successfully presenting combined pieces of attributes as evidence to an authentication provider - typically based on at least two of the following categories: knowledge (something they know); possession (something they have), and inherence (something they are).

The interest for mobile banking services has widely grown in the past few years. However, this type of service is both business and user sensitive on security and usage. Indeed, money is an attractive target for attackers and fraudsters. The service can also be misused for money-laundering and illegal funding. Moreover, this type of service should comply with the financial policy enforced in the country where the service is installed (Gaber, Gharout, Achemlal, Pasquet, & Urien, 2016).

Most mobile banking systems rely on single non dynamic PIN/passwords to verify the user's identity. These passwords face major challenges in management and security. For example, users may use easy passwords that can be guessed, or similar passwords for different accounts, or store their passwords on devices or write them on a piece of paper. Such management and security challenges may be a weak point for hackers to steal passwords through shoulder surfing, snooping, sniffing, guessing, etc. (Aloul, Zahidi, & El-Hajj, 2009).

Multi-factor authentication for mobile payment systems (MPS) can use mobile devices to serve as "something that the user possesses"; a PIN as "something the user knows" and a random one-time password (OTP) for an extra layer of security. The one-time password is normally a randomly generated alphanumeric or numeric code that is sent to a mobile

device in the form of an SMS. It can also be sent but deleted by the application once validation is done not to allow the user to have a view of it. Most people carry with them digital devices all the time making using an OTP an adequate dedicated token. However, the strength of any such authentication scheme will depend on the combinations of one or more of the three properties: what the user knows, what the user has or what the user is as attributes.

1.2 Problem Statement

Use of mobile devices for value-added services other than the traditional voice calls and SMS is on the increase. Value-added services have ranged from media transfers, accessing the internet, playing games and now, as a mobile banking platform. These devices increase platforms on which access to services that traditionally were only accessed through a computer PC to be accessed through the digital device. Among the services that are now accessed through the mobile device is mobile banking. This has expanded the banking ecosystem. Thales security (2014) indicates that such services have brought new players in the banking industry to include mobile operators and handset manufacturers. These new platforms have bred competition amongst banking providers leading to new banking innovations specifically targeting the mobile user. However, using mobile devices to offer platforms for banking has its own challenges credential storage, authentication as well as having a trust relationship with the user as there is no physical contact when conducting a financial transaction. The challenge of trusting the authenticity of a mobile originating transaction is as a result of weak authentication schemes which are targeted by fraudsters, having unstable communication channels and weaknesses in storage and transmission of authentication credentials.

Mobile banking solutions face new security vulnerabilities (Thales Security, 2014). Provisioning of user banking credentials can be done through over-the-air (OTA) and through custom applications. This creates a loophole for eavesdroppers to listen on the network channels and steal customer data. Stolen credentials can be used to access customer information which if disclosed will lead to fines and other litigations. It is critical that there is an understanding of how and where sensitive account data is to be stored and

transmitted. Taking into consideration security risks for a bank and customers, financial organizations will then define and implement data protection mechanisms in applications that user uses to access banking services. However, the security mechanisms should not be complex for the user and should not add any extra cost.

Single-factor authentication (Aloul et al., 2009), for example using passwords, is not secure for mobile and the internet originating transactions. Passwords that are based on first or last names, account holders age, or date of birth, can easily be predicted using dictionary attack applications that are freely available to hackers. Two-factor authentication is being used to mitigate security weaknesses of single factor authentication by using other authentication attributes and increasing options for user account details.

The use of common passwords and PINs and their vulnerabilities on them has been explored (Berry, 2012). Research shows that PINs are weak and vulnerable to simple hacking. "Armed with only four possibilities, hackers can crack 20% of all PINs. Allow them no more than fifteen numbers, and they can tap the accounts of more than a quarter of card-holders."(Leigh, 2013). Use of single-factor authentication such as the use of PIN or password can lead to a risk of compromise when compared with dual-factor authentication scheme. Two-factor authentication is also weak and vulnerable to known attacks. Mobile banking solutions security cannot be guaranteed with single factor authentication. Use of multi-factor authentication with a mix of device specific attributes such as geolocation, device identity and voice biometrics combining with a random one-time password and randomly generated SMS code can be used to address single and two-factor authentication weakness.

1.3 Research Objectives

The research encompasses a study and implementation of a mobile-based multi-factor authentication scheme for mobile banking.

Specific objectives are:

1. To evaluate multi-factor authentication schemes.

2. To determine a mobile-based multi-factor authentication scheme for mobile banking that uses a combination of attributes, stores credentials in distributed servers and is time efficient.
3. To implement a time efficient, secure mobile-based multi-factor authentication scheme using device specific ID, flash call/phone call, a randomly generated one-time password and a PIN for securing mobile banking.

1.4 Significance of the Study

The project will develop a mobile-based multi-factor authentication scheme that will provide an extra layer of security for mobile banking solutions. The implementation will address the limitation of single, two and existing multi-factor authentication schemes. The study and the project will provide a basis for further research in multi-factor authentication and implementation of authentication for mobile payments.

1.5 Project Justification

Mobile money transfer service according to CAK, July – September 2015 report continues to record a continuous increase in usage. Its popularity combined with ease of use makes mobile based transactions a better option for users. During the period under consideration, there was a 3 percent increase in mobile money transfers from 27.7 to 28.7 million. These statistics are good for banking and financial providers. The statistics also give hackers impetus to target mobile banking platforms to steal money from institutions and the users.

Instead of targeting banks, cyber criminals are now routinely targeting payment systems and mobile money service providers with which to access bank systems. In 2015, Safaricom, the leading mobile money provider in Kenya rolled-out of the new service in which M-PESA users are able to see the names of the intended recipients of mobile money transfers to reduce chances of sending money to the wrong person or to fraudsters. (Muchai, Kimani, Kigen, Mwangi, & Shiyayo, 2015) indicated that Kenya has had a sharp increase in the financial fraud perpetrated over mobile devices using mobile banking services. The fraud further exploit system and customer manipulation to send money to recipients who exploit mobile network and mobile banking weaknesses. This project

addresses some of the mobile banking application security loopholes in the current mobile banking solutions by proposing a stronger and efficient authentication scheme.

The project integrates distributed systems; systems security through the use of encryption protocols, hashing, voice calls and random one-time passwords for multi-factor authentication; to offer increased security for mobile banking solutions. This solution will help in curbing money laundering activities from unknown devices and fraudulent users.

Chapter Two: Literature Review

2.1 Introduction

The theoretical framework that will guide the study of multi-factor authentication in mobile banking solutions, empirical studies and conclusions are all outlined in this chapter. The chapter situates the current study within the relevant body of literature that is required to find answers and connect to our research objectives. The literature will be a review of the different authentication schemes in mobile banking solutions. A summary of conclusions drawn from the literature review that relates to the current topic of study is also presented.

2.2 A Review of Theoretical Literature

The theoretical literature on various authentication schemes is evaluated, from the aspects of existing research, implementations, and application to secure authentication.

2.2.1 Single-Factor Authentication (SFA)

Single-factor authentication (SFA) is defined as using a single attribute to give or deny access to a service. The service can be accessed through a web interface or over a mobile device. The SFA scheme uses one set of credentials to identify and give access to the entity requesting the services. The most common SFA is uses of password or PIN to access privileged services. Password/PIN security depends on the safety and integrity of the administration as well as it internal security mechanisms in setting up and managing user details for access. For better security with this scheme, strong passwords, passwords which expire, and a combination of another non-numeric character is used to make the security of passwords/PINs stronger. (Margaret, 2015).

Use of one attribute to access a resource creates a challenge for users creating simple and easy to predict combinations. Creating both a strong and memorable password becomes the prerogative of the administrators or as an enforced requirement from the providers of the services. However, still, passwords with shorter length and complexity are common. These passwords can be subject to dictionary attacks and are therefore vulnerable to misuse. It is important organizations that provide financial services enforce policies for not

easy to predict passwords be it to a machine or human mind. A good password should not be easy to guess, or crack even using brute force, dictionary or any other method.

To have complex passwords, the length, use of capital letters, numerals and special characters increases entropy due to the larger character set (Bruce, 2014). Complex passwords do not abstract users from having their passwords cracked by brute force, dictionary and rainbow table attacks, but makes it a bit harder. If a hacker accesses the password database, then the whole password security is compromised. However, administrators can configure their password system providers to create and append random characters to the hashes of password encryption. This can help against dictionary-based attacks and social engineering such as phishing. Use of passwords alone will not be adequate, other schemes such as biometric authentication distributes credential storage and passwords that expire quickly can provide strong alternatives.

2.2.2 Two-Factor Authentication (2FA)

Two-factor authentication (TFA, T-FA or 2FA) is an authentication approach that makes use of two or more of the three authentication domains: a knowledge factor ("something only the user knows"), a possession factor ("something only the user has"), and an inherence factor ("something only the user is"). These factors are then validated by the authenticating party for authenticity for access to be granted. Some applications such as USSD using the PIN can have the phone number generating the request tied with the session in order for authentication to take place.

Many computer devices use two-factor authentication to give access to resources. In this model, a user who is requesting access presents some evidence based on a known attribute to the authenticating provider to countercheck with what is already stored. When the two matches, authentication is passed otherwise the user is denied access. (Sasidevi, Sugumar, & Priya, 2015). With two-factor authentication, the chance of the user requesting access providing wrong information is reduced by having another party validate against what is in the database. Nevertheless, this is not full proof as applications have been developed

which can masquerade as the user and provide the right credentials. Therefore, use of multiple attributes will decrease the probability of identity theft.

Two-factor authentication requires use of two of the following three authentication domain factors:

1. Something only the user knows (e.g., password, PIN, pattern);
2. Something only the user has (e.g., ATM card, smart card, mobile phone); and
3. Something only the user is (e.g., biometric characteristic, such as a fingerprint).

In banking with an automated teller machine (ATM) (Adeoye, 2012), uses a physical card (ATM card) as "something the user has" and a PIN as "something the user knows". This is an example of a two-factor authentication in which two attributes are compared together to match what the bank system has in store, a failure which denies access to the banking services. In this case use of knowledge factor and possession factor is used to have two-factor authentication of a banking service.

Two-factor authentication (or multi-factor authentication) does not necessarily mean a stronger and secure authentication. An authentication to be both strong and secure, a number of factors have to be considered – ease of breaking the authenticating, and usability. Two-factor authentication may be considered strong when multiple answers for a challenge are considered. Nevertheless, if the challenge questions are of the forms: "something the user has" or "something the user is", the scheme is considered to be two-factor authentication. It may use a PIN and an ATM card for an ATM banking transaction (Agoyi & Seral, 2011). Such a scheme can be advanced by including the use of encrypted SMS message which will be used to authenticate the user credentials before any transaction is authorized. Using encrypted SMS improves banking security against frauds and crime.

2.3 Multi -Factor Authentication (MFA)

The multi-factor authentication scheme is the implementation of authentication using a combination of two or more attributes based on what a user knows, what a user is and what a user has. The following are multi-factor authentications that have been studied and implemented:

2.3.1 Mobile Device-Based Authentication

Apart from a one-time password or PIN or password that a user may have, a mobile phone can be used in as something a user has. With the phone, a voice call synchronized to an online session can be used as part of an account validation, an authentication for, a transaction verification, or as part of a transactional activity validation in a high-risk transactions (Council, 2011).

With the use of two-factor authentication (2FA) schemes, security is strengthened by enhancing password-based systems with secondary tokens. Using mobile devices as part of authentication will eliminate the need for an additional hardware for storing and handling secondary authentication data. The phone device, therefore, enables increased security, reduces cost and ease of use. The use of digital devices as part of authentication is now used in mobile and online banking by financial banking providers as well as other service providers (Dmitrienko, Liebchen, Rossow, & Sadeghi, 2014).

Out-of-band authentication (OOBA) is the use of two separate networks simultaneously to validate user credentials. This eliminates dependency on one channel for completing a transaction. A hacker will have to break into the two separate channels that are separate and independent in order to full gain access to an account. This makes Out-of-band authentication a powerful tool against financial fraud for a compromised channel or account (EMC, 2011). An example of OOBA is where a secure device is paired with user gestures on a device with a built-in accelerometer to transmit authentication data over a network (Chagnaadorj & Tanaka, 2014).

Using a mobile phone for out-of-band communication and authentication ensures to some extent that the right users have access to the right resources thwarting attempts to breach accounts (Andrew, 2007). For example, out of band authentication may be invoked whenever a transaction of a specified threshold amount is requested. In this case, a phone call is made to the account owner to validate the transaction to completion. In this case, only the rightful owner will authorize the transaction and fraud are not possible unless eh owner is under duress such as kidnapping.

Using a trusted service manager (TSM) (Alliance, 2009) for proximity mobile payments abstracts banking providers from mobile device complexities by having account data stored with a trusted provider rather than in a physical card. The user account data originates from the banking provider and is securely transmitted to the TSM then to the handset. The data in transit is encrypted protect the data from eavesdropping and sniffing attacks.

Using wired phones for authentications has little if any research was done. (Mohamed, 2014); however, there is increased interest in the use of mobile devices and mobile applications to offer authentication alternative. Mobile devices can be used as a token through incorporating encrypted SMS text, interactive voice calls, or by having a custom native application on the device itself but which communicates with the third provider through trusted channels. The trusted channels provide alternate routes that offer an extra security layer. Other forms of authentication being explored include the use of secret images and QR_Code which can be scanned and validated with other digital devices having a common algorithm.

2.3.2 Computer Simulated Software Tokens

A mobile device can be used as a soft token by having a software installed on them which communicates with a third party application to authenticate. The device becomes the possession factor for the user. With the application deployed on the digital devices, cost of buying hardware based tokens are reduced. This approach has vulnerabilities to for example the Zeus Trojan (Dmitrienko et al., 2014), a virus which manipulates an online banking session by requiring infected users to enter their mobile phone number. However, with the use of carried tokens (Aloul et al., 2009) and the cost of manufacturing and maintaining mobile devices being transferred to both the user and the provider, an alternative is to install a software token on the mobile phone to be used for authentication.

2.3.3 One Time Pad

Password combine characters and numbers and are memorized for long term use or temporary. For a one-time password (OTP), a random character combination is generated

and time bound before it expires. OTP requires generation and expiry of a new random password for each authentication. Use of a grid card to request a set of characters from a password known only to the account owner can be a source of random passwords. This protects against replay attacks and not against duplication of the whole grid (Panse, 2014).

2.3.4 Voice Biometric Authentication

Voice is one of the emerging biometrics that is like fingerprints and the iris is unique to an individual. With the use of mobile devices for voice calls, algorithms are in place that can match a voice to an account. This can be done accurately with minimal errors and it is done at a remote location. Through numerical representation of the unique sounds, patterns, and rhythms of an individual's voice, a match is conducted against what is in the database to validate the speaker.

Biometrics are based on probabilistic measures which are fairly unpredictable. In such schemes, the chance of prediction is extremely low as well as an error with any probabilistic chance is extremely low. Voice biometric are mathematical models that represent a physical characteristic of the individual. Though such a model face errors, they are minimal enough not to allow a digital representation an error from it is analog representation.

Voice biometrics have characteristics of adjustments for false and positive negatives making them better compared with other biometric measurements. This makes voice biometrics suited for multi-factor authentication, especially for mobile based applications. Voice biometrics offer a degree of certainty that a given authentication attribute is correct. A study at the IBM Watson Research Center, for example, showed that false positives in a voice biometric study were less than 0.00001% of the cases, against .8% false negatives. This shows that very high and accurate results from biometrics alone can be used to reduce inadvertent hackers from gaining access.

2.4 A Review of Empirical Literature

(Antal & Szabó, 2015) carried out research on the touch screen based swipe patterns for biometric authentication. The study investigated user authentication on mobile devices

with touch behavior and micro movements of user created a pattern on a mobile device. The research showed that using sequences of 5 swipes improved the EER rate by 0.2%. Although the study did not implement device based multi-factor authentication, the study showed that single swipes alone with EER values of 0.05 were not enough to permit implementation of an authentication procedure. However, using 5 consecutive swipes in this procedure increases EER. The research did not consider any other authentication scheme to that can be combined with the biometric method studied.

A research conducted on Voice Pitch Based Authentication for Android Application by (Jadhav, Shirsat, Bhargude, & Kamble, 2016) proposed triggering an authentication based on a keyword that depends on the speaker recognition techniques. The voice trigger performs recognition of both the unique keyword and user voice. This can be achieved without using speech synthesis algorithms that have high demand on the computational power of the devices in use. The research used two methods; a template based method and a hidden Markov model (HMM) based method. The results showed that templated based method was 4.1 times faster compared to the hidden Markov method.

Authentication can be achieved using what the customer has (a digital device) and what they know (PIN) (Adeoye, 2012). (Adeoye, 2012) in his study on mobile banking, infrastructure, and mobile banking trends showed that two-factor authentication is better than one-factor authentication, it also found out that the 2FA was porous. The study recommended customer education be used to educate the users on skimming techniques and that banking providers should not offer liability against PIN credit and debit card losses. The study, however, failed to show that adding another layer of authentication will decrease the porosity of two-factor authentication. Further, (Cha, Lee, Park, & Ji, 2015) proposes security enhancement of micro-payment systems with the user based knowledge-based authentication and using a smart watch to have possession-based authentication. The study fails to show how that implementation can be done to increase the security of mobile banking.

(Corella & Lewison, 2012) outlines use of one or multiple factors for authentication of mobile applications. In this setup, an application resident on the mobile device of the user

will have credentials that are generated during the initial installation of the application. The data set for the credentials is a device handle and a key pair to a public key cryptosystem. The user suggests use of RSA in this set up for cryptography. This approach requires improvements to the operating system of the mobile device to allow a single native application to be used securely within the environment notwithstanding other non-trusted applications within the device.

In a study carried out by (Mohamed, 2014), the study proposed a system based on three-factor-authentication with password for login as opposed to using pin or pattern because, in shoulder surfing attacks, password is safer; wireless Bluetooth-based token, because of it is security against shoulder surfing attacks and other attacks, and ear biometric factor because of its uniqueness and advantages over the others types of biometrics. The proposed authentication scheme is not convenient for mobile banking due to the use of ear as a biometric. The use of blue tooth based token is also cumbersome as the blue tooth service has to be switched on and will consume more power. This proposal does not address the convenience and security loopholes that may be created by using blue tooth communication to authenticate.

Considering applications that are run on a smartphone device and communicate with remote service providers such as one when a smartphone user needs to check his bank account or make some transactions remotely, (Sarhan, Hafez, & Safwat, 2015) uses three authentication schemes: two user chosen passwords (one is known to bank representative and the other is anonymous) and bank generated OTP. This approach is a two-factor authentication and does not explore the use of biometrics as well as it relies on what the user knows and not the other factors – what the user has or is.

2.5 Mobile Banking in Kenya

Use of mobile devices as a platform for mobile banking is an emerging new channel in the space of banking and payments. To gain user adoption of mobile banking and payments, confidence in the security of the mobile banking services should be addressed from within the mobile banking solutions (Pegueros, 2012). (Statistica, 2015) in a report by Statista, worldwide mobile payment volume in 2015 was 450 billion U.S. dollars and is expected

to surpass 1 trillion U.S. dollars in 2019. In a report by Forrester, Over the next five years, US mobile payments will grow from \$52 billion in 2014 to \$142 billion by 2019 with both national brands and local merchants (Denée, 2014).

In Kenya, mobile money transfer service has continued to record a steady growth. In July – September 2015 for example, the number of mobile money agents grew to 135,724 up from 129,357 agents posted during the previous quarter (Communications Authority of Kenya, 2016). According to records from Central Bank of Kenya Annual report 2015, there was a 21.59 percent transaction increment between the year 2014 and 2015, and in total, 25.75 billion shillings worth of transactions were recorded.

Mobile banking solutions in Kenya offer the following services: Deposit/Withdraw money, Transfer (send) money to registered/non-registered customers, buy prepaid airtime/purchase credit, manage money transfer account, Payment of bills and services, Purchase of goods and services, Bulk cash payments such as salary payments, International money transfer- Diaspora, ATM withdrawals, Dividend payment, Social/Charitable collections, Banking – savings and loan products

(Muchai et al., 2015) reported that in 2014 alone, cyber-attacks doubled over the last year in Kenya to stand at 5.4 million. The report further indicates that instead of targeting banks, cybercriminals are targeting mobile banking systems to access bank systems.

2.6 Mobile Banking Fraud

Mobile fraud can be in the following forms (Kevin, 2015):

Phishing: Fraudsters may use phone calls, SMS messages, or email to trick users to divulge their PINs or other personal information that is then used to steal from mobile money accounts.

Split Transactions: Mobile money agents may try to earn more for themselves by breaking up legitimate customer transactions into smaller ones. By doing so, agents can earn more commissions as a result of higher transaction volumes.

Unauthorized SIM Swap: A fraudster may attempt to take over someone else's mobile wallet account by pretending to be that person using false identity documents. Once they

assume the other person's identity, they are able to swap SIM cards and obtain full access to funds.

Identity Theft: This type of fraud is the result of an inside job. Less scrupulous employees may abuse their privileges by accessing and exploiting mobile money customer information, stealing funds from accounts for their own benefit.

Counterfeit KYC: By providing false documents during know your customer (KYC) process, fraudsters can gain access to premium mobile wallets that provide higher limits for fund transfers and withdrawals. This presents an opportunity for money laundering to take place without being detected.

2.7 Research Gaps

The review of literature builds consensus on the need to have multi-factor authentication using more than two attributes (what the user knows, what the user has, and what the user is) in mobile banking solutions. Further in the literature review, all researchers point out that single factor authentication using PIN is both weak and vulnerable. Research work has been done on multi-factor authentication using a number of attributes. These research work, however, fails to have a mix of attributes that are utilizing a combination of common authentication methods, unique mobile device attributes, and voice biometrics to enhance mobile banking solutions security. The majority of the researchers used finger and iris biometrics. Use of voice biometrics has not been used in the literature review in combination with other authentication methods. The literature review does not address the vulnerabilities that exist for PIN including Phishing, Split Transactions, Unauthorized SIM Swap, Identity Theft and Counterfeit KYC.

2.8 Conceptual Framework

The schematic diagram below is a representation showing the relationship between variables for the proposed research. To have a secure mobile banking solution, a number of variables will be built into the application including OTP, PIN, device specific token, and user voice profile. These variables will be used by mobile payments users directly or indirectly. The security of the mobile banking solution will be influenced by the ease of

use, complexity of the authentication factors, communication channels and the cryptographic scheme used.

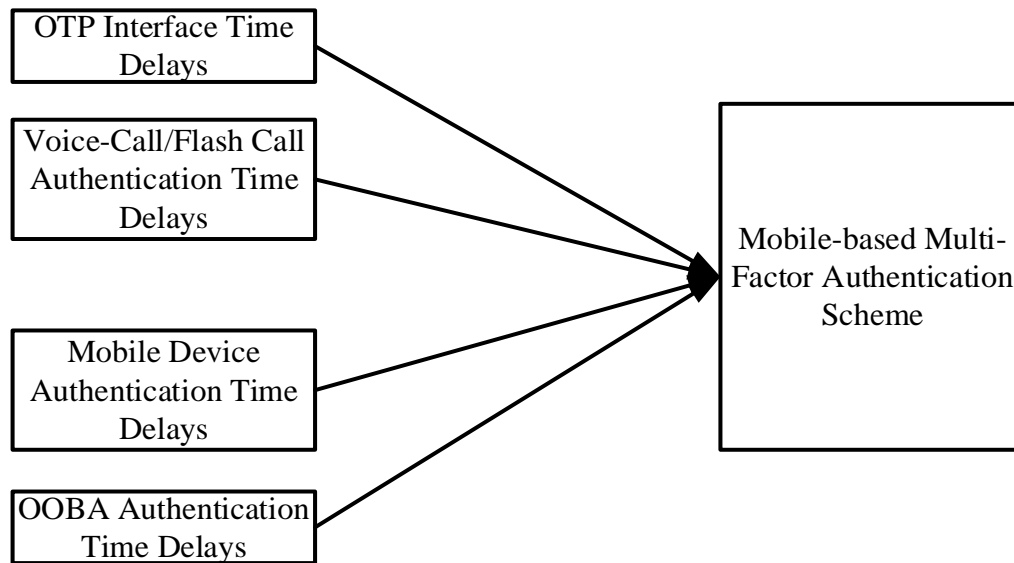


Figure 1 Conceptual Framework Representation

2.9 Summary

A number of research and projects have been done on securing mobile payments. The review of the literature shows consensus that single factor authentication for mobile banking is both a weak and a vulnerable approach while multi-factor authentication using more than two attributes (what the user knows, what the user has, and what the use is) is the favored approach. The literature review nevertheless fails to show an approach with that uses a mix of attributes that offer increased protection against mobile money fraud. Most research studies show the use of a combination of PIN and OTP with user interaction needed to complete authentication. Use of biometrics is another authentication factor for mobile transactions that is most explored in the literature review apart from one-time password and PIN. The combination of voice calls with OTP and PIN and use of device specific software tokens is proposed to address the research gap identified in the literature review.

Chapter Three: Research Methodology and System Development

3.1 Introduction

A software development methodology is a set of rules and guidelines that are used in the process of researching, planning, designing, developing, testing, setup and maintaining a software product. The methodology includes core values that are upheld by the project team and tools used in the planning, development and implementation process (Mihai, 2014). Software Engineering Institute defines Systems Development Methodology in Software Design as a framework for structuring, planning, and controlling information system design and development. Other methodologies for software design and development are waterfall and prototyping. This research shall incorporate new software development and design with prototyping and waterfall models.

3.2 Choice of Development Model

Software Prototyping Model combining with some waterfall model was used to create a prototype. This choice offers quick development and proof of concept of the proposed solution. It also enables incorporating any changes and challenges earlier in the project development stages.

3.2.1 Waterfall Model

The waterfall model is a sequential development approach in which design, development, and implementation are conducted in a sequential manner through the following phases: requirements gathering and analysis, design, implementation, testing (validation), integration and maintenance. (Douglas, 2009) in his comparison of waterfall model and agile methodology argues that a waterfall model is a structured approach; a progressively linear model that follows discrete, easily understandable and explainable phases. The model provides milestones throughout the development cycle. The following was the proposed waterfall phases:

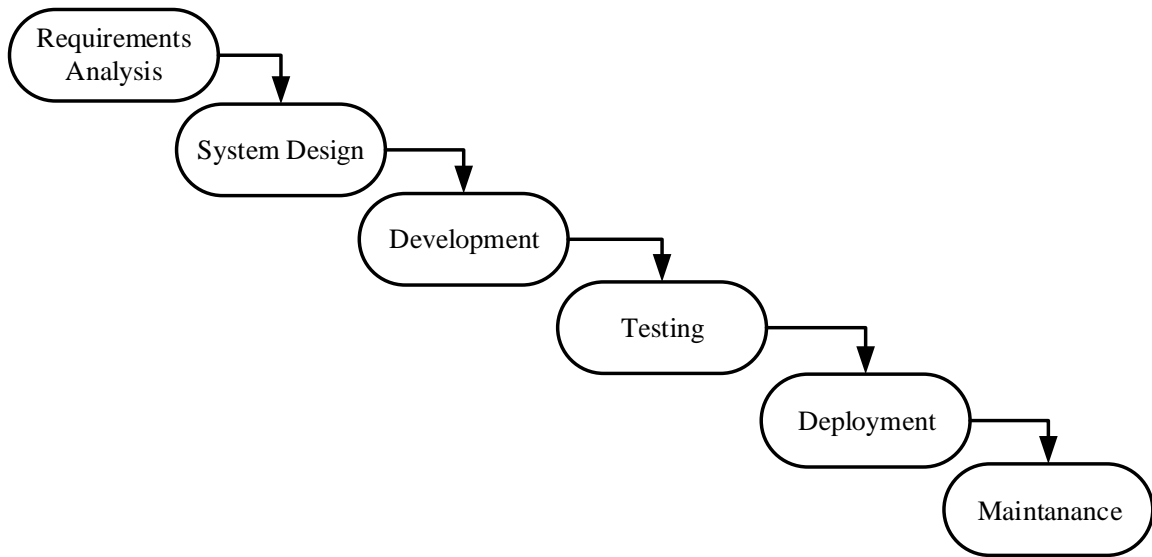


Figure 2 Water Fall Software Development Model

3.2.2 Prototyping

In software development, prototypes are used to emulate a model of the proposed system. The prototype will enable development to uncover any underlying issues and address them early enough before the main application is developed and released. The prototype then acts as a base for replication and further development based on lessons learned (Hackney & Elizabeth, 2015).

The process of prototyping that was used involved the following steps

1. Identification of basic requirements

Basic requirements were determined including the input and output information desired. Details, such as security, can typically be ignored.

2. Development of Initial Prototype

The initial prototype was developed that includes only user interfaces.

3. Review (code and application)

The end-users examined the prototype and provided feedback on additions or changes. This was done over the Google Play Store interface.

4. The prototype was revised and further changes made based on specifications feedback for further improvements.

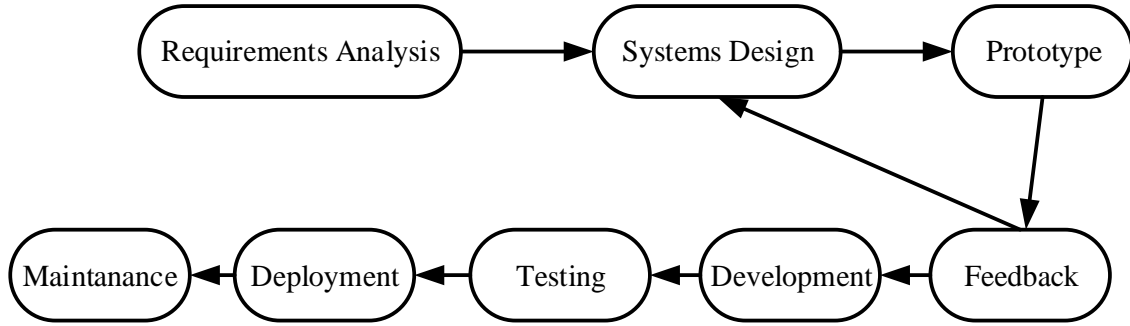


Figure 3 Prototyping Model

3.3 Data Collection and Measurement

The following data was collected from the application database:

- OTP Interfacing Time Delay
- Time delays in voice call validation
- Device soft token authentication Time Delay
- Authentication failure time delays

The data was measured through the use of functions that recorded an occurrence time for each event. Data was collected in excel files and imported to R studio for analysis.

To estimate the random variable of latency in authentication, the Kernel Density Estimation (KDE), a non-parametric method for probability densities estimations was used. The KDE gives a smoothed curve for finite data sample. In KDE plots, for a value of x , a numeric vector of the population, a plot of the plot (density(x)) is used. Let $(x_1, x_2 \dots x_n)$ be an independent and identically distributed sample drawn from some distribution with an unknown density f . The plot will estimate the shape of the function f by taking its kernel density as:

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x-x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right),$$

In the model above, $K()$ is a non-negative function that integrates to one and has mean zero and $h > 0$. The value of h is the smoothing bandwidth which is randomly chosen but can be altered. A kernel with subscript h is a scaled kernel and defined as $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$.

The value of h is chosen as small as possible underlying the trade-off between the bias of

the estimator and its variance. In this research project, h was randomly generated by the system.

3.4 System Design and Development

3.4.1 Development Tools

The system development used the following freely available open source tools. Google Android devices were used for testing.

Operating Systems

- Android Operating System Mobile Device running KitKat version 4.4
- Windows 10 Enterprise

Software Development Integrated Development Environment (IDE)

- Android Studio Build 2.1.3 Stable Release
- IntelliJ IDEA 2016.2.2

Relational Database Management Systems (RDMS)

- SQLite Database
- MySQL Database (version 5++)

Server-Side Integration

- Apache Web Server

Integration Gateways

Sinch API provided interface for flash call and SMS interceptions for verification using the follow technologies

- SMPP for SMS notifications
- GSM for voice calls

Programming Languages

- Java 8
- PHP
- JSON – JavaScript Object Notation for representing RESTful Web Services

3.4.2 Testing Tools

- Android powered mobile device running Android version 2.3 up to version 5.1
- R-Studio software for analyzing the data collected

3.4.3 Application Modules

The application implemented the following authentication components:

1. One Time Password (OTP)

A random one-time password will be generated by the application. The password will be meant to expire after defined time t and will be made of 8 alphanumeric characters. A second alternate OTP will be sent over email as an alternate communication channel (out of band communication) for security purposes. The user will be asked to read the password as displayed on the phone for voice biometric authentication or to manually enter the password on the web interface. The system will validate irrespective of the input method.

2. PIN

The system will ask the user to use a numeric password. The chosen password shall be four characters long and shall not be made of at least 3 leading similar numbers. The maximum entry for a PIN will be twice. If a PIN is wrongly entered 3 times, the user has the option of using an OTP and Ooba with voice biometric validation for resetting the PIN.

3. Device Based Soft Token

A device specific token will be generated and encrypted with a private key. The token will be used to authenticate the transaction at the background without user intervention. Where the user authentication fails based on the two authentication methods above, a new token will be generated and stored securely in a server. The token is then sent to an email for the user and accessing it will authenticate the user and allow the creation of a new PIN.

The user of the mobile banking application enters a PIN on a mobile application interface to initiate the connection to the web server application. At the web server, the PIN is authenticated and validated against the minimum requirements. A randomly created set of characters or numerical are also sent to the account owners registered a phone number. The application can be configured to allow both web OTP authentication and mobile flash call for the OTP for authentication to make the OOB authentication more secure. In the background, the phone and web server create a device specific token based session to

validate the phone IMEI. On authentication of the OTP, the user gains access to the mobile payment platform on the mobile device to carry out any mobile banking transaction.

The following is the process flow chart for the mobile based multi-factor authentication.

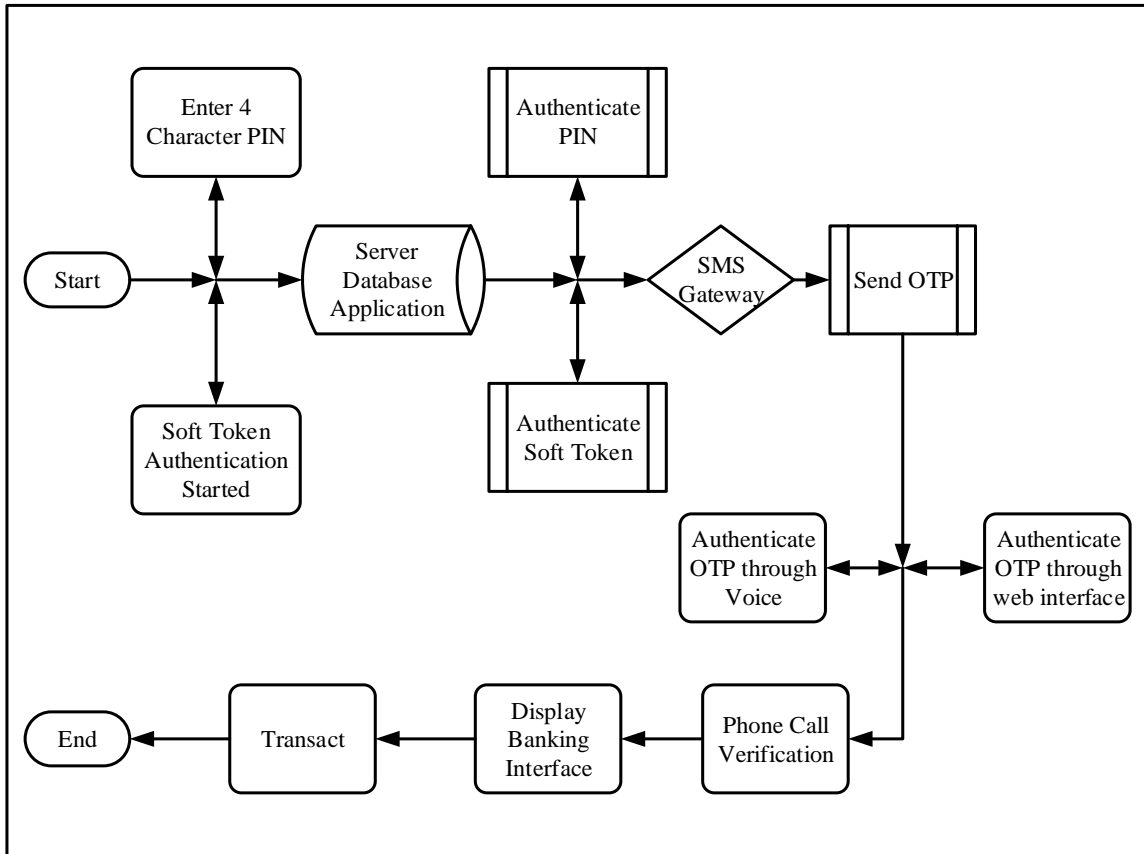


Figure 4 Multi-Factor Authentication Process Flow Chart

3.5 User Interface Design

The following are mock-up designs that were developed during the prototype phase.

Step 1: Phone Registration

A screenshot of a web form for phone registration. At the top right is a grey button labeled "Next". The main heading is "Please Enter Your Phone Number to Continue". Below this are two input fields: "Country" with the value "+254" and "Phone Number" with the value "0721966366". A blue underline is present under the "Phone Number" field. At the bottom, there is a note: "Phone Calling Carrier charges may apply".

Figure 5 Step 1: Phone Registration

Step 2: Phone Verification



Figure 6 Step 2: Phone Verification

Step 3: PIN Registration

A screenshot of a PIN registration form. At the top right is a grey button labeled "Next". The main heading is "Please chose a 4 Number PIN". Below this are two input fields: "Enter PIN" and "Confirm". Both fields have a green "Text" label above them. A red error message is displayed at the bottom: "PIN cannot have 3 leading/similar number such as 2221/3451."

Figure 6 Step 3: PIN Registration

Step 4: Choosing Authentication Scheme

Next

Chose your default Authentication Type

- PIN and OTP
- PIN and Phone Call
- OTP and Phone Call


Figure 7 Step 4: Choosing Authentication Scheme

Step 5: Authenticating Based on Chosen Scheme Type

Authentication **Next**

Enter PIN

Enter OTP



Voice Call Wait as we verify your mobile number

Figure 8 Step 5: Authenticating Based on Chosen Scheme Type

Step 6: Mobile Banking Deposit

Deposit **Withdraw**

Balance \$ 5,289

Amount

Deposit

Figure 9 Step 6: Mobile Banking Deposit

Step 7: Mobile Banking Withdrawal

Figure 10 Step 7: Mobile Banking Withdrawal

3.6 Database Design

a) SQLite

a. Mobile Data Main Table

Column Name	Type	Length	Primary Key
Transaction_ID	INT	100	Yes
Session_Key	VARCHAR	256	Yes

Table 1 Mobile Data Main Table

b) MySQL

a. Mobile Data Offline Table

Column Name	Type	Length	Primary Key
Transaction_ID	INT	100	Yes
Session_Key	VARCHAR	256	Yes
PIN	VARCHAR	100	No
OTP	VARCHAR	100	No
Authentication_Status	VARCHAR	10	No
IMEI	VARCHAR	20	Yes
MSISDN	VARCHAR	12	Yes
Money_In	INT	10	NO
Money_Out	INT	10	NO
Authentication_Type	VARCHAR	100	NO

Latency_(Delay)	LONG	100	NO
-----------------	------	-----	----

Table 2 Mobile Data Main Offline Table

b. Metrics Log Table

Column Name	Type	Length	Primary Key
Transaction_ID	INT	100	Yes
Session_Key	VARCHAR	100	Yes
Authentication_Type	VARCHAR	100	No
Session_Start	Time (Long)	10	No
Session_End	Time (Long)	10	No
Latency	LONG	100	No
Authentication_Status	VARCHAR	10	No

Table 3 Metrics Log Table

Database Schema

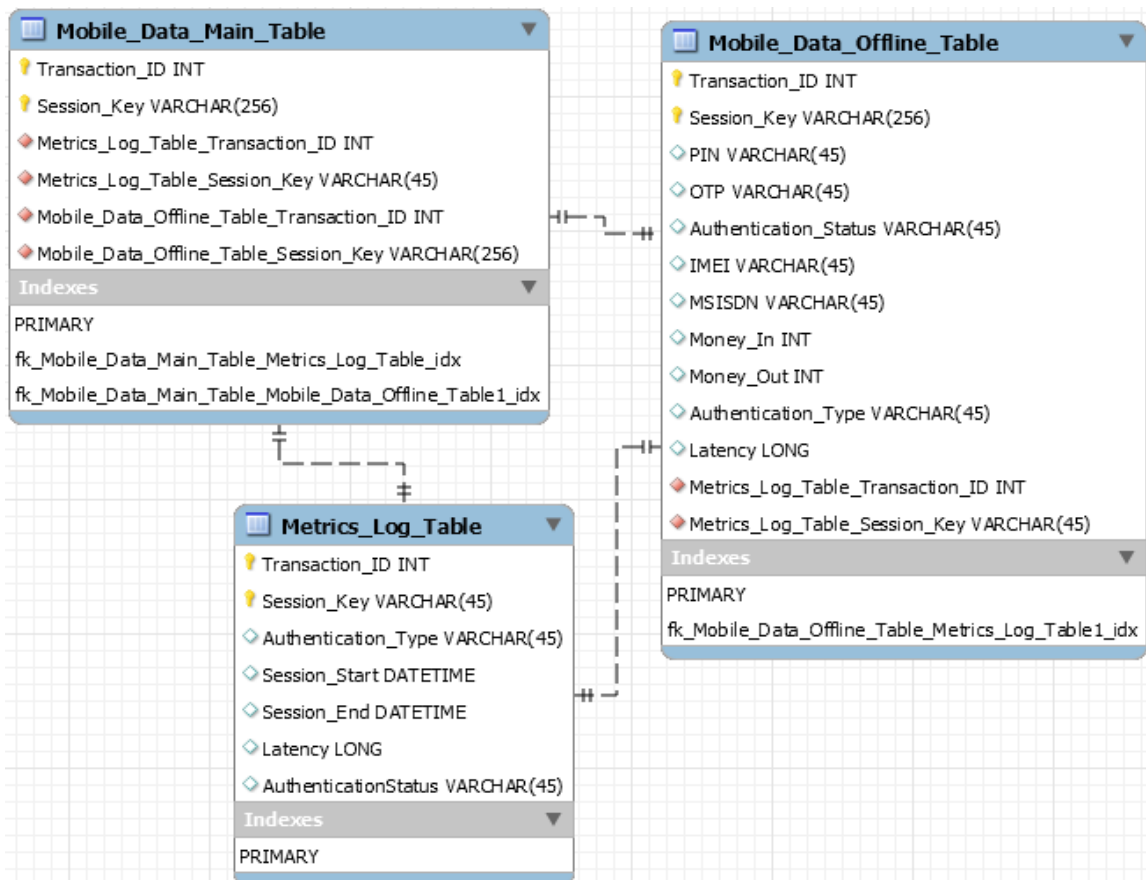


Figure 11 Database Schema Showing Table Relationships

3.7 Process Flow

1. User installs and launches application on phone
2. User registers to use application by entering four numerical PIN number
3. Application creates user profile based on PIN, Phone number and device ID (token created based on device ID tied to IMEI and MSISDN)
4. User chooses default authentication method (by default, IMEI and MSISDN authentication will run on the background and change of SIM Card will be detected and verification voided and new registration required)–
 - a. PIN and OTP (over SMS),
 - b. PIN and Phone verification (through calling),
 - c. OTP (over SMS) and Phone Verification
5. User launches application for banking transaction simulation to make payment/withdrawal
6. The user is authenticated/not authenticated and the transaction is successful/fails.

Sinch API verification flow chart for voice call, flash call and SMS

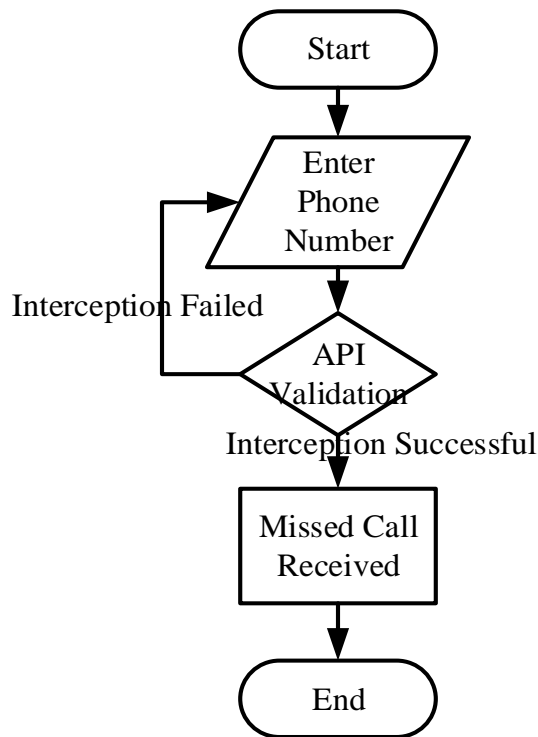


Figure 12 Sinch API Validation Process

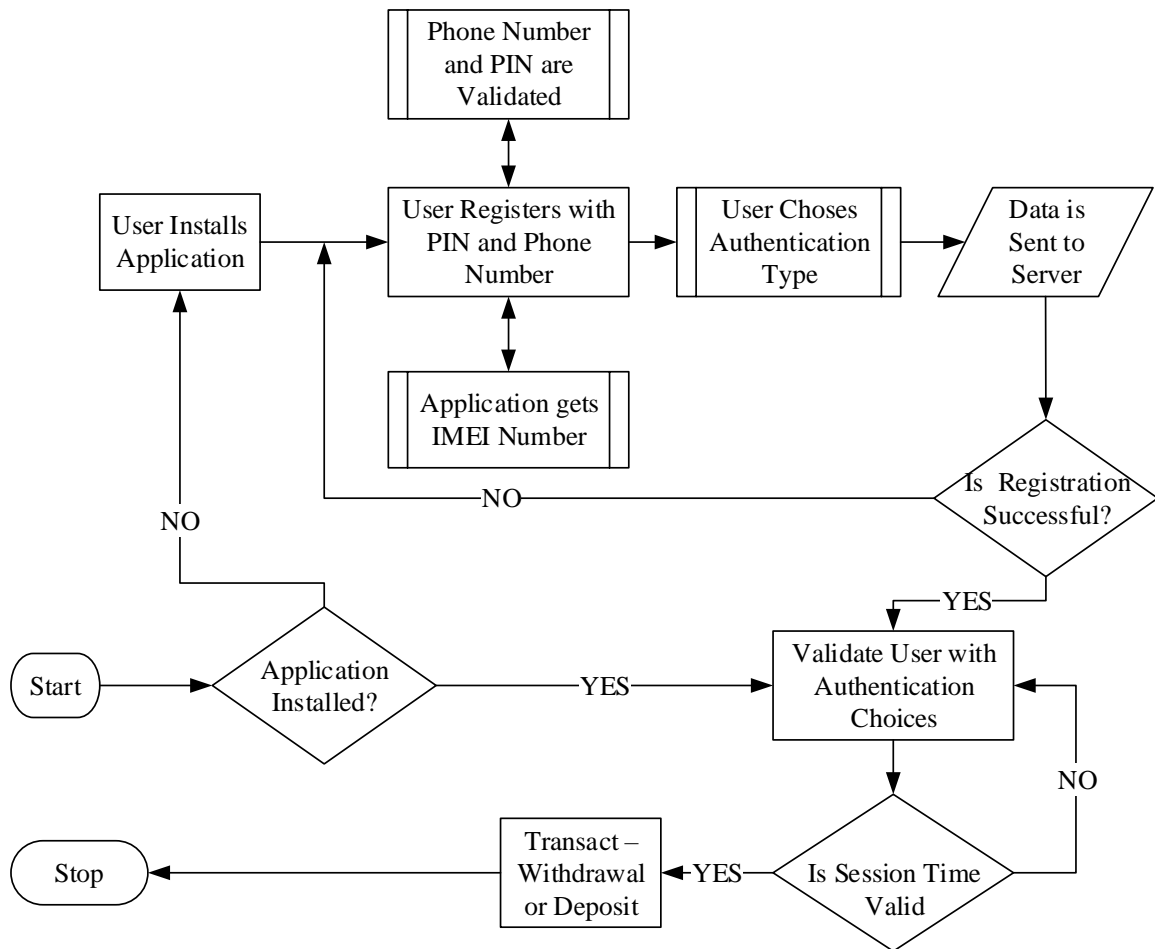


Figure 13 Process Flow Diagram - Entity Relationship Diagram

Time Delay Logging

```

public void onAuthenticationVerified() {
    //hideProgressDialog();
    mProgressDialog.dismiss();
    // Log.e("PREFERENCE:", pref);
    success = "SUCCESS";
    endTime = System.currentTimeMillis();
    Log.e("END TIME:", endTime.toString());
    saveToMySQL();
    logData(phoneNumber, deviceID, pin, pref, success, startTime, endTime);
    transact();
}
  
```

Saving Time Delays to the Database

```
private void saveToMySQL() {
    mProgressDialog.setMessage("Loading.....");
    mProgressDialog.setCancelable(false);
    StringRequest postRequest = new StringRequest(Request.Method.POST,
Global.AUTHENTICATION_URL, new Response
    .Listener<String>() {
        @Override
        public void onResponse(String response) {

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            mProgressDialog.dismiss();
            Toast.makeText(getActivity(), "Failed to insert", Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<String, String>();
            params.put("phoneNumber", phoneNumber);
            params.put("deviceID", deviceID);
            params.put("pin", pin);
            params.put("encryptedPIN", getEncryptedPIN());
            params.put("pref", pref);
            params.put("success", success);
            params.put("startTime", formattedTime(startTime));
            params.put("endTime", formattedTime(endTime));

            return params;
        }
    };
}
```

```

// Adding request to request queue
MyApplication.getInstance().addToReqQueue(postRequest);
}

```

Random Transaction ID

```

private String transactionID() {
    int len = 10;
    String AB = "123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    SecureRandom rnd = new SecureRandom();

    StringBuilder sb = new StringBuilder(len);
    for (int i = 0; i < len; i++)
        sb.append(AB.charAt(rnd.nextInt(AB.length())));
    return sb.toString().toUpperCase();
}

```

Encryption and Decryption of PIN and Device ID

```

public String encrypt(String plainText) throws Exception {

    //get salt
    salt = generateSalt();
    byte[] saltBytes = salt.getBytes("UTF-8");

    // Derive the key
    SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
    PBEKeySpec spec = new PBEKeySpec(
        password.toCharArray(),
        saltBytes,
        pswdIterations,
        keySize
    );

    SecretKey secretKey = factory.generateSecret(spec);
    SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");

    //encrypt the message

```

```

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, secret);
AlgorithmParameters params = cipher.getParameters();
ivBytes = params.getParameterSpec(IvParameterSpec.class).getIV();
byte[] encryptedTextBytes = cipher.doFinal(plainText.getBytes("UTF-8"));
return Base64.encodeToString(encryptedTextBytes, 1);
//return Base64().encodeAsString(encryptedTextBytes);
}

public String decrypt(String encryptedText) throws Exception {

    byte[] saltBytes = salt.getBytes("UTF-8");
    byte[] encryptedTextBytes = Base64.decode(encryptedText, 2);

    // Derive the key
    SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
    PBEKeySpec spec = new PBEKeySpec(
        password.toCharArray(),
        saltBytes,
        pswIterations,
        keySize
    );

    SecretKey secretKey = factory.generateSecret(spec);
    SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");

    // Decrypt the message
    Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
    cipher.init(Cipher.DECRYPT_MODE, secret, new IvParameterSpec(ivBytes));

    byte[] decryptedTextBytes = null;
    try {
        decryptedTextBytes = cipher.doFinal(encryptedTextBytes);
    } catch (IllegalBlockSizeException e) {
        e.printStackTrace();
    } catch (BadPaddingException e) {
        e.printStackTrace();
    }
}

```

```

    }

    return new String(decryptedTextBytes);
}

public String generateSalt() {
    SecureRandom random = new SecureRandom();
    byte bytes[] = new byte[20];
    random.nextBytes(bytes);
    String s = new String(bytes);
    return s;
}

```

PHP API Code for Authentication

```

<?
$reponse = array();

if ( !(empty($_POST['phoneNumber'])) )
{
    $phoneNumber=$_POST['phoneNumber'];
    $deviceID=$_POST['deviceID'];
    $pin=$_POST['pin'];
    $encryptedPIN=$_POST['encryptedPIN'];
    $pref=$_POST['pref'];
    $success=$_POST['success'];
    $startTime=$_POST['startTime'];
    $endTime=$_POST['endTime'];

    $result = mysql_query("INSERT INTO
tbl_auth(id,phoneNumber,deviceID,pin,encryptedPIN,pref,success,startTime,endTime)
VALUES(',$phoneNumber','$deviceID','$pin','$encryptedPIN','$pref','$success','$startTime','$en
dTime')");

    if ($result>0){

```

```

    $response["successful"] = 1;
}
else{
    $response["successful"] = 0;
}
echo json_encode($response);
}
?>

```

3.8 Actual Application User Interface

The following are screen shots of the actual application while in use using an android 5.0.1 mobile device.

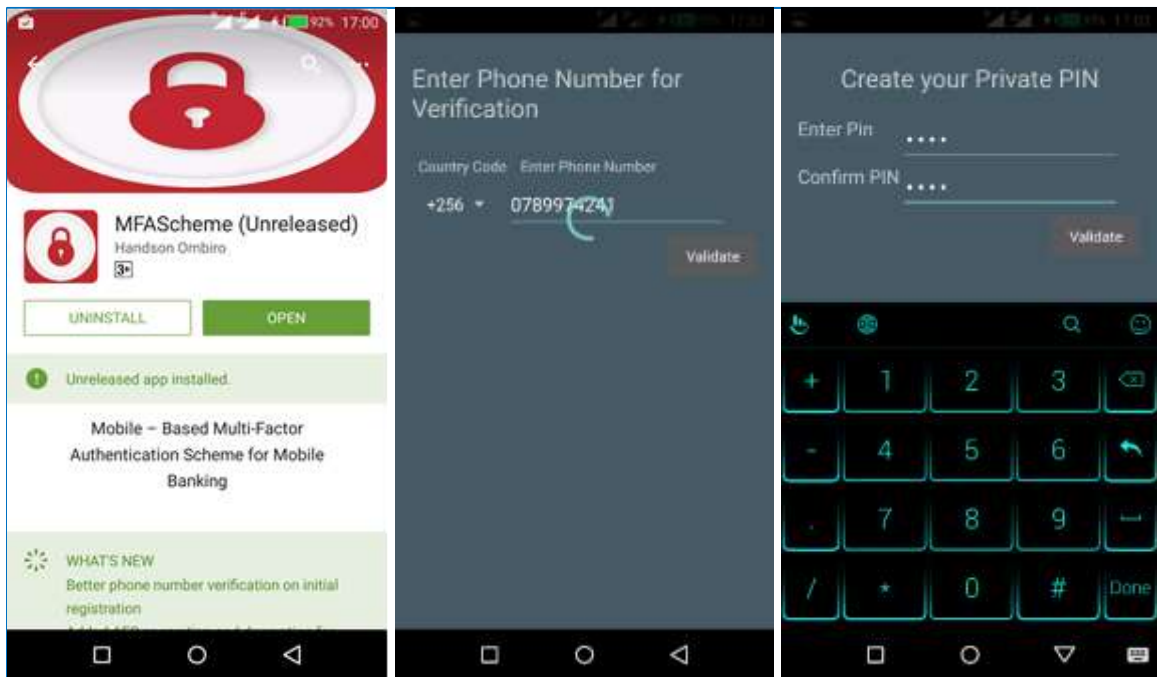


Figure 14 Application Download, Phone and PIN Registration

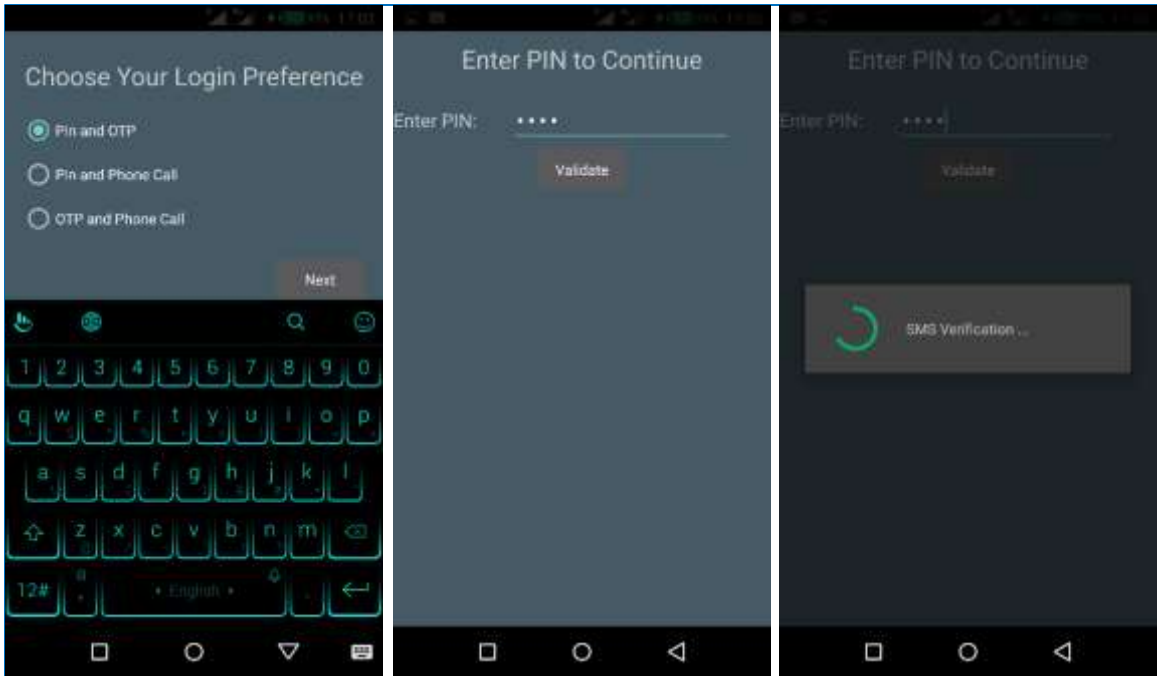


Figure 15 Login Preference, PIN, and OTP Authentication

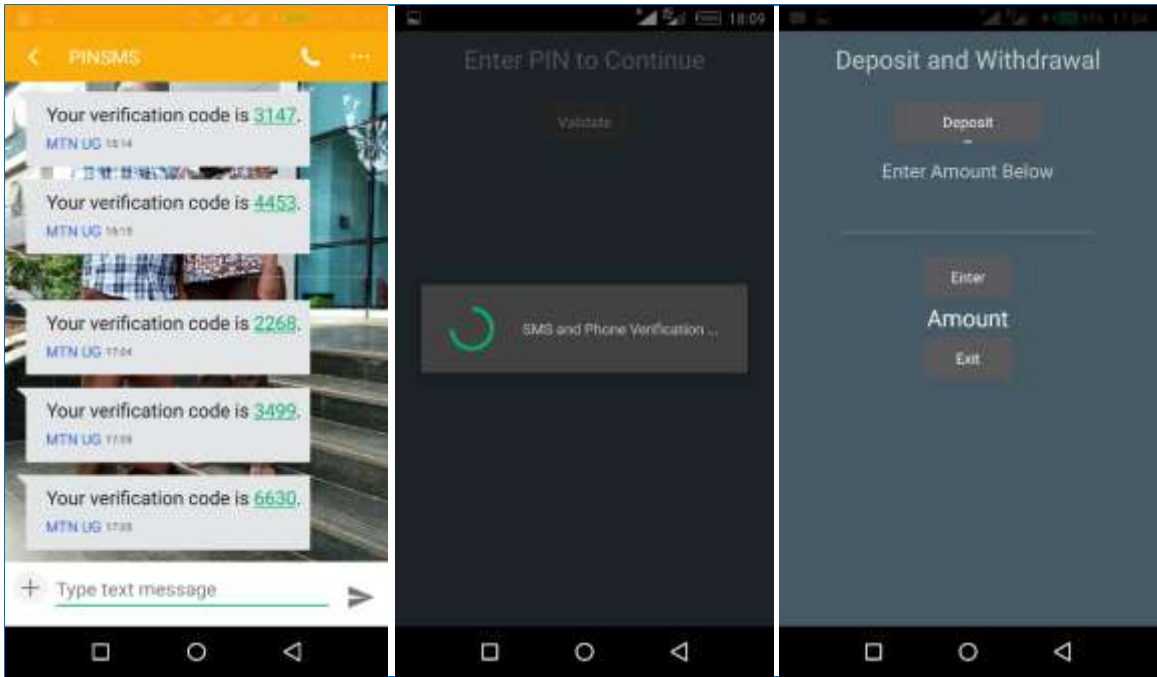


Figure 16 OTP Logs, OTP and Phone Call Authentication and Banking Interface

3.9 System Performance Results

The application performance results were based on time delays for chosen authentication type for each transaction from the time the authentication starts to the time the transaction is closed. The time delays are based on the following three scenarios.

- a) Type of authentication method chosen
- b) Time delay (latency) based on session start and session end for each complete authentication success/fail
- c) Authentication Status - fail or success

These data is to be used in measuring and analyzing the performance of the authentication scheme using the Kernel Density Estimation.

Chapter Four: Research Results and Analysis

4.1 Introduction

This chapter is a recording of the results of the different authentication tests.

4.2 Testing Scripts

Authentication	Start Time	End Time	Pass/Fail
PIN and OTP	The timestamp when PIN and OTP Authentication is initiated	The timestamp when PIN and OTP authentication is completed	Successful or failed authentication
PIN and Phone Verification	The timestamp when PIN and voice call/flash call is initiated	The timestamp when PIN and voice call/flash call authentication is completed	Successful or failed authentication
OTP and Phone verification	The timestamp when OTP and voice call/flash call is initiated	The timestamp when PIN and voice call/flash call authentication is completed	Successful or failed authentication

Table 4 Test Scripts Table

The choice of authentication was also recorded for every instance of the transaction process. Fail instances were logged based on the cause of failure. The causes recorded are:

- a) Incorrect phone number provided
- b) Wrong phone number provided
- c) Sinch API service error
- d) System errors such as disabled network or unreachable network

The following SQL syntax was used to collect time delays and authentication type:

```
SELECT * FROM `tbl_auth` WHERE `pref` = 'PIN and OTP' AND `encryptedPIN` <> 'NULL' AND `success` = 'SUCCESS'
```

```
SELECT * FROM `tbl_auth` WHERE `pref` = 'PIN and Phone Call' AND `encryptedPIN` <> 'NULL' AND `success` = 'SUCCESS'
```

SELECT * FROM `tbl_auth` WHERE `pref` = 'OTP and Phone Call' AND `encryptedPIN` <> 'NULL' AND `success` = 'SUCCESS'

4.3 PIN and OTP Authentication Report

PIN and OTP Authentication Pass Results

NO.	SUCCESS	STARTTIME	ENDTIME	DELAY IN SECONDS
24	SUCCESS	11:07:43 PM	11:08:05 PM	22.0
25	SUCCESS	11:23:48 PM	11:23:59 PM	11.0
26	SUCCESS	8:51:48 AM	8:51:59 AM	11.0
35	SUCCESS	12:20:44 PM	12:20:54 PM	10.0
42	SUCCESS	1:08:31 PM	1:08:45 PM	14.0
44	SUCCESS	9:36:51 PM	9:37:06 PM	15.0
51	SUCCESS	12:53:43 PM	12:53:49 PM	6.0
76	SUCCESS	4:18:07 PM	4:18:33 PM	26.0
77	SUCCESS	4:19:09 PM	4:19:39 PM	30.0
78	SUCCESS	6:28:17 PM	6:28:22 PM	5.0
79	SUCCESS	9:19:27 PM	9:19:34 PM	7.0
80	SUCCESS	10:04:17 PM	10:04:36 PM	19.0
89	SUCCESS	7:44:23 AM	7:44:47 AM	24.0
90	SUCCESS	7:47:34 AM	7:47:40 AM	6.0
91	SUCCESS	7:55:54 AM	7:56:00 AM	6.0
122	SUCCESS	3:15:08 PM	3:15:26 PM	18.0
123	SUCCESS	3:15:41 PM	3:15:48 PM	7.0
125	SUCCESS	3:19:53 PM	3:20:31 PM	38.0
126	SUCCESS	6:58:20 PM	6:58:34 PM	14.0
127	SUCCESS	6:58:20 PM	6:58:34 PM	14.0
128	SUCCESS	6:59:17 PM	6:59:31 PM	14.0
129	SUCCESS	6:59:17 PM	6:59:31 PM	14.0
130	SUCCESS	11:56:58 AM	11:57:06 AM	8.0

Table 5 PIN and OTP Authentication Pass Results

PIN and OTP Authentication Pass Kernel Density Plot

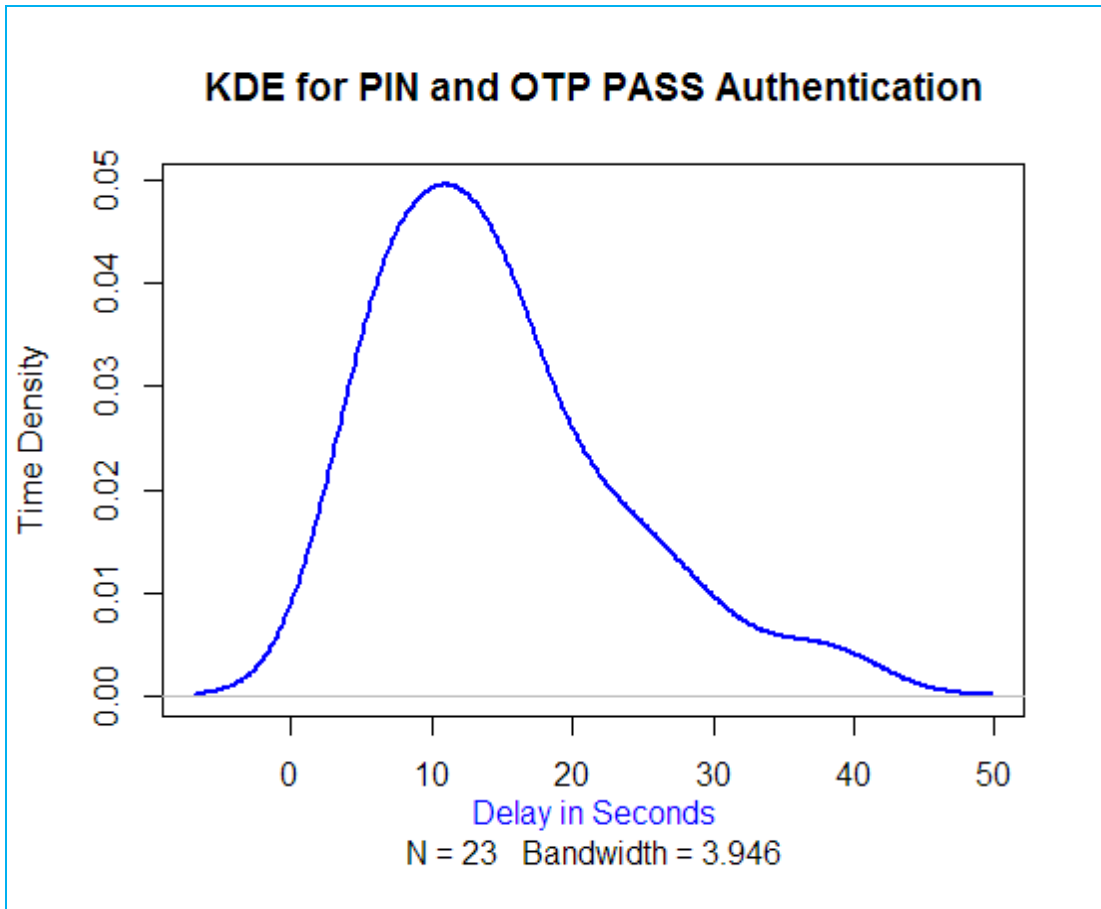


Figure 17 PIN and OTP Authentication Pass Kernel Density Plot

PIN and OTP Authentication Report Interpretation

The KDE peaks at approximately 13 seconds at 0.05. The graph illustrates a normally (center) skewed distribution. The worst authentication delay time is approximately 13 seconds at a time density of 0.05. This is a normally distributed curve but with low time density and shorter time delay.

4.4 PIN and Phone Authentication Report

PIN and Phone Authentication Pass Results

NO.	SUCCESS	STARTTIME	ENDTIME	DELAY IN SECONDS
12	SUCCESS	12:28:34 PM	12:28:46 PM	12.0
19	SUCCESS	9:45:52 PM	9:46:06 PM	14.0
20	SUCCESS	10:00:01 PM	10:00:16 PM	15.0

22	SUCCESS	10:42:01 PM	10:42:51 PM	50.0
23	SUCCESS	11:00:50 PM	11:01:05 PM	15.0
27	SUCCESS	11:32:24 AM	11:32:38 AM	14.0
28	SUCCESS	11:41:15 AM	11:41:32 AM	17.0
29	SUCCESS	11:44:42 AM	11:44:59 AM	17.0
30	SUCCESS	11:56:13 AM	11:56:25 AM	12.0
32	SUCCESS	12:07:36 PM	12:07:49 PM	13.0
33	SUCCESS	12:14:25 PM	12:14:39 PM	14.0
34	SUCCESS	12:15:23 PM	12:15:35 PM	12.0
37	SUCCESS	12:40:45 PM	12:40:57 PM	12.0
38	SUCCESS	12:50:13 PM	12:50:30 PM	17.0
39	SUCCESS	12:51:12 PM	12:51:28 PM	16.0
40	SUCCESS	1:00:02 PM	1:00:18 PM	16.0
41	SUCCESS	1:05:16 PM	1:05:47 PM	31.0
43	SUCCESS	1:12:19 PM	1:12:39 PM	20.0
45	SUCCESS	9:47:56 PM	9:48:24 PM	28.0
50	SUCCESS	12:19:28 PM	12:20:09 PM	41.0
52	SUCCESS	12:58:57 PM	12:59:15 PM	18.0
65	SUCCESS	2:41:28 PM	2:41:46 PM	18.0
67	SUCCESS	3:00:06 PM	3:00:52 PM	46.0
69	SUCCESS	3:31:55 PM	3:32:08 PM	13.0
70	SUCCESS	3:32:35 PM	3:33:46 PM	71.0
81	SUCCESS	10:08:16 PM	10:08:29 PM	13.0
82	SUCCESS	10:08:50 PM	10:09:10 PM	20.0
83	SUCCESS	10:08:50 PM	10:09:10 PM	20.0
84	SUCCESS	10:10:15 PM	10:10:28 PM	13.0
96	SUCCESS	9:28:01 AM	9:28:20 AM	19.0
98	SUCCESS	9:29:48 AM	9:29:59 AM	11.0
99	SUCCESS	9:36:16 AM	9:36:27 AM	11.0
100	SUCCESS	9:37:30 AM	9:37:44 AM	14.0

Table 6 PIN and Phone Authentication Pass Results

PIN and Phone Authentication Pass Kernel Density Plot

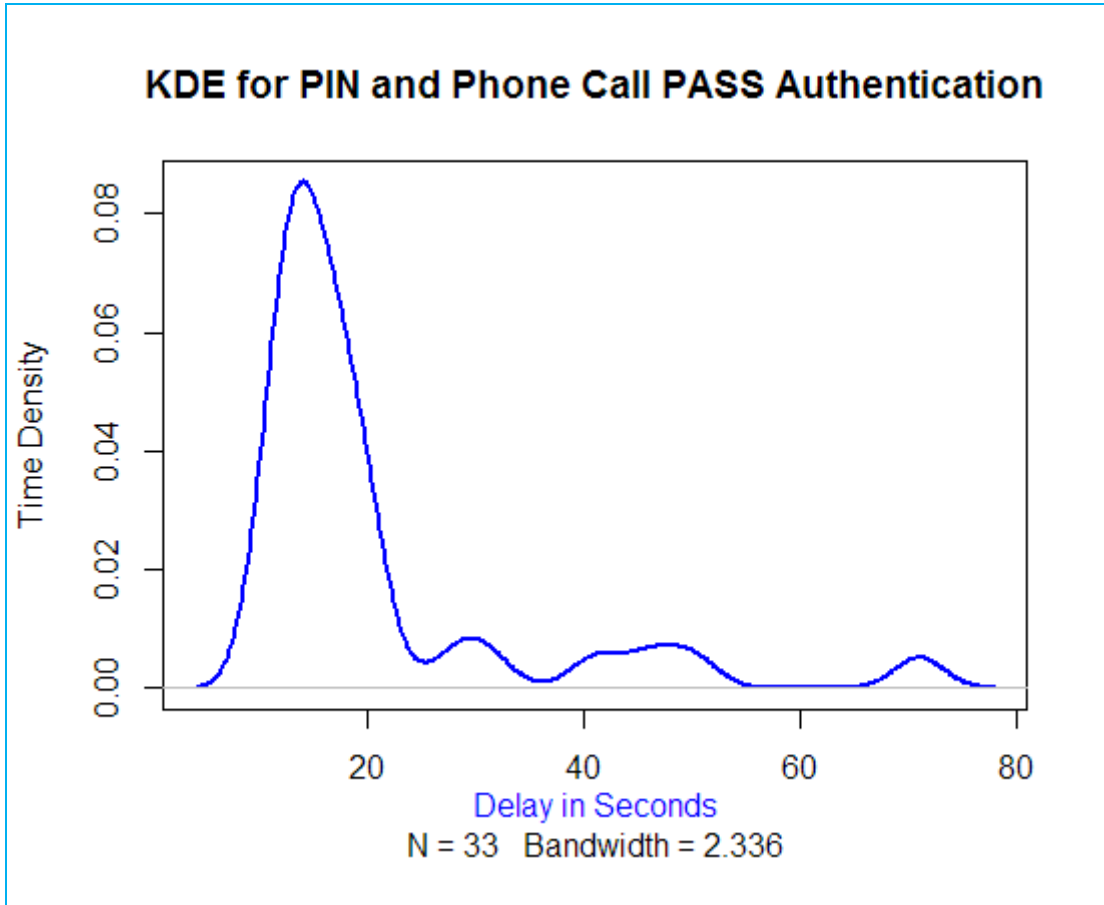


Figure 18 PIN and Phone Authentication Pass Kernel Density Plot

PIN and Phone Authentication Report Interpretation

The KDE peaks at approximately 15 seconds at 0.08. The graph illustrates a positively (right) skewed distribution. The worst authentication delay time is approximately 15 seconds. This shows a positive Gaussian distribution with a shorter time density of 0.008 and shorter time delay.

4.5 OTP and Phone Authentication Report

OTP and Phone Authentication Pass Results

NO.	SUCCESS	STARTTIME	ENDTIME	DELAY IN SECONDS
11	SUCCESS	12:20:37 PM	12:20:50 PM	13.0
13	SUCCESS	12:31:11 PM	12:31:39 PM	28.0
14	SUCCESS	12:38:36 PM	12:40:10 PM	94.0

15	SUCCESS	12:42:39 PM	12:43:04 PM	25.0
16	SUCCESS	12:52:09 PM	12:52:38 PM	29.0
17	SUCCESS	1:09:31 PM	1:09:59 PM	28.0
18	SUCCESS	1:11:49 PM	1:12:27 PM	38.0
21	SUCCESS	10:19:39 PM	10:20:28 PM	49.0
36	SUCCESS	12:27:32 PM	12:28:23 PM	51.0
46	SUCCESS	11:44:26 AM	11:44:55 AM	29.0
47	SUCCESS	11:54:23 AM	11:56:21 AM	118.0
48	SUCCESS	11:54:23 AM	11:56:21 AM	118.0
49	SUCCESS	11:54:51 AM	11:55:05 AM	14.0
59	SUCCESS	1:51:21 PM	1:52:01 PM	40.0
60	SUCCESS	2:19:41 PM	2:20:05 PM	24.0
63	SUCCESS	2:30:08 PM	2:30:54 PM	46.0
64	SUCCESS	2:34:08 PM	2:34:27 PM	19.0
72	SUCCESS	3:37:16 PM	3:38:50 PM	94.0
73	SUCCESS	4:05:20 PM	4:06:30 PM	70.0
74	SUCCESS	4:10:14 PM	4:10:52 PM	38.0
75	SUCCESS	4:14:14 PM	4:14:48 PM	34.0
85	SUCCESS	10:25:13 PM	10:25:45 PM	32.0
87	SUCCESS	10:26:15 PM	10:26:59 PM	44.0
88	SUCCESS	10:27:18 PM	10:27:35 PM	17.0
92	SUCCESS	8:49:15 AM	8:49:51 AM	36.0
93	SUCCESS	8:50:33 AM	8:50:50 AM	17.0
94	SUCCESS	8:51:19 AM	8:51:35 AM	16.0
95	SUCCESS	9:25:04 AM	9:25:20 AM	16.0
103	SUCCESS	12:03:03 PM	12:03:24 PM	21.0
109	SUCCESS	1:55:37 PM	1:56:11 PM	34.0
110	SUCCESS	2:24:04 PM	2:24:28 PM	24.0
111	SUCCESS	2:27:05 PM	2:27:35 PM	30.0
119	SUCCESS	3:03:56 PM	3:04:07 PM	11.0
121	SUCCESS	3:09:00 PM	3:09:33 PM	33.0

Table 7 OTP and Phone Authentication Pass Results

OTP and Phone Authentication Pass Kernel Density Plot

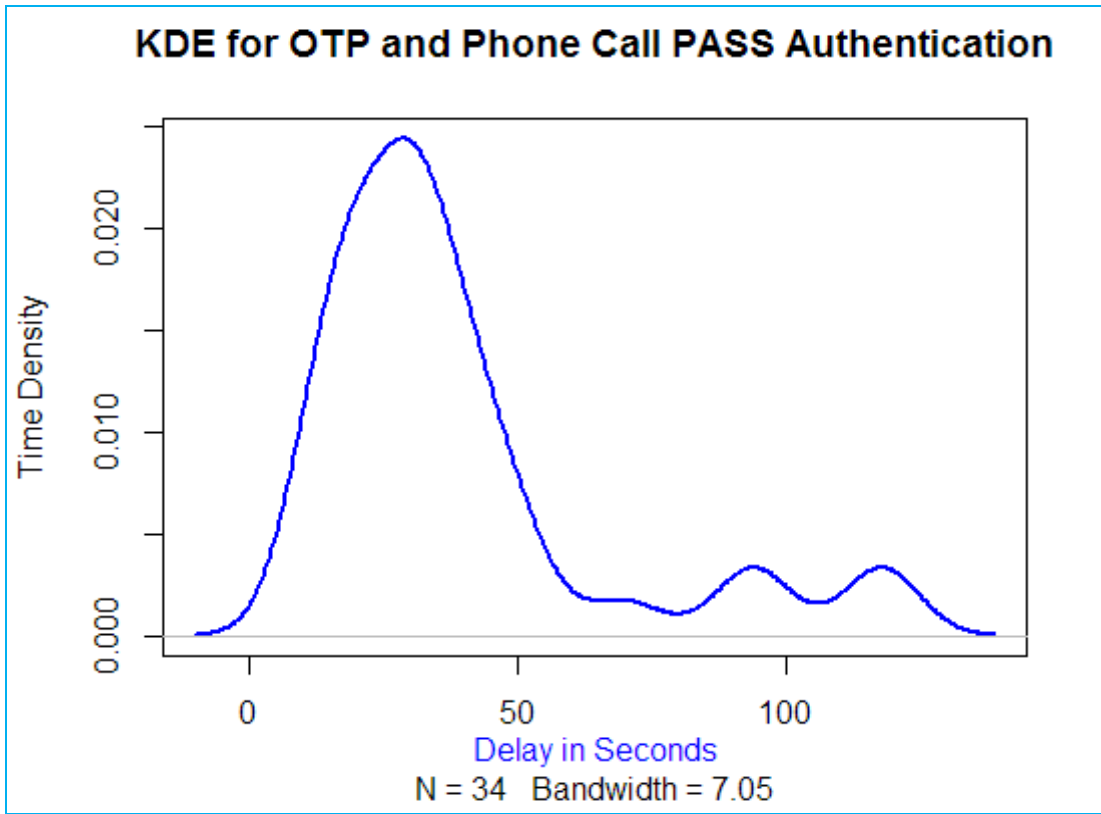


Figure 19 OTP and Phone Authentication Pass Kernel Density Plot

OTP and Phone Authentication Report Interpretation

The KDE peaks at approximately 25 seconds at 0.03. The graph illustrates a positively (right) skewed distribution. The worst authentication delay time is approximately 25 seconds (a very long delay) at a shorter time density of 0.03.

4.6 Authentication Failed Report

NO.	SUCCESS	STARTTIME	ENDTIME	DELAY IN SECONDS
2	FAIL	10:55:05 PM	10:56:02 PM	57.0
31	FAIL	12:00:07 PM	12:00:55 PM	48.0
53	FAIL	1:09:02 PM	1:09:59 PM	57.0
54	FAIL	1:13:29 PM	1:13:59 PM	30.0
55	FAIL	1:19:09 PM	1:19:57 PM	48.0
56	FAIL	1:20:15 PM	1:21:23 PM	68.0
57	FAIL	1:23:53 PM	1:24:35 PM	42.0

58	FAIL	1:26:32 PM	1:27:32 PM	60.0
61	FAIL	2:28:49 PM	2:29:55 PM	66.0
71	FAIL	3:35:18 PM	3:36:18 PM	60.0
86	FAIL	10:26:15 PM	10:26:47 PM	32.0
97	FAIL	9:28:53 AM	9:29:39 AM	46.0
101	FAIL	11:54:14 AM	11:55:15 AM	61.0
102	FAIL	11:54:14 AM	11:55:15 AM	61.0
112	FAIL	2:32:47 PM	2:33:21 PM	34.0
113	FAIL	2:43:26 PM	2:44:15 PM	49.0
115	FAIL	2:57:18 PM	2:57:46 PM	28.0
116	FAIL	2:57:57 PM	2:58:53 PM	56.0
117	FAIL	2:59:06 PM	3:00:01 PM	55.0
118	FAIL	3:01:11 PM	3:02:36 PM	85.0
120	FAIL	3:08:03 PM	3:08:34 PM	31.0

Table 8 Authentication Failed Results

Authentication Failed Kernel Density Plot

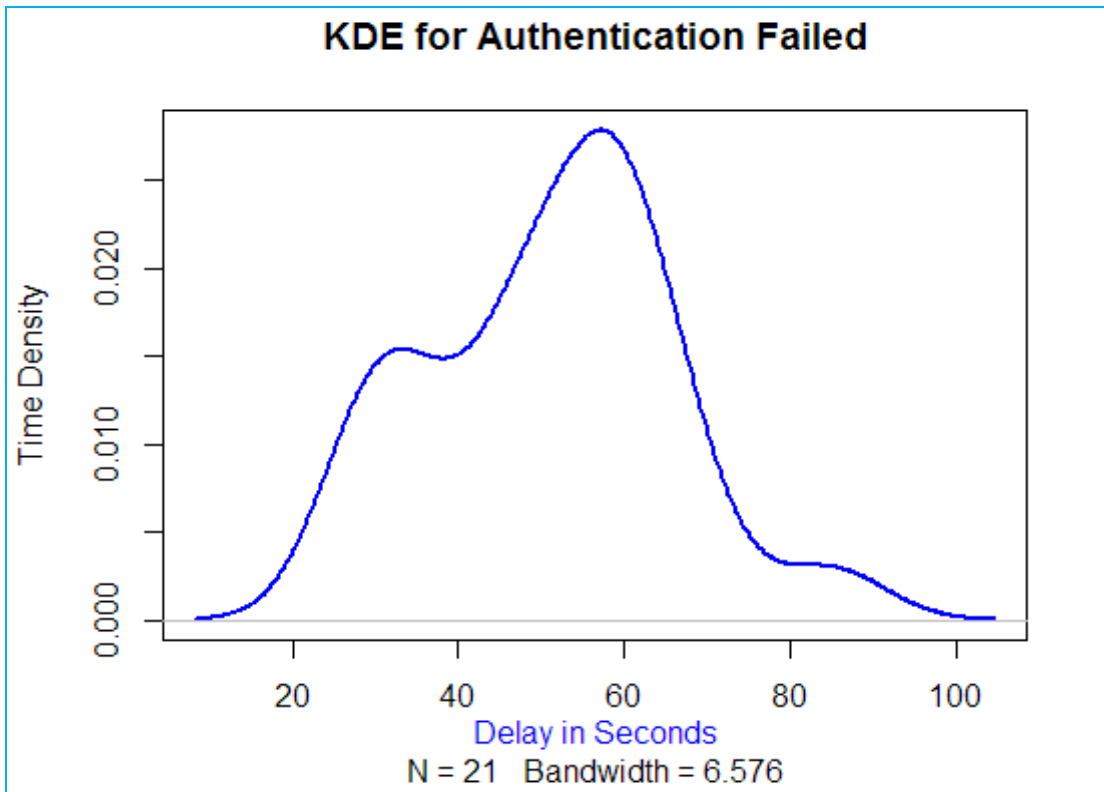


Figure 20 Authentication Failed Kernel Density Plot

Authentication Failed Report Interpretation

The KDE peaks at approximately 60 seconds at 0.03. The graph illustrates a negatively (left) skewed distribution. The worst case delay time is for authentication was 60 seconds (the longest time) with a short time density of 0.03.

4.7 Summary

Kernel Density plots were used as they are better than histograms to visualize the distribution of a variable because they eliminate the bias associated with bin size. Data analysis showed varied skew levels: centrally placed, negatively skewed and positively skewed distribution. Taking as a constant the net time delay of both device ID validation and encryption, other factors affecting time delays range from network latency to user response to authentication requests. The results independent of external factors showed that for a secure and efficient authentication was with PIN and OTP followed by PIN and phone call and lastly OTP and phone call combinations.

Chapter Five: Summary and Recommendations

5.1 Introduction

This chapter discusses the summary of the findings from analysis of research results obtained from the prototype. It also relates the findings to the literature review and research gap. Further, there is a discussion on limitations of the study, recommendations, and further academic research areas.

5.2 Summary of the Study

Transactions originating from mobile devices interface with financial services providers for processing and authentication. These transactions are prone to fraud through identity theft and eavesdropping on network channels. Two-factor authentication to validate transactions has been employed by mobile banking providers to a large extent. Research has been conducted on multi-factor authentication based on different attributes without necessarily showing which one is efficient and secure. The combination of different attributes and encrypting data in transit and at rest using AES has been used in this research to show the possibility of having a secure and time efficient multi-factor authentication.

The attributes combined were based on what the user knows (PIN) and what the user has a mobile device (device specific token) and phone number. The research showed that encryption of both stored data and data in transit using AES affects the time authentication. Background authentication of the mobile device, phone call and SMS without user intervention was used to increase time efficiency.

The authentication credentials were stored in different location: AES encrypted PIN was stored in a remote server; device ID was stored in local device using secured android shared preference file which is only accessible by the application and on a remote server for matching; Phone number for calling and sending OTP provided by an external API server.

Authentication time delays were measured and recorded in a database for the different authentication combinations: PIN, device ID and OTP; PIN, device ID and Phone Call; and, OTP, device ID and Phone Call. Time delay plots using a smoothed Kernel Density Estimation showed that combination of PIN and OTP had shorter time delays than the other schemes. OTP and Phone Call combination had the longest time delay. There are external

factors that affect time delays apart from the authentication scheme including network delay and API delays as the authentication API was from one single server.

5.3 Further Developments

- Incorporating a combined asynchronous SMS (OTP) and phone call authentication for unified and time efficient authentication. This will offer increased security as it will not be user dependent, has time bound random one-time password and is tied to the user phone number and device.
- The inclusion of what the user is (fingerprint, voice) instead of the PIN.
- Providing a single attribute authentication while the rest of the factor combinations are used in the background such as phone number, device, and biometric matching.

5.4 Limitations of the Study

The study did not implement what the user has an attribute (biometric) because the platform did not support it (Sinch API for Android supports only biometric/voice recognition for iOS Operating System).

There was no asynchronous authentication of attributes – each attribute was authenticated separately leading to session delays

5.5 Future Research areas

256 AES encryption on the phone was time-consuming, there is a need to evaluate further different encryption and storage of mobile banking credentials that will be both secure and time efficient. There was no complete elimination of human interaction in SMS based random code authentication. Further research should explore the use of automatic random codes which are automatically authenticated both by the device and by an external authentication provider. Finally, this research did not have a trust relationship with the API providers. A study should be done to find out how authentication API providers can have a trust relationship for user authentication of mobile financial transactions.

5.6 Conclusion

This research project provides a foundation for providing service oriented multi-factor authentication system that integrates more than four attributes with the little interaction of

the user and the financial service provider. Multi-factor authentication based on device and human interaction can be used to secure mobile financial transactions in a synchronous way enabling the user and provider seamless provision of security and financial services over the phone. Further security can be provided through encryption and storage of data in distributed sites reducing the risk of identity theft.

References

- Adeoye, O. S. (2012). Evaluating the performance of two-factor authentication solution in the banking sector. *International Journal of Computer Science Issues*, 9(4), 457–462.
- Agoyi, M., & Seral, D. (2011). The use of SMS encrypted message to secure automatic teller machine. *Procedia Computer Science*, 3, 1310–1314.
<https://doi.org/10.1016/j.procs.2011.01.008>
- Alliance, S. C. (2009). Security of Proximity Mobile Payments. *European Telecommunications*, (May). Retrieved from
<http://www.smartcardalliance.org/newsletter-200905-feature/>
- Aloul, F., Zahidi, S., & El-Hajj, W. (2009). Two factor authentication using mobile phones. *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, (February 2016), 641–644.
- Andrew, R. R. (2007). Out-of-Band Authentication Protects Online Financial Data. Retrieved March 1, 2016, from <http://authenticate.com/solutions/authentication-concepts/band-authentication/>
- Antal, M., & Szabó, L. Z. (2015). Biometric Authentication Based on Touchscreen Swipe Patterns. *Procedia Technology*, 22(October 2015), 862–869.
<https://doi.org/10.1016/j.protcy.2016.01.061>
- Berry, N. (2012). The Most Common PIN Numbers: Is Your Bank Account Vulnerable. Retrieved March 1, 2016, from
<http://www.theguardian.com/money/blog/2012/sep/28/debit-cards-currentaccounts>
- Bruce, S. (2014). Choosing Secure Passwords. Retrieved May 30, 2016, from
https://www.schneier.com/blog/archives/2014/03/choosing_secure_1.html

- Cha, B., Lee, S., Park, S., & Ji, G. L. Y. (2015). Design of Micro-payment to Strengthen Security by 2 Factor Authentication with Mobile & Wearable Devices, *109*, 28–32.
- Chagnaadorj, O., & Tanaka, J. (2014). Gesture input as an out-of-band channel. *Journal of Information Processing Systems*, *10*(1), 92–102.
<https://doi.org/10.3745/JIPS.2014.10.1.92>
- Communications Authority of Kenya. (2016). FIRST QUARTER SECTOR STATISTICS REPORT FOR THE FINANCIAL YEAR 2015 / 2016, *2016*(September 2015), 1–28. Retrieved from www.ca.go.ke
- Corella, F., & Lewison, K. (2012). Strong and Convenient Multi-Factor Authentication on Mobile Devices, 1–31.
- Council, F. F. I. E. (2011). Authentication in an Internet Banking Environment, *1*(703), 1–14. <https://doi.org/10.1109/ICISP.2006.3>
- Denée, C. (2014). US Mobile Payments Will Reach \$142B By 2019. Retrieved March 31, 2016, from <http://blogs.forrester.com/deneecarrington/14-11-17-us-mobile-payments-will-reach-142b-by-2019>
- Dmitrienko, A., Liebchen, C., Rossow, C., & Sadeghi, A.-R. (2014). Security analysis of mobile two-factor authentication schemes. *Intel Technology Journal*, *18*(4), 138–161. Retrieved from <http://ezproxy.library.capella.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=iuh&AN=97377858&site=ehost-live&scope=site>
- Douglas, H. (2009). Comparing Traditional Systems Analysis and Design with Agile Methodologies. Retrieved May 29, 2016, from <http://www.umsl.edu/~hugheyd/is6840/waterfall.html>

- EMC. (2011). RSA ADAPTIVE AUTHENTICATION AND OUT-OF-BAND AUTHENTICATION Combating Advanced Attacks and Protecting High Risk Transactions with Layered Authentication.
- Gaber, C., Gharout, S., Achemlal, M., Pasquet, M., & Urien, P. (2016). Security challenges of mobile money transfer services. *ResearchGate Conference Paper: March 2016*, (February). Retrieved from <https://www.researchgate.net/publication/279059997>
- Hackney, B., & Elizabeth, M. (2015). UXL Encyclopedia of Science. In *Science in Context* (2nd ed., Vol. 3). Farmington Hills, MI. Retrieved from <http://ic.galegroup.com/ic/scic/ReferenceDetailsPage/ReferenceDetailsWindow?failOverType=&query=&prodId=SCIC&windowstate=normal&contentModules=&display- query=&mode=view&displayGroupName=Reference&limiter=&currPage=&disableHighlighting=false&displayGroups=>
- Jadhav, P., Shirsat, P., Bhargude, P., & Kamble, S. (2016). Voice Pitch Based Authentication for Android Application, *6*(3), 3019–3021. <https://doi.org/10.4010/2016.703>
- Kevin, J. (2015). Emerging Fraud Risk in the Mobile Wallet Ecosystem. Retrieved March 31, 2016, from <https://www.trulioo.com/blog/2015/09/08/emerging-fraud-risk-in-the-mobile-wallet-ecosystem/>
- Leigh, L. (2013). PINs and Passwords, Part 1. Retrieved April 12, 2016, from <http://www.sleuthsayers.org/2013/08/pins-and-passwords-part-1.html>
- Margaret, R. (2015). Single-Factor Authentication (SFA). Retrieved April 3, 2016, from

- <http://searchsecurity.techtarget.com/definition/single-factor-authentication-SFA>
- Mihai, D. (2014). Comparative study on software development methodologies. *Database Systems Journal*, *V*(3), 37–56.
- Mohamed, T. S. (2014). Security of Multifactor Authentication Model to Improve Authentication Systems, *4*(6), 81–87.
- Muchai, C., Kimani, P., Kigen, M., Mwangi, M., & Shiyayo, B. (2015). Achieving Enterprise Cyber Resilience Through Situational Analysis. *Kenya Cyber Security Report 2015*. Retrieved from <http://serianu.com/downloads/KenyaCyberSecurityReport2015.pdf>
- Panse, D. (2014). Multi-factor Authentication in Cloud Computing for Data Storage Security, *4*(8), 629–634.
- Pegueros, V. (2012). Security of Mobile Banking and Payments.
- Sarhan, H., Hafez, A. A., & Safwat, A. (2015). Secure Android-based Mobile Banking Scheme, *118*(12), 21–26.
- Sasidevi, J., Sugumar, R., & Priya, P. S. (2015). New-Multi-Phase Distribution Network Intrusion Detection, *5*(3), 1277–1280.
- Statistica. (2015). Global mobile payment transaction volume from 2015 to 2019 (in billion U.S. dollars). Retrieved March 31, 2016, from <http://www.statista.com/statistics/226530/mobile-payment-transaction-volume-forecast/>
- Thales Security. (2014). Mobile Payments: Today’s Challenge. Retrieved March 1, 2016, from <https://www.thales-ecurity.com/solutions/by-technology-focus/mobile-payments>