Design, Testing and Formal Verification of Secure Mobile Payment Protocols

A thesis submitted to University of Hyderabad in partial fulfillment

for the degree of

Doctor of Philosophy

by

Sriramulu Bojjagani

Reg. No. 13MCPC13

SCHOOL OF COMPUTER AND INFORMATION SCIENCES

UNIVERSITY OF HYDERABAD

HYDERABAD -500046

Telangana

India

August, 2018

CERTIFICATE

This is to certify that the thesis entitled **" Design, Testing and Formal Verification of Secure Mobile Payment Protocols"** submitted by **Sriramulu Bojjagani** bearing **Reg. No. 13MCPC13** in partial fulfillment of the requirements for the award of **Doctor of Philosophy** in **Computer Science** is a bonafide work carried out by him under my supervision and guidance at the Center for Mobile Banking (CMB), IDRBT, Hyderabad.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Parts of this thesis have been published online in the following publications:

1. International Journal of Communication Systems, Wiley 2017 (Chapter 3)
2. Proceedings of WiMob 2015, IEEE (Chapter 3)
3. Proceedings of SIRS 2015, Springer (Chapter 5)
4. Proceedings of CIC 2017, IEEE (Chapter 5)

Further, the student has passed the following courses towards fulfillment of course-work requirement for Ph.D:

	Course Code	Name	Credits	Pass/Fail
1	BT701	Data Structures and Algorithms	4	Pass
2	BT702	Operating System and Programming	4	Pass
3	BT711	Advanced Networking	4	Pass
4	BT716	Mobile Computing and Secure Mobile Banking	4	Pass

Supervisor	Director	Dean
Prof. V. N. Sastry	Dr. A.S. Ramasastri	Prof. K. Narayana Murthy
IDRBT	IDRBT	School of Computer and
Hyderabad-500 057, India	Hyderabad-500 057, India	Information Sciences, UOH
		Hyderabad-500 046, India

DECLARATION

I, **Sriramulu Bojjagani**, hereby declare that this thesis entitled **"Design, Testing and Formal Verification of Secure Mobile Payment Protocols"** submitted by me under the guidance and supervision of **Prof. V. N. Sastry**, is a bonafide research work and is free from plagiarism. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodganga/INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.

Date:

Signature of the Student

(Sriramulu Bojjagani)

Reg. No.: 13MCPC13

//Countersigned//

Signature of the Supervisor

(Prof. V. N. Sastry)

Dedicated To My Parents

Acknowledgements

First and above of all, I praise **God Almighty**. I am very much grateful to him for his incredible love towards me, the gift of everlasting life and the grace of amazing energy to work hard.

This thesis would not have been possible without the help and support of the kind people around me, to only some of whom it is possible to give the particular mention here. Among them, first of all, I owe my greatest gratitude to my supervisor **Dr. V. N. Sastry** (Professor, IDRBT, Hyderabad) for his precious regular encouragement and timely support from the introductory level to the concluding level that enabled me to understand and implement the concepts learned. He guided me in every stage of my research and introduced me to the most challenging and exciting field of designing and testing of mobile banking applications. I wish to express my special gratitude and acknowledge for his continuing support, cooperation, and encouragement. I would also like to thank **Dr. N. P. Dhavale** (Associate Professor, IDRBT, Hyderabad) who suggested and motivated me to test various mobile bank apps during my course.

Later, I would especially like to thank my family. My parents **Late Sri. Mahalakshmaiah** and **Smt. Vimala Kumari** for their best support and encouragement at all times in pursuing my dreams. I would also like to thank my wife **Arundathi**. My children **Siri** and **Suhas Banyan**, and my sister **Naga Lakshmi** for their true love and the motivation to finish my degree with expediency. I would also like to thank my uncle **Late Y. Satyanarayana**, and my aunt **Smt. Aruna** for their support. Without such a team behind me, I doubt that I would be in this place today.

It is my privilege to thank **Prof. K. Narayana Murthy**, Dean, School of Computer and Information Sciences (SCIS), UoH, Hyderabad for his

academic support throughout research work. It is an honor for me to thank **Dr. A. S. Ramasastri (Director, IDRBT, Hyderabad)** for his suggestions throughout my research. I also thank **Mr. B. Sambamurthy**, Former Director, IDRBT for extending his cooperation at the preliminary stage of my research. I would like to extend my sincere thanks to **Dr. G. R. Gangadharan** and **Dr. V. Radha** who have provided helpful feedback as Doctoral Committee Members and have been great teachers who have prepared me to get to this place in my academic life.

I would like to express my special thanks to my research colleagues Piyush Kumar, Ritika Singla, K. Srikanth, Rahul Kuntala, and Shashi Kiran for their technical support in my research work. It is always a pleasure to express my sincere thanks to my research colleagues including Dr. M.Sandhya, Dr. D.Tiwari, Dr. D.Pradeep, Dr. Ghanashyam, Dr. Gopal Narayan Rai, Dr. Jatoth Chandra Shekar, Sravan Kumar, Anup Kumar Maurya, D. Mallikarjun Reddy, V. Manojkumar, V.T. Manu, K.K. Sitara, G. Jaya Krishna, Srinadh Swamy, Kumar Ravi, N. Satya Krishna, Ch. Charmila, U. Ravi, Praveen, G. Jaya Rao, Avinash, Mahesh and Nisanth Kartheek, for their daily interactions with me and kind support. I would like to express my thanks to my colleagues including T. Suresh, Chandu (RGUKT-APIIIT) for keeping me motivated and support throughout work. Nothing would have been possible without the prayer support of two churches namely the Layman's Evangelical Fellowship International (LEFI) and Calvary Temple, Hyderabad headed by Pastors. When I was facing troubles, unhealthy, and financial needs, I never forget the kind prayers of my church believers namely Israel Subbarao, Peter Paul, Nehemiah, Anand Jaya Prakash and others. I express my sincere and heartfelt thanks to all.

Sriramulu Bojjagani

Abstract

Designing secure mobile payment protocols is one of the most challenging and emerging fields in current technology because these protocols must satisfy security requirements and withstand various attacks, such as man-in-the-middle (MitM), replay, improper session handling etc. We systematically reviewed the published literature on mobile payments, protocols, and security infrastructure during 2000−2018. The survey has revealed trends, technologies, new patterns, and gaps in the existing literature. The scope of study encompasses three phases: stakeholders, technologies, and security in mobile payments and describes critical challenges and recommendations that will shape the focus of future research efforts. This systematic literature review (SLR) will be useful for practitioners, researchers, mobile payment application developers, security testers, third-party organizations, bankers, and consumers.

Mobile payments are classified as remote payments and proximity payments. In remote payments, we focused on short message service (SMS-based) mobile banking, and for proximity payments, we concentrated on near field communication (NFC-based) mobile payment. Earlier works involved designing these methods using Rivest-–Shamir-–Adleman (RSA) and other symmetric key approaches which involve more communication and computation overhead. Hence, we focused on designing these two approaches with elliptic curve cryptography (ECC). Further, the proposed protocol is verified for its correctness and security properties because most current protocols have not been verified using formal methods. The protocol framework is experimentally simulated and validated by formal methods using a model checking tool called automated validation of internet security protocols and applications (AVISPA) and Scyther tools and theoretically verifying with the Burrows-–Abadi-–Needham (BAN) logic.

Mobile devices have become targets for hackers and malicious users owing to the multifold increase in their capabilities and usage. Security threats are prominent in mobile payment and mobile banking applications (MBAs). We have proposed a threat model to detect various vulnerabilities rigorously. We have performed a systematic investigation of different unknown vulnerabilities, particularly in mobile banking applications, and shown how MBAs are vulnerable to MitM attacks. We observe that some MBAs use simple HTTP protocols to transfer user data without concern for security requirements. In most of these cases, MBAs receive fake or self-signed certificates thus blindly accepting all certificates as sound and valid, which leads to MitM attacks.

Contents

Acknowledgments **v**

Abstract **vii**

List of Figures **xv**

List of Tables **xix**

List of Notations **xx**

1 Introduction **1**

1.1 Mobile Payment System (MPS) 1

1.1.1 Participants of MPS 2

1.1.2 Models for MPS 5

1.1.3 Architecture for MPS 5

1.2 Technologies, Cryptography Mechanisms, and Formal Verification Methods used in the Thesis 7

1.2.1 Wireless public key infrastructure (WPKI) 7

1.2.2 Cryptography mechanisms 9

1.2.3 Formal verification methods 10

1.3 Objectives of the Thesis 16

1.4 Motivations of the Research 17

1.5 Contributions of the Thesis 18

1.6 Thesis Organization 19

2 Systematic Survey on Mobile Payments, Protocols, and Security Infrastructure 22
2.1 Research Methodology 22
2.1.1 Research tasks 24
2.1.2 Developing and validating the literature review protocol . . . 24
2.2 Mobile Banking 29
2.3 Mobile Payments 31
2.4 Classification of Mobile Payment Methods 34
2.4.1 Mobile channels 35
2.4.2 Remote payments 36
2.4.3 Proximity payments 39
2.4.4 Devices 41
2.4.5 Based on payment value 43
2.4.6 Based on timing of payment 46
2.5 Security Issues in Mobile Payments 48
2.5.1 Cryptography techniques in mobile payment system 51
2.5.2 Security threats in mobile payments 56
2.6 Analysis of SLR Results 59
2.6.1 Distribution of articles based on mobile payment stakeholders 59
2.6.2 Distribution of articles based on technologies in mobile payments 62
2.6.3 Distribution of articles based on security issues in mobile payments 65
2.7 Discussion and Future Directions 68
2.7.1 Discussion on stakeholders in mobile payments 68
2.7.2 Discussion on technology in mobile payments 69
2.7.3 Discussion on security in mobile payments 69
2.7.4 Benefits for practitioners and researchers 69
2.8 Summary 70

3 Secure End-to-End SMS-based Mobile Banking Protocol 72
3.1 Introduction 73
3.1.1 Organization of chapter 74
3.1.2 Contributions 74

3.2 Proposed scheme 75
3.2.1 Participants in SMS banking and initial conditions 75
3.2.2 Detailed protocol 77
3.3 Security Analysis of Proposed Approach 83
3.3.1 Informal security analysis of proposed approach 83
3.3.2 Formal verification of proposed protocol using AVISPA tool . 86
3.3.3 Formal security analysis of proposed approach using Scyther . 88
3.3.4 Formal security analysis of SMS using BAN logic 92
3.4 Performance analysis 94
3.4.1 Performance comparison of ECC, RSA, and other cryptography schemes 95
3.5 Implementation 97
3.5.1 Experimental setup 97
3.5.2 Message structure 98
3.5.3 Application demonstration and working environment 99
3.6 Summary 102

4 Secure End-to-End Proximity NFC-based Mobile Payment Protocol 104
4.1 Introduction 106
4.1.1 Organization of the chapter 108
4.1.2 Contributions 108
4.2 Proposed protocol 108
4.2.1 Entities in NFC payment protocol and Initial conditions . . . 109
4.2.2 Detailed protocol 109
4.3 Security Analysis of Proposed Approach 114
4.3.1 Informal security analysis of proposed approach 115
4.3.2 Formal security analysis using BAN logic 118
4.3.3 Formal security analysis of proposed protocol using AVISPA tool 123
4.4 Performance Analysis 125
4.4.1 Performance comparison 127
4.5 Implementation 129
4.5.1 Experimental setup 129

4.5.2 Message structure 129
4.5.3 Application demonstration and working environment 129
4.6 Summary 132

5 VAPTAi: Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps 134
5.1 Introduction 135
5.1.1 Organization of the chapter 136
5.1.2 Motivation and contributions 136
5.2 Proposed Threat Model 138
5.2.1 Insecure data storage (V1) 138
5.2.2 Lack of binary protections (V2) 139
5.2.3 Unintended data leakage (V3) 139
5.2.4 Malware in apps (V4) 139
5.2.5 Weak cryptography (V5) 140
5.2.6 Insufficient transport layer protection (V6) 140
5.2.7 Privilege escalation attack (V7) 142
5.3 Security Testing Framework 142
5.3.1 Static analysis of MBAs 142
5.3.2 Dynamic analysis 145
5.4 Case study 146
5.5 Results and Observations 147
5.5.1 Static analysis for Android MBAs 148
5.5.2 Dynamic analysis of Android MBAs 152
5.5.3 Static & Dynamic analysis for iOS MBAs 155
5.5.4 Compliance of MBAs to the OWASP 161
5.6 Summary 162

6 Novel Authentication Protocol for Prevention of Phishing and Pharming Attacks in Mobile Banking 164
6.1 Introduction 165
6.1.1 Organization of the chapter 166
6.1.2 Motivation and Contributions 166

6.2 Proposed Method 167
6.2.1 Proposed model for defending against phishing attacks 168
6.2.2 Pharming attack 172
6.2.3 Adversary $\mathcal{A}$ model 176
6.3 Security Analysis of the Proposed Framework 180
6.3.1 Security characteristics compared with related schemes 183
6.4 Case Study 184
6.5 Implementation 185
6.5.1 Experimental setup 186
6.6 Summary 189

7 Conclusions and Future Directions **191**
7.1 Conclusions 191
7.1.1 End-to-End security 192
7.1.2 Optimal consumption of resources 193
7.1.3 Formal verification 193
7.1.4 Security testing 193
7.2 Future Directions 193

List of Publications **195**

References **197**

Appendix **249**

A Formal Verification of Secure Mobile Payment Protocols **250**
A.1 SMS: HLPSL specification 250
A.2 SMS: SPDL specification 259
A.3 NFC: HLPSL specification 262
A.4 HLPSL specification for security against phishing attack 274

B Elliptic Curve Cryptography **282**
B.1 Introduction 282
B.2 Elliptic Curves Over Finite Fields 282
B.2.1 Field overview of GF (2^k) 283

B.2.2 Field overview of GF (P) 283
B.3 Domain Parameters 283
B.4 Elliptic Curve Digital Signature Algorithm (ECDSA) 284
B.4.1 Security of ECDSA 284
B.4.2 Key pair generation of ECC 285
B.4.3 Signature generation using ECDSA 285
B.4.4 Signature verification using ECDSA 285
B.5 Elliptic Curve Discrete Logarithm Problem (ECDLP) 286
B.6 Elliptic Curve Integrated Encryption Scheme (ECIES) 286
B.6.1 Scheme Setup 287
B.6.2 Key Deployment 288
B.6.3 Encryption Operation 288
B.6.4 Decryption Operation 290

C Fact Sheet for Publications **292**

D Similarity Report **296**

List of Figures

1.1 The participants in different levels of a mobile payment system 3
1.2 Architecture for MPS and their interactions during a mobile payment transaction 7
1.3 Formal verification approaches 10
1.4 AVISPA architecture 12

2.1 Tasks and research process flow 23
2.2 Survey on three RDs 25
2.3 Survey papers distribution based on an application of mobile payments and other areas 29
2.4 Year-wise distribution of surveyed papers 29
2.5 Number of mobile payment users from 2009 to 2016, by region (in millions) 32
2.6 Global mobile payment spending from 2010 to 2017, by segment (in billion U.S. dollars) 32
2.7 Mobile payments survey categorization 34
2.8 Survey of literature on mobile channel-based payments 38
2.9 Survey of literature on mobile device based payments 42
2.10 Survey of literature on mobile payment value and timing 44
2.11 Distribution of the articles based on their classification of mobile payment channels, devices, payment value and timing of payment 48
2.12 M-PKI Architecture 49
2.13 Survey of literature on security issues in mobile payments 52
2.14 Taxonomy of vulnerabilities affecting an M-payment system 57

2.15 Progress of mobile payment research year-wise w.r.t. technology, security, and stake-holders category 60
2.16 Distribution of the survey publications-based on the mobile payment stakeholders 62
2.17 Distribution of the survey publications-based on the technologies in mobile payments 64
2.18 Distribution of the survey publications-based on the security issues in mobile payments 67
2.19 Overall distribution of relevant publications according to the research domain classification 68

3.1 SMS banking architecture 76
3.2 Messages flow between participants in SMS-based mobile banking protocol 79
3.3 Sequence diagram of SMS-based mobile banking protocol 79
3.4 Static testing of our proposed SMS banking protocol 85
3.5 AVISPA OFMC back-end result 87
3.6 AVISPA SATMC back-end result 88
3.7 SMS framework using "Verification claim" 90
3.8 SMS framework using "Automatic claim" 91
3.9 Payer sends SMS for IMPS mobile payment using ECC P-224 101
3.10 Encrypted messages in out-box 101
3.11 Payee select the curve for decryption 101
3.12 Payee select a message for decryption 101

4.1 NFC operating modes 107
4.2 NFC-based mobile payment protocol architecture 110
4.3 Static testing of our proposed NFC-reader application 116
4.4 Static testing of our proposed NFC-writer application 116
4.5 AVISPA OFMC back-end result 124
4.6 AVISPA SATMC back-end result 125
4.7 Customer fill the details in WA 131
4.8 Encrypted data of customer written onto the NFC tag 131
4.9 Details entered onto the RA of merchant's reader device 132

4.10 RA encrypts the payment information and merchant specific data using AB's public key 132

5.1 App vetting process of NIST 135
5.2 Top 10 risky behaviors: (Appthority report) 137
5.3 Threat model for MBAs in context 138
5.4 Testing process of static analysis 143
5.5 Testing process of dynamic analysis 146
5.6 Vulnerabilities are identified on the attack surface 148
5.7 List of permissions used by the app 149
5.8 Static analysis of Android MBAs 151
5.9 User_ID & password is shown as plain-text 153
5.10 Balance inquiry request comparison 154
5.11 Balance inquiry response comparison 154
5.12 OTP is in plain-text, not encrypted 155
5.13 User's credentials are in plain-text in plist files 160
5.14 User's confidential data in plain-text 161

6.1 Proposed method in case of legitimate site 168
6.2 Proposed method for defending against pharming attack 172
6.3 Proposed method in case of bogus site for prevention of phishing attack 177
6.4 Proposed method in case of bogus site for prevention of pharming attack 177
6.5 AVISPA-OFMC result for phishing attack 181
6.6 AVISPA-SATMC result for phishing attack 181
6.7 AVISPA-OFMC result for pharming attack 183
6.8 AVISPA-SATMC result for pharming attack 183
6.9 Protocol steps for secure against phishing 186
6.10 Authentication server generates OTP 187
6.11 Mobile PIN input page 188
6.12 User select type of ECC curve for signature encryption 188
6.13 ABC bank home page 188
6.14 Login page for ABC mobile banking 188
6.15 Request redirected to AS 188

A.1 HLPSL for Payer. 252
A.2 HLPSL for Issuing Bank 253
A.3 HLPSL for Payment Gateway 255
A.4 HLPSL for Acquiring Bank 256
A.5 HLPSL for Payee 257
A.6 HLPSL for session, environment and goal 259
A.7 SPDL for SMS Protocol. 262
A.8 Role specification for the customer in HLPSL 264
A.9 Role specification for the merchant in HLPSL 266
A.10 Role specification for the acquiring bank in HLPSL 268
A.11 Role specification for the payment gateway in HLPSL 270
A.12 Role specification for the issuing bank in HLPSL 272
A.13 Role specification for the session, environment in HLPSL 274
A.14 Mobileuser HLPSL code for secure against phishing 276
A.15 Bank's HLPSL code for secure against phishing 277
A.16 Authentication server HLPSL code for secure against phishing 279
A.17 Session HLPSL code's for secure against phishing 281

C.1 Publication [1] 292
C.2 Publication [2] 293
C.3 Publication [3] 294
C.4 Publication [4] 295

List of Tables

1 Notations used in this thesis xxi

1.1 Mobile payment participants and expectations 4
1.2 List of entities with their keys and related certificates 8
1.3 Differences between formal approaches 15

2.1 Revised search keywords 26
2.2 Criteria for inclusion and exclusion 27
2.3 The number of collected papers in each phase using various search engines 28
2.4 Definitions of terms in comparative functionality analysis 33
2.5 Mobile payment technologies and their advantages, disadvantages, and applications 35
2.6 Differences between micro, mini, and macro payments 46
2.7 Distribution of articles based on classification of cryptography algorithms 54
2.8 Gaps identified in three phases of literature survey on mobile payment systems 66

3.1 Authentication phase 80
3.2 Transaction phase 82
3.3 Summary of resilience to the attacks 95
3.4 Comparable key sizes (in bits) for equivalent levels 96
3.5 Key generation compared with ECC and RSA 97
3.6 Comparison of key generation time with other schemes in (ms) 97
3.7 SMS framework data structure 99

3.8 Comparison of performance of SMS security payments on different server environments 102

4.1 Classification of the NFC literature 105
4.2 The goals described to analyze the proposed framework. 121
4.3 Comparison of proposed protocol (PP) and summary of resilience to various attacks 126
4.4 Comparision of RSA and ECC concerning key size, key generation, sign generation and verification 128
4.5 Performance comparison 128
4.6 Execution time for cryptographic operations using ECC 128
4.7 List of software and Hardware used for NFC implementation 130
4.8 The NFC framework data structure 130

5.1 Related work on mobile security testing 136
5.2 List of penetration testing tools, software, and devices used for our demonstration of vulnerability detection 144
5.3 Data set, (D_i indicates developer D) for an app. 147
5.4 The list of exported activities, services and broadcast receivers for D4-Bank_5 147
5.5 Static & dynamic analysis of iOS MBAs 160

6.1 Proposed method against phishing attack 171
6.2 Proposed method against pharming attack 175
6.3 Summary of resilience to the attacks 184
6.4 Data Set, (D_i indicates developer D) for an MBA 185

C.1 Fact sheet for publication [1] 292
C.2 Fact sheet for publication [2] 293
C.3 Fact sheet for publication [3] 294
C.4 Fact sheet for publication [4] 295

List of Notations

Table 1: Notations used in this thesis

Notation		Description
C, M, AB, PG, IB, AS, B	:	Customer, Merchant, Acquirer Bank, Payment Gateway, Issuing Bank, Authentication Server, Bank. Respectively
PR, PE, U	:	Payer, Payee, User. Respectively
$\mathcal{A}$	:	Adversary/Attacker/Intruder
CAN/MAN	:	Customer Account Number/Merchant Account Number
DB	:	database
UID_i	:	Unique ID code of i $\epsilon \{IB, AB\}$
C_x	:	Cipher text generated from message x
M_x	:	Message of X text entered by customer in mobile application
SC_X	:	Short code of bank X
$MMID_Y$	:	MMID of beneficiary Y, receiving fund from X
$MPIN_X$	:	MPIN of X, used for authentication of customer X
$ENC_{K_X^+}$	:	ECIES encryption function in mobile application
$DEC_{K_X^-}$	:	ECIES decryption function in mobile application
N_X	:	Nonce generated by entity X $\epsilon \{C, M, PR, IB, PG\}$
TS_X	:	Time stamp generated by entity X $\epsilon \{C, PR, IB, PG\}$
PoS	:	Point of Sale
$SIGN_X^Y(MS)$	:	Signature generated by X and verified by Y on message MS
K_X^+	:	Public key of entity X $\epsilon \{C, M, AB, U, AS\}$
K_X^-	:	Private key of entity X $\epsilon \{C, M, AB, U, AS\}$
ID_X	:	Identification of entity X
ACC_X	:	Account number of entity X $\epsilon \{C\}$
POI_X	:	Payment Order Information generated by entity X $\epsilon \{M\}$
PI	:	Payment Information.

MOB_X	:	Mobile number of X $\epsilon \{C\}$
Amt	:	Amount given by Customer
TID_X	:	Transaction ID generated by X $\epsilon \{M, C\}$
$(PI)_{K_{X,Y}}$	:	Payment Information encrypted with symmetric key and PI shared between X and Y
$hash(K, MESSAGE)$	:	Hash of key and message
$SK_{X,Y}$	:	Session Key used by X and Y
ID_X	:	Identification of X $\epsilon \{PR, PE\}$
BI	:	Bank Information.
SMS_X	:	SMS generated by entity X
BAL_X	:	Balance Amount of entity X
M	:	Mobile Device of the User
URL	:	Uniform Resource Locator of the Bank
$Link$	:	Link in the website of Bank which redirects to AS
RC_U	:	Credentials of User available with AS
AI	:	Accessed information with M
$X\|\|Y$	:	Concatenation Information of X and Y
X_{BAN}	:	X is the principles used in BAN Logic $\epsilon \{C, M, AB, PG, IB\}$
S	:	Statements like $POI_M, TID_M, \alpha, \beta$ etc.
$X \overset{MESSAGE}{\leftrightharpoons} Y$	:	$MESSAGE$ is a confidential information and it is only known to X or Y and perhaps to the trusted principals
$X_{BAN}\| \equiv S$	:	X_{BAN} believes S, or X_{BAN} believes S is true
$A\ believes \overset{O_x}{\prec} \begin{Bmatrix} B \\ C \end{Bmatrix}$	:	A believes secure and timely O_x, where x is between B and C.
$X_{BAN}\| \sim S$	:	X_{BAN} once said S. X_{BAN} sent a information containing S and it could be a new or old data.
$X_{BAN} \Rightarrow S$	:	X_{BAN} has jurisdiction over S. That is X_{BAN}'s tells truth about S should be believed
(T'_x)	:	Validity time periods in certificates. Every party believes certificates are within the validity period.
$X_{BAN} \triangleleft S$	:	X_{BAN} collects the data that containing S and it can read or repeat S
$< S >_{S1}$	:	$S1$ is a confidential and its presence gives the identity of whoever generates $< S >_{S1}$

Chapter 1

Introduction

The evolution and adoption of mobile technology have revolutionized our daily lives. It has impacted the way data can be gathered, the ability to communicate, and the means to buy and sell goods and services. These interactions occur in the virtual environment with no constraints on place or time. The rapid growth of mobile network technology and the security infrastructure framework for mobile payments has enabled people to perform m-commerce transactions anywhere at any time. M-commerce transactions involving financial services directly or indirectly are performed via a wireless telecommunication network. Mobile services in the financial sector are known as "Mobile Financial Services (MFS)" such as "Mobile Payments" and "Mobile Banking" [1, 2]. Mobile payments can be offered by an autonomous business body which need not necessarily be a bank. The MFS offered by various banks using mobile devices include mobile banking, stock market transactions, portfolio investments, personal financial management, tax payments, social media banking, etc. In this thesis, we address the issues and novel solutions involving secure mobile payments.

1.1 Mobile Payment System (MPS)

An MPS assists in performing payment transactions through mobile devices. It is beneficial to transfer funds in real-time when customers buy goods and services from merchants or retailers. Here, we discuss the participants of MPS (the stakeholders), their expectations, various models of MPS, an architecture of MPS showing various phases,

and transaction flows.

1.1.1 Participants of MPS

The participants or actors or stakeholders involved in mobile payment systems are categorized into three distinct levels by Eisenmann et al. [3] as (A) user level, (B) platform level, and (C) sponsor level, as shown in Figure 1.1.
(A) User level: In this level, there are two groups of users. Demand-side users are generally called "customers", and supply-side users, who offer goods, complements, and services are commonly called "merchants".

- Customer (C): A person who buys the goods or services from the merchant or seller using his/her mobile device with low computational capabilities, (e.g., mobile phone, personal digital assistant (PDA), smartphone, etc.) that has a built-in keyboard, display, and short-range communication link such as Wi-Fi, near field communication (NFC), BlueTooth, radio frequency identification (RFID), etc.

- Merchant (M): An entity that sells products or services to earn the profit.

(B) Platform level: Acts as a mediator between the user level and sponsor level. The platform level consists of the mobile payment channels, systems, and protocols. Examples are "unstructured supplementary service data (USSD)", "short message service (SMS)", NFC, "interactive voice response (IVR)", etc.
(C) Sponsor level: The participants involved in this level are banks (issuer bank and acquirer bank), telecommunications (telco) operators, certificate authorities, payment gateways, government organizations, etc. The responsibilities for this sponsor level are to organize secure channels between the entities and provide the platform-mediated network for mobile payment technology.

- Issuer Bank (IB): The customer's financial institution, who is responsible for the user's account and offers the electronic payment instruments to be used by a customer.

- Acquirer Bank (AB): It is a merchant's financial institution and is responsible for management of the merchant's account and for verification of the deposited payment instrument.

- Payment Gateway (PG): It acts as an intermediary between both the acquirer and issuer banks for settlement the payments between customer and merchant with the help of the bank's secure private network [4].
- Certification Authority (CA): It is an entity that issues digital certificates to certify the ownership of a public key by the named subject of the certificate. This allows others (relevant parties) to rely upon signatures or on assertions made about the private key that correspond to the certified public key. A CA acts as a trusted third party; trusted both by the subject (owner) of the certificate and by the party relying on the certificate. The format of these certificates is specified by the X.509 standard.

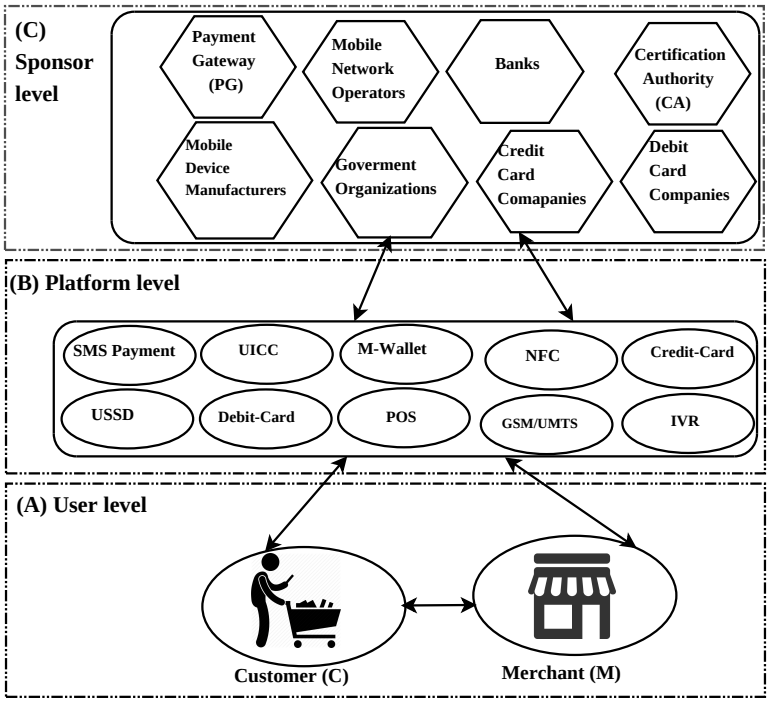


Figure 1.1: The participants in different levels of a mobile payment system

The goals and expectations of various participants are described as shown in Table 1.1.

Table 1.1: Mobile payment participants and expectations

Participants	Expectations
Customer [5, 6, 7, 8, 9, 10]	(i) Person-to-Person (P2P) transfers (ii) Pervasive use "anytime, anywhere and in any currency" (iii) Minimum or zero cost of usage (iv) Interoperability between various mobile network operators, devices and banks (v) The ability for anonymous payments (ex: cash) (vi) Payments performed in the secure environment so that trust and privacy achieved at the technical and social level (vii) Better personalized service (viii) Supports for all types of payments like mini, micro, and macro payments (ix) On-line mobile payment transaction status overview
Merchant [11]	(i) Minimum or zero new investment and operational costs (ii) Faster transaction time (iii) On-line status of the mobile payment service (iv) Mobile payment transactions performed in a highly secure environment so that merchant expects high security, trust, and privacy (v) The possibility of customizing the service (vi) Integration or simplification with existing payments systems
Banks (Issuer and Acquirer) [12, 13, 14]	(i) Better volumes in banking: more digital or card payment transactions and less cash flow transactions (ii) Integration/use of current infrastructure payments systems (iii) Mobile payment transactions performed in a highly secure environment so that banks expect high security, trust, and privacy and minimum fraud-loss (iv) Branding and customer loyalty (v) Payment applications designed and controlled by the bank (vi) New business cases
Telecom network providers [13, 15, 16, 17, 18]	(i) New income generation through increased traffic (ii) Customer loyalty increase (iii) Potential to add value to existing services (iv) Increase in "average revenue per user (ARPU)"
Mobile phone manufacturer	(i) Low time-to-market (ii) Massive market adoption with new embedded hardware or software features of the devices (iii) Multi-application capability (iv) Increase in ARPU (v) Established new relationship with application providers, or mobile network operators (MNOs), or banks
Government	(i) Revenue through taxation of m-payments (ii) Level playing regulations

1.1.2 Models for MPS

Mobile payments are becoming more popular today; there is no dominant approach among mobile payment models in the market. There are four economic models associated with different mobile payment systems [19].

1. **Bank-centric model:** In this model, banks are responsible for the control of business transactions over extended periods of time, and the bank acts as a central node. Bank types are issuing banks (owning customer accounts) and acquiring banks (owning merchant accounts), and payment gateways or clearinghouses (settle payment transactions between issuing and acquiring banks).

2. **Operator-centric model:** In this model, the mobile network operator (MNO) in a strategic role provides the connection between the customer and merchant. The MNO performs user transactions on either a pre-paid user account or post-paid account basis.

3. **Collaborative model:** There is a need for collaboration between banks and mobile network operators to provide mobile payment solutions, which is a win-win model for finance as well as for the telecommunication industries because banks have several million customers and mobile network operators also have millions of users, leading their joint success.

4. **Independent service provider (ISP) model:** In this model, a third party is responsible for user transactions, and it acts as an intermediary between operators and banks. Amazon Payments, PayPal, Authorize.net, etc. are examples of independent service providers for payment services.

These models address various solutions to cooperation and coexistence of the key players involving the actions of telco operators, banks, and possibly third parties [20].

1.1.3 Architecture for MPS

The interactions of the participants during mobile payment transactions are shown in Figure 1.2. An interaction consists of four phases to complete the mobile transaction with the merchant.

1. Registration Phase: Before performing any transaction with the merchant, the customer and merchant should register his/her mobile number with the bank. The customer can then place an order with the merchant by the following steps.
Step 1: Customer places an order on an item through a mobile wallet or any other merchant application.
2. Authentication Phase: In this phase, the bank verifies the customer's account details and AB forwards the request to a PG. The PG is one of the additional entity it acts as to settle the payments between customers' bank (i.e., IB) and merchants' bank (i.e., AB) using the banks' private network.
Step 2: The request is received by the merchant and sent to the acquiring bank for payment verification but not with the payment gateway [21, 22, 23].
Step 3: The customer also sends a payment request for the deduction of the amount from his/her account to the issuing bank.
Step 4: The acquiring bank forwards the payment request to the payment gateway.
Step 5: The PG forwards the payment request to the issuing bank.
Now, the issuing bank verifies the customer account details and also verifies the availability of sufficient funds; then the transaction amount is debited from issuing bank and credited to the merchant's acquiring bank [24],[25].
3. Transaction phase:
Step 6: If all verifications are performed by the IB, then the payment response is sent to the PG.
Step 7: The PG receives the payment response and sends to the corresponding acquiring bank.
4. Notification phase:
Step 8: If the transaction is successful or unsuccessful, the customer receives payment acknowledgment (or negative notification) from the IB.
Step 9: The merchant also receives the payment notification from the acquiring bank.
Step 10: If the payment is successful, the customer receives the goods from the merchant as shown in Figure 1.2.

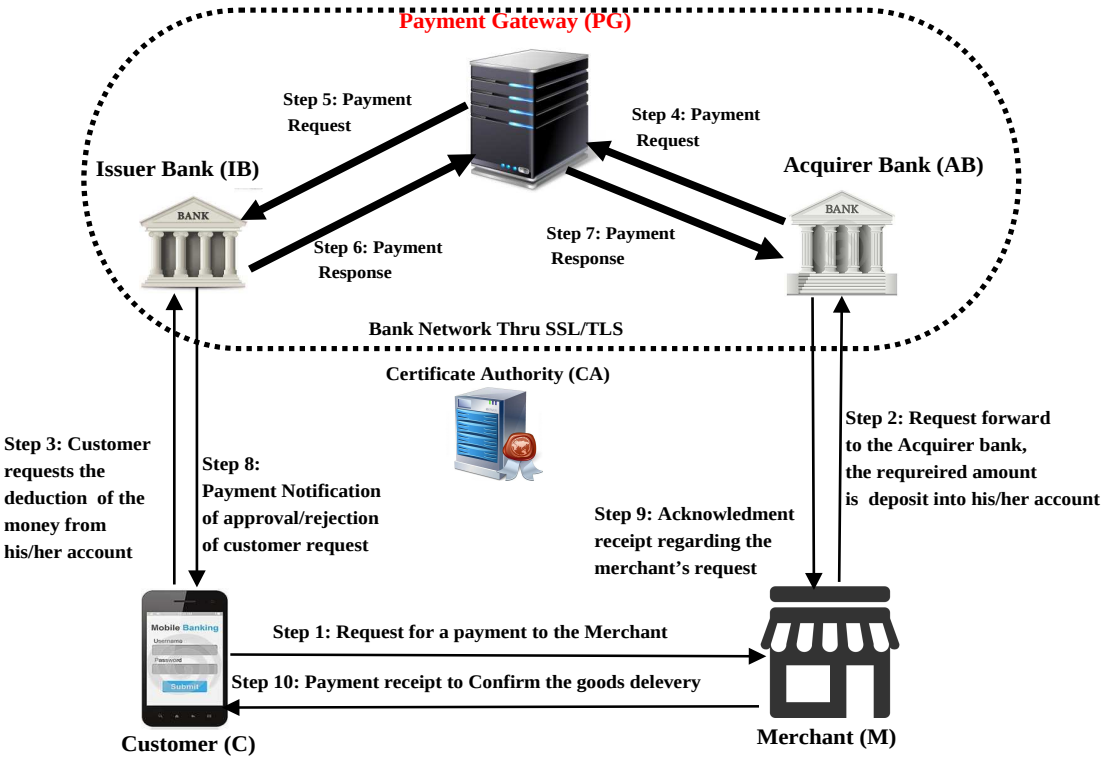


Figure 1.2: Architecture for MPS and their interactions during a mobile payment transaction

1.2 Technologies, Cryptography Mechanisms, and Formal Verification Methods used in the Thesis

1.2.1 Wireless public key infrastructure (WPKI)

The open-air transmission of data over the wireless networks makes it easier for security to be compromised. Moreover, wireless networks are also constrained by low bandwidth, more latency, less computational power, less memory size, less connection stability, and cryptography exporting restrictions. Owing to these problems, it is very difficult to apply wired PKI system in the wireless environment. A mobile phone lacks in its computing abilities such as key generation, digital signature generation and verification, certificate validation, and "certificate revocation list (CRL)" verification, and memory size of storing certificates. Many of the technologies and procedures used

for e-commerce security also apply to m-commerce security. However, special security measures for m-commerce may be required, such as WPKI and wireless transport layer security (WTLS). Wireless application protocol (WAP) is a de facto industry standard defined by participants in an "open mobile alliance (OMA)". It is a secure specification that allows users to access information instantly via handheld devices and supports most wireless networks and operating systems. WTLS is a security layer for the WAP protocol, which provides data integrity, authentication, and privacy for WAP services. WTLS is based on the widely used transport layer security (TLS) v1.0 security layer used on the Internet. Because of the nature of wireless transmissions, modifications were made to the TLS v1.0 to accommodate the special needs of wireless communications. There are two major WPKI operations in our non-repudiation protocol one is the PKI encryption and decryption, and the other is the digital signature-based evidence generation and verification.

This thesis focuses on the enhancement of application-level security thereby ensuring end-to-end security. Application level security is provided by our proposed secure mobile payment protocols. The list of entities with their key pairs and certificate are given in Table 1.2. Where $T_x^{'}$ is the validity period of the certificate issued for entity "X".

Table 1.2: List of entities with their keys and related certificates

Principal/Entities	Key pair	Certificate
Payment Gateway (PG)	(K_{PG}^{+}, K_{PG}^{-})	$Cert_{PG} = (PG, K_{PG}^{+}, T_{PG}^{'})K_{CA}^{-}$
Issuer Bank (IB)	(K_{IB}^{+}, K_{IB}^{-})	$Cert_{IB} = (IB, K_{IB}^{+}, T_{IB}^{'})K_{CA}^{-}$
Acquirer Bank (AB)	(K_{AB}^{+}, K_{AB}^{-})	$Cert_{AB} = (AB, K_{AB}^{+}, T_{AB}^{'})K_{CA}^{-}$
Customer (C)	(K_{C}^{+}, K_{C}^{-})	$Cert_{C} = (C, K_{C}^{+}, T_{C}^{'})K_{CA}^{-}$
Merchant (M)	(K_{M}^{+}, K_{M}^{-})	$Cert_{M} = (M, K_{M}^{+}, T_{M}^{'})K_{CA}^{-}$
Certifying Authority (CA)	(K_{CA}^{+}, K_{CA}^{-})	$Cert_{CA} = (CA, K_{CA}^{+}, T_{CA}^{'})K_{CA}^{-}$

1.2.2 Cryptography mechanisms

Cryptography is the practice of hiding information by changing its form into a non-obvious code. Basic services provided by cryptography are:

- The ability to store information in a form which is not easy to reveal the original information.
- The ability to exchange information between participants in a secure way.

1.2.2.1 Asymmetric cryptography

In asymmetric cryptography, two different keys are used: the public key which is used for encryption and the private key which is used for decryption and signing. Asymmetric cryptography can be used to satisfy different security requirements:

1. Public key encryption: A message encrypted with a public key of a pre-defined receiver can be decrypted only by the legitimate receiver, but nobody else. This case used for confidentiality.
2. Digital signature: A message signed with a private key can be verified by anyone who has the public key. If using the public key results in successful decryption, it proves that the encrypted document has been created by the owner of the private key. This essentially requires keeping the sender's private key secret of course.

The important algorithms used for asymmetric cryptography are (i) elliptic curve cryptography (ECC), (ii) RSA, (iii) Diffie-Hellman key exchange, etc. In this thesis, we have implemented all proposed protocols using ECC.

1.2.2.1.1 ECC: The vast majority of services that make use of public key cryptography for encryption and digital signatures use the RSA algorithm. Currently, a competing system has emerged to challenge RSA called ECC. When compared with RSA, the principal advantage of ECC is that it requires smaller bit size for the same security level. Hence, ECC reduces the processing overhead. ECC is fundamentally more difficult to explain and to process than RSA. A mathematical technique is required based on the use of a mathematical construct known as an elliptic curve. A detailed explanation of ECC is presented in the Appendix B.

1.2.3 Formal verification methods

Definition: Formal verification is the use of mathematical techniques to ensure that a design conforms to some precisely expressed notion of functional correctness. Traditionally, security protocols have been designed and analyzed heuristically.

Owing to the rapid growth of Internet usage, network users, and payment services, the development of security protocols has become necessary, and their formal verification is required. Designing security protocols is one of the more challenging issues; these protocols must not only support communication but also must satisfy the security goals. Because people tend to trust security protocols, the correctness and validation of these properties are highly desirable. Currently, many security experts are facing difficulties related to the strength of cryptographic algorithms employed; when designing any security protocol, it is mandatory to consider all possible behaviors of hypothetical attackers, including violations of the protocol rules, and any potential forgery of messages. In the protocol description, security attributes and intruder knowledge are defined formally.

There are three approaches for formal verification [26] [27] as explained in subsequent sections and as shown in Figure 1.3.

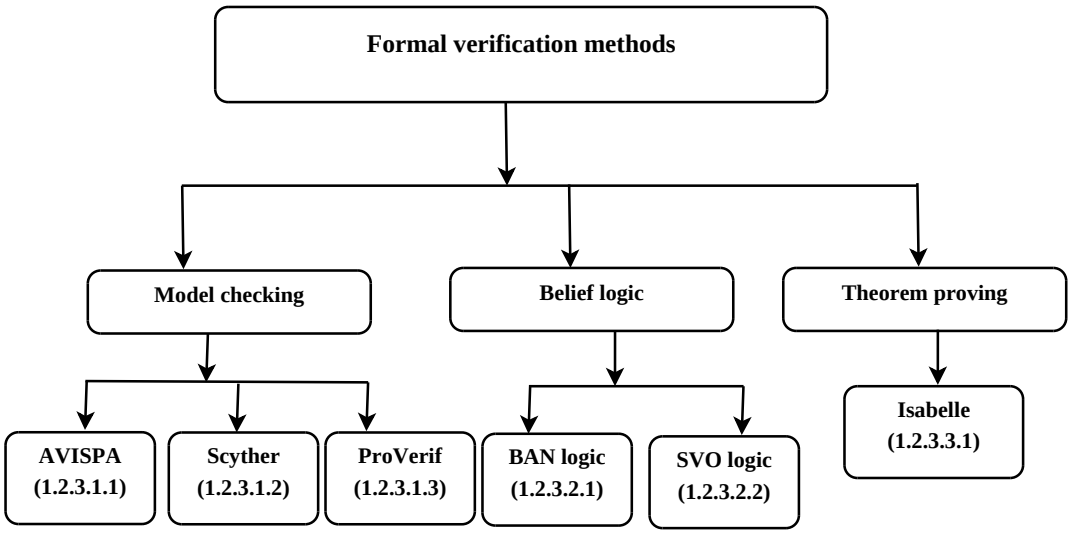


Figure 1.3: Formal verification approaches

1.2.3.1 Model checking

Definition: Model checking is an automated technique that, given a finite-state model of a system and a property stated in some appropriate logical formalism (such as temporal logic), systematically checks the validity of this property. It is used as the general approach in various applications including software, hardware, and protocol description verification. These protocol verifications are performed considering the presence of attacker, and if found to be defective, then provides a counterexample of how a protocol fails regarding the security properties.
Some of the essential points on model checking are listed below.

- To strengthen the security features of mobile payment protocols, it is essential to use tools that support the rigorous analysis of security features by either finding vulnerabilities or establishing their correctness.
- Model checking tools assist in detecting attacks. These tools should be automated, readily usable, robust, and expressive. With these properties, they can be easily integrated into the protocol development environment and application standardization process.
- Model checking is one of the most promising techniques to address security properties and has been widely used for verifying and analyzing various wireless network security protocols [28, 29, 30].

The common tools for model checking are:- (A) AVISPA[31], (B) Scyther [32], and (C) ProVerif.

1.2.3.1.1 AVISPA: The "automated validation of internet security protocols and applications (AVISPA)" [33, 34, 35] is a push-button tool used for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques. AVISPA provides a language called the " high-level protocol specification language (HLPSL)" [36] for describing security protocols and specifying their intended security properties, and provides a set of tools to formally validate them. The

structure of the AVISPA Tool is shown in Figure 1.4. An HLPSL specification is translated into an Intermediate Format (IF) using a translator called HLPSL2IF. The IF is a lower-level language than HLPSL and is read directly by the back-ends of the AVISPA Tool. The four different verification tools (back-ends) that can analyze IF specifications are:

1. "On-the-Fly Model Checker (OFMC)" [37]
2. "Constraint Logic-based Attack Searcher (CL-AtSe)"
3. "SAT-based Model-Checker (SATMC)"
4. "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)"

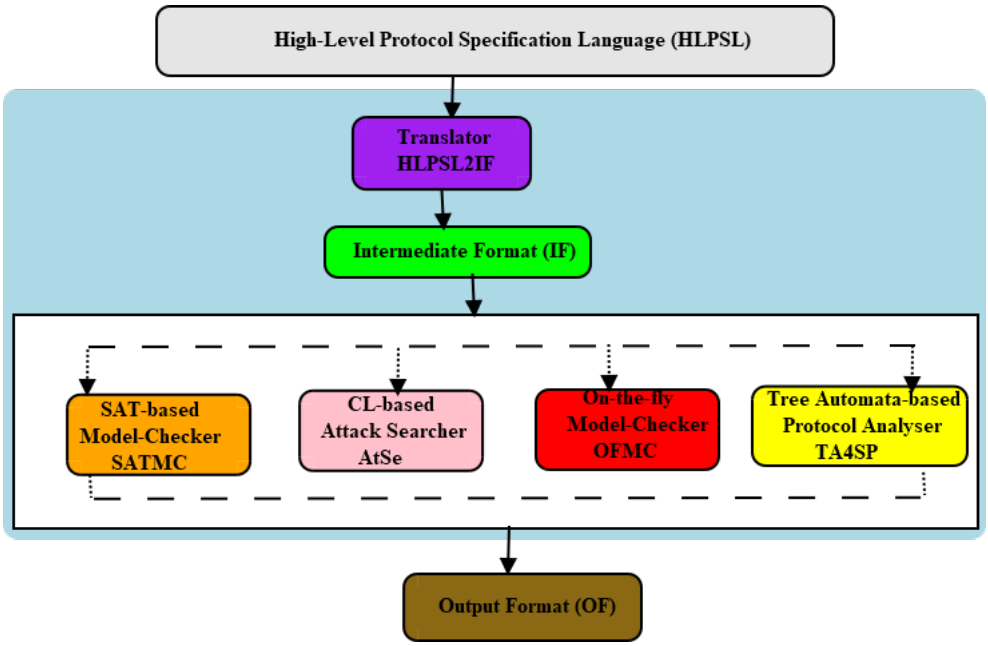


Figure 1.4: AVISPA architecture

1.2.3.1.2 Scyther: Scyther [38] is a tool for the formal analysis of security protocols under the perfect cryptography assumption, in which it is assumed that all cryptographic functions are perfect: the adversary learns nothing from an encrypted message

unless he knows the decryption key. The tool can be used to find problems that arise from the manner in which the protocol is constructed.
Scyther supports an input language known as "software protocol description language (SPDL)" [32] which is based on a C or Java-like syntax. The main purpose of the language is to describe protocols, which are defined by a set of roles. Roles, in turn, are defined by a sequence of events, most of which are events that denote the sending or receiving of terms.

1.2.3.1.3 ProVerif: ProVerif is a tool for automatically analyzing the security of cryptographic protocols. Support is provided for, but not limited to, cryptographic primitives including symmetric and asymmetric encryption, digital signatures, hash functions, bit-commitment, and non-interactive zero-knowledge proofs. ProVerif is capable of proving reachability properties, correspondence assertions, and observational equivalence. These capabilities are particularly useful to the computer security domain because they permit the analysis of secrecy and authentication properties. Moreover, emerging properties such as privacy, traceability, and verifiability can also be considered. Protocol analysis is based on an unbounded number of sessions and an unbounded message space. Moreover, the tool is capable of attack reconstruction: when a property cannot be proved, ProVerif attempts to reconstruct an execution trace that falsifies the desired property.

1.2.3.2 Belief logic

Belief logic was the first attempt to verify network security protocols, and was developed by "Mike Burrows, Martin Abadi, and Roger Needham [39]."

1.2.3.2.1 BAN logic: "Burrows–Abadi–Needham logic (also known as the BAN logic)" is a set of rules for defining and analyzing information exchange protocols. Specifically, BAN logic helps its users to determine whether the exchanged information is trustworthy, secured against eavesdropping, or both. BAN logic begins with the assumption that all information exchanges happen on media vulnerable to tampering and public monitoring. This has evolved into the popular security mantra, "Don't trust the network." BAN logic uses postulates and definitions as do all axiomatic systems to

analyze the security protocols.

BAN logic protocol analysis: There are four steps to protocol analysis using BAN logic [39].

(i) Idealize the protocol.

(ii) Write assumptions about the initial state.

(iii) Annotate the protocol: For each message transmission "P $\rightarrow$ Q: M" in the protocol, assert that Q received M.

(iv) Use the logic to derive the beliefs held by protocol principles.

1.2.3.2.2 SVO logic: The successor of the BAN Logic is SVO Logic; introduced by "Syverson and Van Oorschot" [40]. SVO logic combines the three features and resolves the limitations of BAN Logic so that the rigorous verification of protocols is performed. The three components are: (i) Gong, Needham, and Yahalom (GNY), (ii) Abadi and Tuttle (AT), and (iii) Van Oorschot's (VO).

1.2.3.3 Theorem proving

Theorem proving is one of the formal verification approaches used for the mathematical reasoning of a system. The tools for the theorem proving approach are based on either "Higher Order Logic (HOL) or First-Order Logic (FOL)".

1.2.3.3.1 Isabelle: Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas with logical calculus. The main application is the formalization of mathematical proofs and in particular formal verification, which includes verifying the correctness of computer hardware or software and proving properties of computer languages and protocols.

A comparison among various model checking tools and theoretical approaches of BAN logic is provided in Table 1.3.

In this thesis, we have used manual verification of security protocols using BAN logic and simulation verification using the model checking tools of AVISPA and Scyther.

Table 1.3: Differences between formal approaches

Characteristic tool	AVISPA	Scyther	ProVerif	BAN Logic
Input	● Tool accepts HLPSL language	● Tool accepts SPDL languange	● Horn Clauses or ●Pi-Calculus	● Principals and Postulates
User Interface	● Provides own graphical user interface	Provides own graphical user interface	● Supports through command line interface	NA
Session	●Runs in both bounded and unbounded number of sessions	●Runs in either bounded or unbounded number of sessions	●Runs in only for an unbounded number of sessions	NA
Output	●Protocol holds for n fixed depth the protocol is false, and attack trace is shown protocol holds for all traces ●Attack Traces are generated when flaw existed in a designed protocol	●Attack graphs are generated when an attack is found in the protocol	●Trace is generated only for the property which is checked	●Based on inference rules and assumptions
Advantages	●Channels need to specify for communication	● Good in verifying multi-protocol attacks	●Channels need to specify for communication	● Applicable for simple and small protocols
Disadvantages	● It doesn't verify multi-protocol attacks	●There is no concept of channels	●Runs in only for an unbounded number of sessions	●It works on based on assumptions ●Not good for finding attacks

1.3 Objectives of the Thesis

The objectives of this research are as follows

1. To design secure mobile payment protocols using lightweight cryptography in order to provide less communication overhead and computation cost.

2. To design a secure mobile banking protocol based on SMS using elliptic curve cryptography (ECC).

3. To analyze the existing approaches for designing NFC based mobile payments and provide solutions for better security and performance.

4. To design an authentication protocol to overcome the attacks of phishing and pharming on mobile wallets.

5. To verify the proposed secure mobile payment protocols using formal verification methods such as BAN logic and perform simulations with formal verification tools such as AVISPA & Scyther.

6. To propose a threat model to detect and mitigate various unknown vulnerabilities in Android & iOS mobile applications.

7. To propose a security testing framework for identification of Man-in-the-Middle attacks in mobile applications and analyze the compliance of mobile banking applications to the "open web application security project (OWASP)" listed mobile security risks.

8. To propel innovative solutions to the evolving challenges in the area of mobile banking and its security by characterizing the mobile security, explore vulnerabilities in mobile applications, explore different tools to perform security testing, and to perform experiments to identify the security flaws in mobile banking applications.

9. To understand the research gaps that should be addressed and to indicate future research directions in this area.

1.4 Motivations of the Research

- The present study was undertaken to provide some improvements to the currently common mobile payment protocols, as wireless technology is more complex compared to wired networks. Full deployment of M-commerce for business end-to-end security needs to be ensured.
- There are three levels of security:
    1. Application level: Where we conduct mobile payments, thus protecting the legitimate security concerns of all stakeholders involved.
    2. Communication level: This level of security in the wireless infrastructure protects each wireless communication and is the topic of considerable on-going research.
    3. Device level: In this level, security is achieved with the assistance of mobile forensic tools in external or internal devices, e.g., mobile device, secure digital (SD) cards, personal log files −− stolen by an adversary, compromised devices, and physical threats −− fall comes under this category.
- From the above discussion, we have observed that current security solutions at the communication layer in mobile environments are not adequate for financial transactions. Therefore, security at the application layer must be added to achieve end-to-end protection for mobile financial transactions.
- Our research work focuses on the higher level security requirements of mobile payment and transaction security.
- Existing mobile banking applications (MBAs) use simple HTTP protocol to transfer customer data without major concern for security requirements. In most cases, MBAs receive fake or self-signed certificates. These are blindly considered as sound and valid, which leads to SSL/TLS Man-in-the-Middle attacks.
- To ensure end-to-end security for mobile payment protocols, the wireless public key infrastructure (WPKI) should be adopted.

- Some mobile payment protocols in the literature adopt WPKI for guaranteeing end-to-end security, but these protocols are not proposed in the secure element (SE) level.
- Some mobile payment protocols in the literature use the subscriber identification module (SIM) as SE, but the SIM is personalized by the mobile network operator (MNO), and is complicated to install and personalize for mobile payment applications by the issuer bank using over-the-air (OTA) programming.
- Existing mobile payment protocols in the literature consume more client resources (i.e., communication and computational costs).
- Most of the existing mobile payment protocols are not verified with formal approaches; this violates the security properties, and there is no guarantee that a protocol is free from various attacks or flaws.
- The above observations, have inspired us to conduct the research presented in this thesis.
- Formal verification of the mobile payment protocols must be performed, as it can detect flaws that lead to protocol failures, by using manual (such as BAN logic) and automated tools (such as AVISPA and Scyther).

1.5 Contributions of the Thesis

This thesis provides the following contributions for design, test and formal verification of secure mobile payment systems.

1. A systematic literature review is conducted on "Mobile Payments, Protocols, and Security Infrastructure" from which the approaches, payments methods, stakeholders, and security infrastructure for mobile payments have been identified.
2. A model is proposed to address the security of SMS using elliptic curve cryptography. It provides end-to-end SMS communication between the customer and the bank through the mobile application. The primary objective of the proposed

model is to design and develop a security framework for SMS banking that reduces computation and communication costs.

3. Secure end-to-end proximity NFC-based mobile payment protocol: used between peer-to-peer (P2P) payment, and payer-to-merchant (P2M) payment is presented, which uses elliptic curve cryptography to encrypt customer data. The proposed protocol provides end-to-end secure communication between customer and merchant through a bank by using secure reader and writer applications.

4. Propose a "VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps" as a security testing framework for identification of MitM and other associated attacks in mobile applications.

5. A secure authentication protocol for defending against phishing attacks in mobile banking with formal verification is proposed.

6. All mobile payment protocols proposed are verified using BAN and the automated formal verification tools AVISPA & Scyther.

7. Security testing on various mobile banking applications (MBAs) is performed as a case study, and the proposed model identifies various real-time attacks using both Android and iOS devices with the help of application level, device level, and communication level tools.

1.6 Thesis Organization

The research covering mobile banking and payment protocols and focusing on how these approaches ensure end-to-end security and which also includes a threat model for mobile applications is organized into seven chapters is described below:

- Chapter 1 provides an overview, the aims, objectives, and motivation on mobile payments and formal methods for security analysis. The chapter also provides the research contributions, the structure of the thesis, and conclusions.

- Chapter 2 presents a systematic literature survey on mobile payment systems. This chapter covers the topics of current trends, technologies, new patterns, and gaps in the existing literature.
- Chapter 3 discusses the SMS based mobile banking protocol that provides a wide channel of communication for banking in mobile commerce and mobile payments.
- Chapter 4 presents the secure end-to-end proximity NFC-based mobile payment protocol. The proposed protocol provides end-to-end secure communication between customer and merchant through a bank using reader and writer applications.
- Chapter 5 provides a threat model to detect and mitigate various unknown vulnerabilities in Android and iOS mobile banking applications, This chapter addresses security testing framework for identification of MitM attacks in mobile applications and analyze the compliance of mobile banking applications to the OWASP listed mobile security risks.
- Chapter 6 presents on mobile banking authentication protocol that discusses an authentication server (AS), which sends a nonce to the mobile user device to be signed so that he/she can avoid phishing and pharming attacks.
- Chapter 7 concludes the thesis and outlines future research directions.
- The appendix consists of the following:

    1. All of the proposed techniques are verified by using formal methods of model checking tools such as AVISPA and Scyther written in High-Level Protocol Specification Language (HLPSL), and Software Protocol Description Language (SPDL) which are available in the Appendix A.
    2. Elliptic curve cryptography (ECC) addressed in the Appendix B
    3. The publications considered in this thesis are listed in Appendix C.

Chapter 2

Systematic Survey on Mobile Payments, Protocols, and Security Infrastructure

One of the fastest-growing technologies over conventional payment methods is mobile payments because most people use smartphones for bill payments, online shopping, mobile banking, etc. This chapter presents a systematic literature survey of mobile payments and characterizes the state-of-the-art research in this area covering articles published from 2000 to 2018. We analyze the trends, technologies, new patterns, and gaps in the existing literature. The scope of the survey is based on three research approaches: stakeholders, technologies, and security in mobile payments. We describe critical challenges and provide recommendations that will shape the focus of upcoming research efforts.

2.1 Research Methodology

A "systematic literature review (SLR)" is a research methodology which comprises interpretation of research studies, critical assessment of all available research topics, and evaluation of reports, or it finds a solution for a particular research problem [41]. SLR resolves bias and follows an exact and strictly sequential methodological process in the literature survey. Our survey follows the suggestions for a SLR process suggested by

Kitchenham [41]. We have also considered applying SLR to the software engineering domain as proposed by Breton et al. [42]. The entire process includes three main tasks: (i) Review planning, (ii) Conducting the review, and (iii) Reporting the results of the review [43, 44] as shown in Figure 2.1. Based on these suggestions and guidelines, we have constructed the Research Questions (RQ) given below, which serve as the cornerstones for conducting the systematic literature survey.

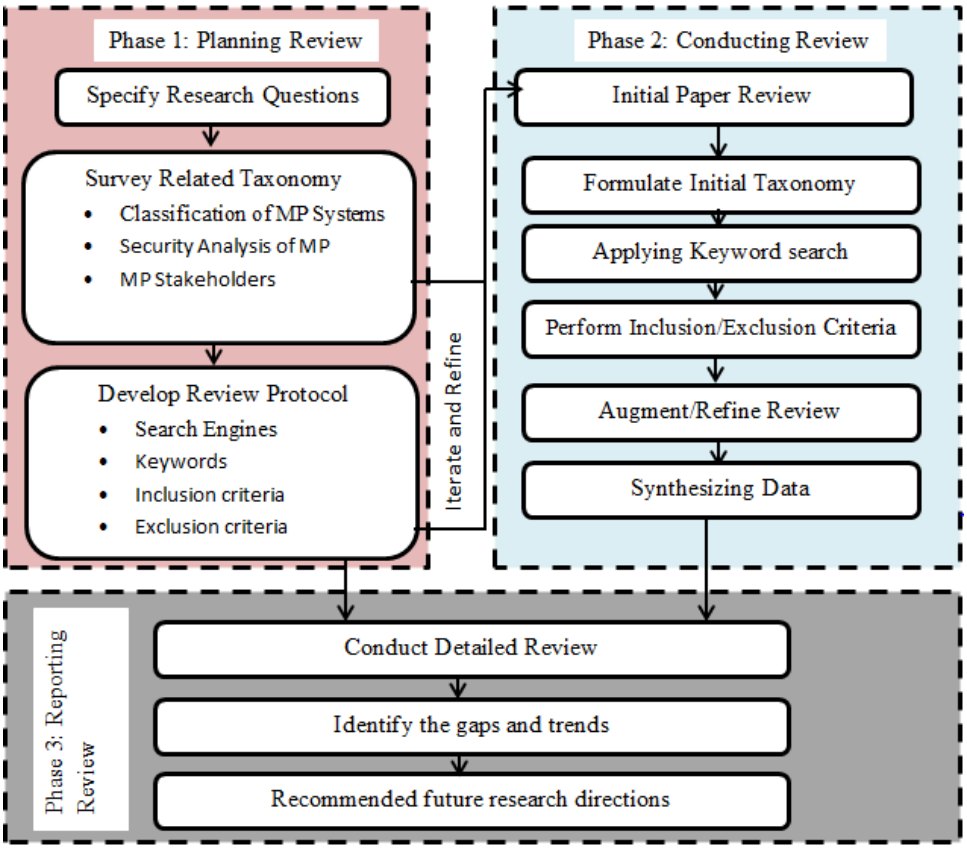


Figure 2.1: Tasks and research process flow

- RQ1: How the classification and characteristics of the different mobile payment systems used by various countries affect the telecommunications infrastructure and success of the markets of mobile payment services.
- RQ2: What levels of security and trust mechanisms (are provided for mobile payments) for all stakeholders involved in mobile payments.

- RQ3: What is the impact of the continuous evolution of existing infrastructure, financial services on design and development of secure mobile payment protocols, and how do these methods assist in the success of the mobile payment services market?
- RQ4: What are the current research concerns and what is the future scope of mobile payment systems?

2.1.1 Research tasks

To address the above "research questions (RQ)", we have orderly arranged our work into a process flow customized to our determined goals, while stick to the three-phase systematic literature review process including review planning, conducting the review, and documenting the analysis. In this planning phase, first, we define the review protocol, which consists of the choice of search engines, the initial selection of keywords regarding classification of mobile payments based on technology, and the selection of candidate papers. The initial selection based on keywords is an iterative process that involves inclusion/exclusion criteria based on the title, abstract, and the scope of the survey. In this process, inclusion/exclusion criteria and keyword search expressions may also need to be refined to stimulate up to date searches. We maintained the balance between the literature review protocol and the resulting paper collection, and we then conduct the paper reviews to validate the selections.
For RQ1 and RQ2, we focused on comprehensive taxonomy suitable for classifying mobile payment systems. The extracted data from various research papers cover the abstract, introduction, proposed method, contribution, results and conclusion sections to evaluate new technologies and methods that could expand and refine our taxonomy. For RQ3 and RQ4, we analyzed the results from RQ1 and RQ2 and strove to identify gaps and trends, again using the taxonomy as a critical aid.

2.1.2 Developing and validating the literature review protocol

In this section describes the study on review protocol, containing our search technique and detailed discussion on inclusion/exclusion criteria.

2.1.2.1 Search technique

We have used prestigious databases and search engines in our literature review protocol such as IEEE Xplore, Google Scholar, Springer link, Science direct, and ACM digital library, with the aim of identifying standard refereed research articles, including journal papers, reputed conference proceedings, and short papers. Here we discuss the scope of our literature review as illustrated in Figure 2.2. We have focused on preferred keywords to search the research papers' meta-data, titles, and abstracts, such as keywords and tags. To explore the related keywords in three research domains, the search query is organized as a conjunction of three research domains as inclusion criteria, namely, RD1: Mobile payment systems, RD2: Mobile payment stakeholders, and RD3: Security infrastructures and goals. These RDs are used in the SLR under various forms and using synonymous (coincident) words. The keywords which are used for the search are summarized in Table 2.1. The list of keywords used in the SLR was continuously refined, and new keywords were included during the search process. In summary, our search query is described by the following formula:

$$search_query = \bigcap_{d \in \{RD1, RD2, RD3\}} \left(\bigcup_{keyword \in (KW)_d} keyword \right)$$

Where RD_i represents the three research domains, and KW_d is the set of corresponding keywords defined for domain d.

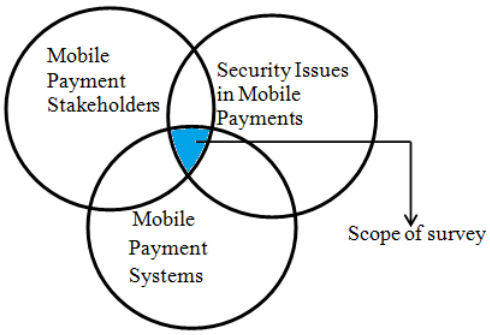


Figure 2.2: Survey on three RDs

Table 2.1: Revised search keywords

Research Domain (RD)	Keywords (KW)
Mobile payment stakeholders	Customer, Merchant, Acquiring Bank, Issuing Bank, Payment Gateway
Mobile payment systems	NFC, SMS, USSD, Bluetooth, RFID, UICC, SIM, M-Wallet, Mobile banking
	Remote-payments, Proximity-payments, Account based, Token-based payments
	QR code, IrDA, Micro payment, Mini payment, Macro payment
Security infrastructures and goals	MitM, Phishing, Session Hijacking, Masquerade
	Traffic analysis, Account lock-out, Malware in-app, Unintended data leakage

2.1.2.2 Selecting studies

To select appropriate studies, search keywords, as conducted by [42], were selected to conform with our RQ's as described in Section 2. We collected 515 peer-reviewed research papers that were published between 2000 and 2018 (Mar 2018). Not all the retrieved research papers based on the search query were suitable within the three research domains of the survey. Hence, we used the following criteria, i.e., inclusion/exclusion, necessary for further filtering the research papers. The inclusion/exclusion criteria applied to title, abstract, and conclusions as shown in Table. 2.2.

Table 2.2: Criteria for inclusion and exclusion

Criteria	Prenomial	Scope of survey	Goals
Inclusion	I. The inclusion domain covers the research papers that are in the form of peer-reviewed articles and written in English. We have not considered other language papers even thou relevant topic. II. The search query was used to search for keywords, abstracts, and titles. In some exceptional cases, it is possible that our search query may have failed to identify some relevant publications. III. Research papers that are straightforwardly mentioned about stakeholders, systems, protocols, architectures, security issues, and new payment evolution methods that are used in mobile payment history.	1. Mobile payment stakeholders	I. The participants and their expectations in mobile payments. II. Study of strategic issues. III. Multi-perspective framework for mobile payment systems
		2. Mobile payment systems	I. We presented a detailed classification of mobile payment systems based on technologies and channels used in current mobile payment scenarios. II. Discussed approaches on protocols for secure mobile payment transactions. III. Approaches on biometrics in mobile payment systems.
		3. Security issues in mobile payments	I. The study considers taxonomy of vulnerabilities affecting a mobile payment systems. II. Approaches to Security and trust for mobile payments. III. Approaches on Formal verification of protocols, architectures, tools for mobile payments and mobile banking.
Exclusion	I. We have excluded white papers and another kind of technical reports for new technology innovations and security related issues in mobile payments. II. We do not consider papers regarding abstracts, editorials, and short papers which is less than four pages. III. We excluded short papers and non-peer reviewed scientific studies because they would not provide valid and relevant data for our survey.	1. Mobile payment stakeholders	I. We are not considered for mobile stakeholders of individual opinions, and country wise payment adoptions for this survey.
		2. Mobile payment systems	I. Exclusively focused on developed techniques, protocols, systems and models in mobile payments.
		3. Security issues in mobile payments	I. Considered only on methods for mitigation of security risks and threats, but not on any vulnerability detection techniques; examples are static, dynamic, manual and automated testing. II. We are not focused on a mobile device operating systems such as Android, Windows iOS, Symbian, BlackBerry, and low-end device operating systems.

2.1.2.2.1 Selected papers: The statistics of each phase of paper collection for each database are shown in Table. 2.3 and the phases are illustrated in Figure 2.2. The table

contains different databases, and selection of papers is divided into four categories, where the first row indicates keyword-based search of the full paper for each database. Here we have used the search engine associated with each of the five databases; each database searches our query differently. We performed second level verification over initially collected papers, in a consistent manner, based on the same keywords, which are represented in the second row as initial filtering. After this step, the search results of all databases were merged into a single repository for further review and filtering. The fourth row represents (after applying merging on selected papers). In Figure 2.3, We demonstrate the survey papers distribution based on an application of mobile payments and other domains such as mobile payments, mobile banking, security and mobile stakeholders. The number of papers selected for this survey led to 316 publications. The number of selected papers by year-wise publication is given in Figure 2.4. We collected papers published from 2000 to March 2018. Initially, the papers on mobile payments were scant, but gradually increased between 2003 and 2017, and reached a peak in 2010.

Table 2.3: The number of collected papers in each phase using various search engines

Selection phase	**Search databases**					
	IEEE Xplore Digital Library	Springer Link	ACM Digital Library	Google Scholar	ScienceDirect	All
Search based on keyword	1890	1028	536	4229	1123	—-
Initial filtering	783	378	238	408	269	—-
Merging						2195
Selection criteria						316

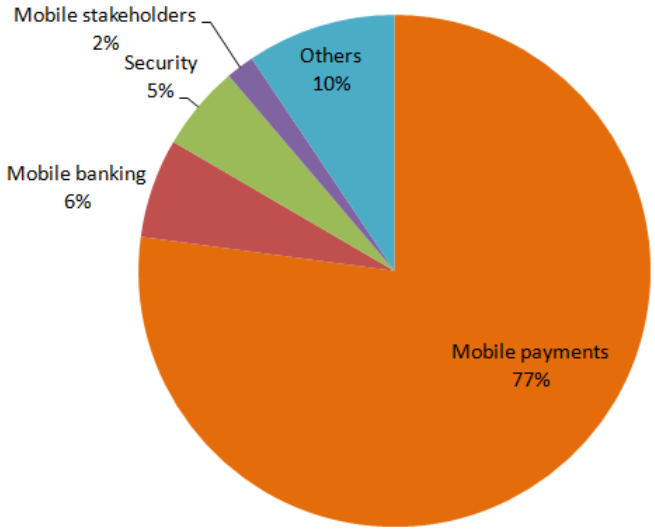


Figure 2.3: Survey papers distribution based on an application of mobile payments and other areas

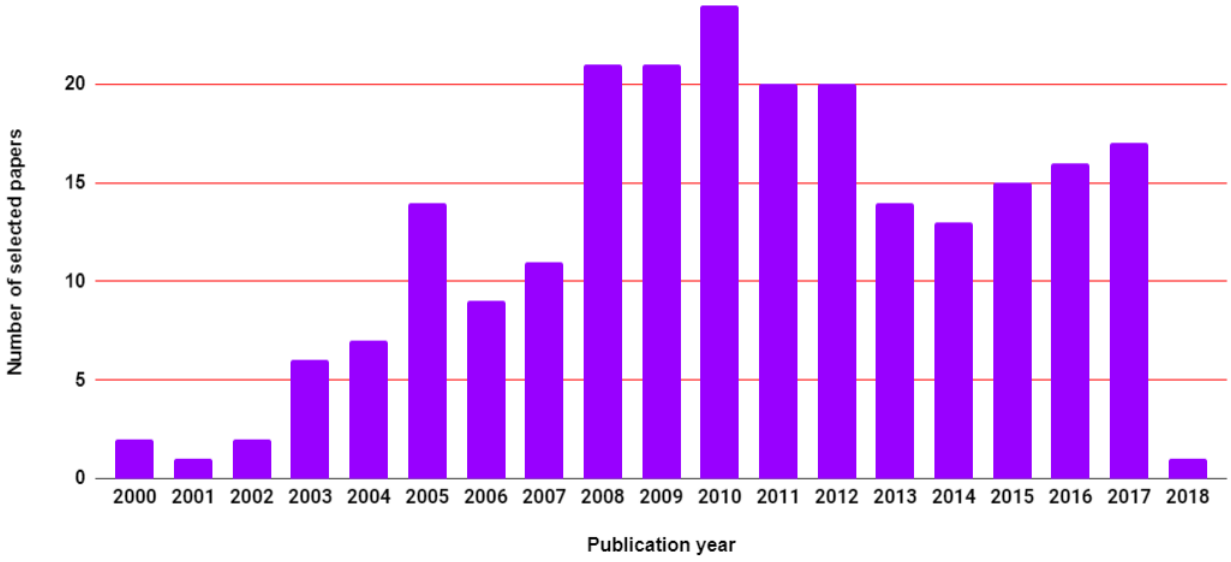


Figure 2.4: Year-wise distribution of surveyed papers

2.2 Mobile Banking

One of the cornerstones for the growth of the IT sector is banking investments −−"Not only to run the internal business activities and to promote products but also to dema-

terialize the customer relationships [45]". Mobile banking uses a mobile device to access a bank account with the help of either a mobile wallet or mobile app (application) which is downloadable in the Google play store or app stores; they are available for various operating systems such as Android or iOS in a mobile phone. Currently, mobile banking is an additional channel for furnishing the usual services provided by online banking (OLB). Regarding mobile banking, statements given by authors such as (Pousttchi and Schurig, 2004) "The simultaneous and increasing diffusion of mobile phones and especially WAP-enabled devices has made the transformation of banking applications to mobile devices a logical development in electronic banking" as cited in [46] are applicable. As argued by the information systems (IS) researchers (Herzberg 2003;Kleijnen et al. 2004; Laukkanen 2007; Laukkanen and Lauronen 2005), it has been proposed that "mobile banking can be considered as one of the most significant technological innovation, which is emerging as a key platform for expanding access to banking transactions via mobile or handheld devices, and operating wireless communication technologies" (as cited in [47]). Mobile-banking (m-banking) may have new characteristics such as simplicity, usability, flexibility, and ubiquity compared to conventional banking methods (e.g., ATM, non-mobile Internet banking), and the public has shown interest in new devices and technology innovations in the financial sector, supporting the adoption of mobile banking [48]. With the development of m-banking, banks enable customers to conduct activities such as account balance inquiry, money transfer from one account to another through a mobile phone, bill payments, mini-statements, etc. instead of visiting home banks and Internet banking based on computers [49]. As discussed by [50], mobile banking is a means of execution of bank transactions using mobile phones and the author summed this up by stating that "The term mobile banking usually refers to the following three basic applications: mobile account, mobile brokerage (purchase and sale of securities) and mobile financial information services (account balance, securities deposit account, current stock prices, stock alerts, etc.)". In the past, many bank customers did not utilize the services of banks using mobile phones as a consequence of the lack of security or encryption of SMS messages, so customers were not able to send personal confidential information through the mobile network [51]. As stated by Mayer, Davis, & Schoorman in 1995, "Knowledge-based trust is a function of individual perceptions of the competence, benevolence, and integrity of a product, service, or person" [47]. In case of m-

banking, the customer assumes that the services provided by the bank are benevolent and a bank provides good-faith agreements (integrity) and other security-related benefits such as authorization and confidentiality where no untrusted party involvement exists during transaction executions. Customers' belief or faith play a major role in banking practices adoption. The customer perspectives of adoption of mobile banking have been analyzed by various recent studies [52, 53, 54].

2.3 Mobile Payments

Mobile payments are one of the means to increase revenue growth in financial and commercial industries. The gross domestic product (GDP) is predicted to grow by approximately 38% from 2010 to 2020. This is because of predicted increases in conventional banking and mobile payments. The growth of mobile payments and mobile payment users is increasing daily, and is quite impressive, as shown in Figure 2.5. Mobile payment users are emerging from many parts of the world (accounting for $447.9 million in transactions as per 2016 statistics). Mobile payments spending expenditures throughout the world were $1.3 trillion in 2017. According to Juniper Research, "Mobile Payment – Check It Out (August 2012), p.3", increased by 400% in 2012 [55]). These statistics give some insight into how mobile devices have played a significant role in payments. The massive increase of mobile payments of approximately $1.721 above billion in 2017 is shown in Figure 2.6, much of which involved person-to-person money transfers, i.e. (from one account to another account). Part of this growth is attributable to many conventional banks offering their services via mobile bank apps through application-based or peer-to-peer based approval. Details on mobile payment modes and channels are discussed in next subsection. A considerable amount of money transfer has also occurred because of the overall growth of expenditures, as seen in places such as Kenya using M-PESA and many countries, especially in India[56] through demonetization of hard currency to digital currency. Merchandise purchases using smart devices as the payment platform are the second-highest expenditures (approximately $168 billion in 2017). For example, PayPal is one of the most popular merchandise as third-party application provider online commerce sites that permit using the mobile device transactions. Other services contributing to the increasing substantial amounts of global expenditure in 2017 using mobile devices are bill

payments ($36.8 million), airtime top-ups (prepaid Top-Ups) ($8.5 million), ticketing ($6.8 million) and other payments using mobile devices amounting to ($3.9 million). The differences between mobile banking services, payment services, and value-added services are shown in Table 2.4. Global mobile payment spending from 2010 to 2017, segment-wise (in billion U.S. dollars).

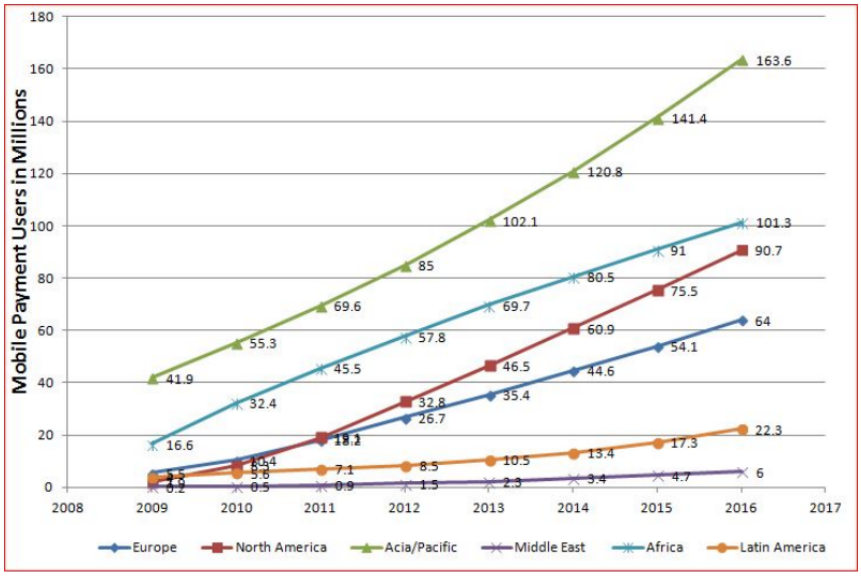


Figure 2.5: Number of mobile payment users from 2009 to 2016, by region (in millions)

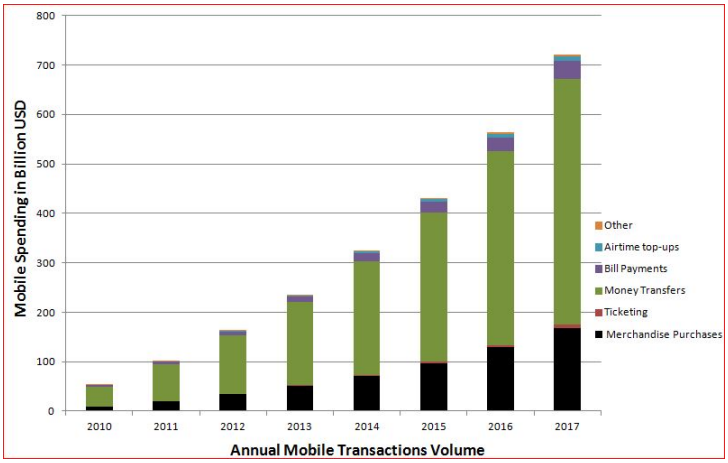


Figure 2.6: Global mobile payment spending from 2010 to 2017, by segment (in billion U.S. dollars)

Table 2.4: Definitions of terms in comparative functionality analysis

Mobile banking services	**Mobile bank app and Basic services**	Bank enables customers to get various services mobile using smartphone mobile accessing through mobile bank app via bank server such as mini statement, transaction history, location of bank branch/ATM, viewing account statements, cheque book request, receiving bank alerts, fund transfers within the same bank and different banks, etc.
	Text banking	It is applicable for low-end mobile devices to do banking. Mobile bank users can send a text message for knowing the account balance, historical transactions, funds transfers, mini statement and verify the credit card bill payment or next credit card bill due, etc. examples of text message banking are SMS, USSD.
Mobile payment services	**P2P payments**	Collect request which can be organized and paid as per need and convenience. Peer to Peer payment (P2P) is a system that enables several bank accounts into one mobile application, (irrespective of any bank which is participating). It is one method to integrate multiple banking features, merchant payments and seamless fund routing into one platform or one hood. Typical examples of such applications are virtual address as e-mail address, phone number, etc.
	Bill payments and Recharges	Make post-paid & pre-paid payments in advance, view upcoming bills, create and modify payee list.
	Virtual wallet	Banks provide mobile wallets for storing digital cash for purchasing goods and online payments.
	POS solutions/NFC	Tap and pay: Users can store their personal confidential information regarding account/card details on a microchip to use mobile phones directly for payments at ATMs and merchant POS machines.
Value added services	**Image/camera based Banking**	Take a snapshot of the paper, bill/cheque to deposit/send to/from the bank. Also, capture account opening details via an image of the customer's identity cards such as their driver's license.
	Cloud storage	Store documents in the cloud to view while on the move. For example bill payments, account statements, etc.
	Cross-selling & Marketing	Customers can use and buy banking products using mobile phones; They can also look at today's offers, benefits, reward points, and deals with other retail partners. Examples are bank provides housing loan deal in collaboration with a real-estate builder, free coffee, etc.
	Personal financial Management	Tools to maintain every customer data on expenses, set budget limits, govern monthly expenditure, view consumed amount as pie charts and bar graphs and get alerts if account balance falls below the set limit.
	Virtual support	Virtual support for real-time customer interaction 24/7 via phone calls, sending e-mails, and conduct of tutorials.
	Wearables	Perform banking services on Google glasses, smart watches, and other such wearable devices.
	Biometric apps and Security features	It allows the customer to perform specific activities on mobile banking; such as balance inquiries, fund transfers, locate nearest ATMs, branches, using voice commands on mobile devices. Banks also includes additional security mechanism, i.e., fingerprint print scanning for login to his/her account.
	Augmented reality Technology	Customers can view in a 3D format their historical data on transactions, account balances, find ATMs location, etc. by positioning the mobile camera over their cards.
	Social media banking	View account balances, fund transfers, information on transactions, and customer service assistance or help via Facebook and Twitter.

2.4 Classification of Mobile Payment Methods

Mobile payment methods can be classified into various categories based on the method of payment and type of channel used, as shown in Figure 2.7. This diagram specifies the mobile payments survey based on channels, device, payment value, and timing of mobile payments. These payments, categorized according to the medium of payment, are as follows [57]:

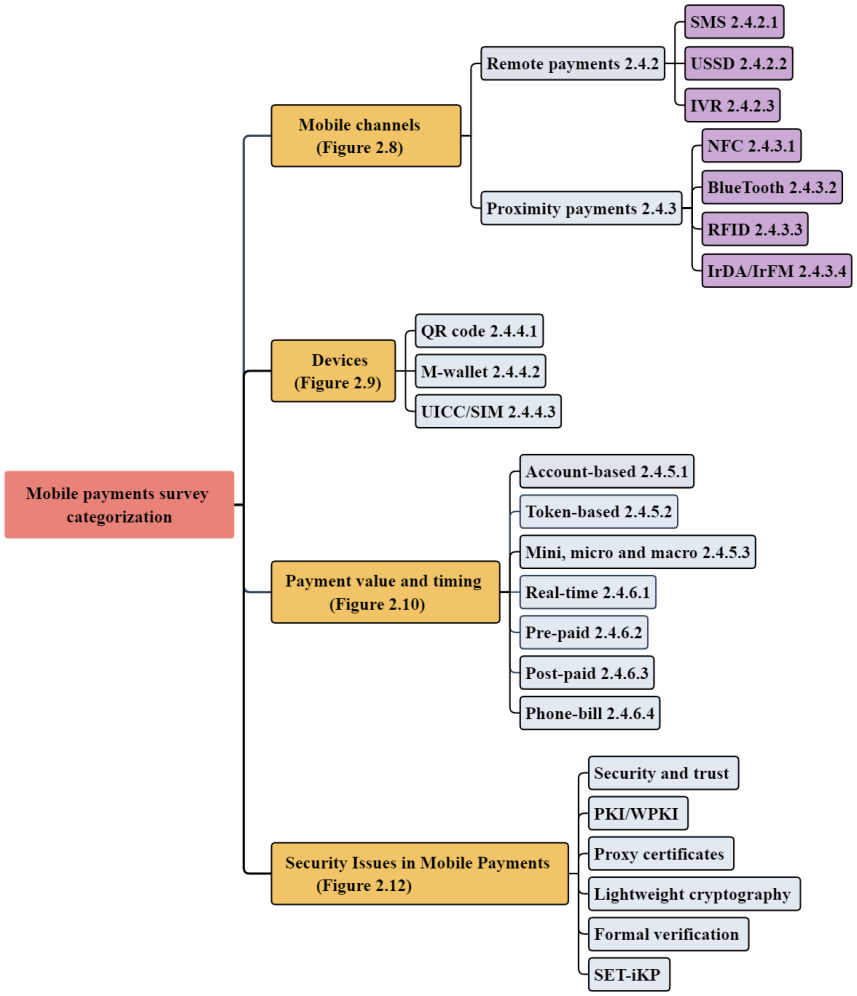


Figure 2.7: Mobile payments survey categorization

2.4.1 Mobile channels

In this section, we describe the most popular and well-established mobile payment services that make use of the various channels as shown in Table 2.5.
The types of mobile payments based on channels can be divided into two parts: one is remote payments the other is proximity payments.

Table 2.5: Mobile payment technologies and their advantages, disadvantages, and applications

Channel	Advantages	Disadvantages	Applications
USSD	• Applicable for all mobile devices including low-end devices • Synchronous connection • Simple to use	• Cash transfer limit is low • Confidential data in plain-text format • Difficulty in remembering codes • USSD message not stored in mobile inbox • USSD messages are in flash form • Commonly USSD is MNO strategy model	• Remote payments • Mobile banking
Bluetooth	• Compatibility with smartphone and tablets • Supported by PayPal and Apple • Easy to use • Minimum cost • Less power consumption • Compact size • User friendliness	• Lack of security • User lack of trust in payment systems • Operates at a short distance-10 to 100 meters	• Proximity payments • Mobile payments
SMS	• Applicable for all mobile devices including low-end devices • Simple to use	• SMS usage cost is more •Confidential data in plain-text format • Difficulty in remembering SMS codes • Delayed payment • Missing confirmation of SMS delivery • Commonly SMS is MNO strategy model • SMS size is not more than 160 bytes	• Remote payments • Peer-to-Peer payments • Mobile banking
NFC	• Fast payment transaction process •Promising pilot tests • Easy to use • NFC project biggest success in many countries	• Tag cloning and tag impersonation • Tag content changes • Tag replacement and tag hiding • Use up to 20 cm • Available only on smart phones • Value-added services are required for success	• Proximity payments • e-payment • e-ticketing • loyalty services • smart advertising • mobile banking • social services
RFID	• Less power consumption • Synchronous connection	• Operates at a short distance up to 10 feet • RFID tags cost is more when compared with paper barcodes • Security risks	• Access control • inventory control • Access control smart cards • Toll collection systems • Railroad car monitoring • Public transportation
M-Wallet	• Customers can keep several credit or debit card payments using the single wallet. •All transactions in M-wallet are based on text messages hence it is applicable for all types of mobile devices.	• SMS delay, which can raise the time necessary to complete a transaction [4] • Loss of message that will cause the failure of a transaction	• Mobile payments • Mobile banking • Recharge • Recurring payment

2.4.2 Remote payments

Payments are independent of a customer's location. Examples are SMS-based payments, USSD, UICC, m-tickets, digital cash, peer-to-peer payments, etc. The remote payments are explained in detail in the next subsection.

2.4.2.1 SMS

The "short message service (SMS)" is used widely as a channel for communication and payment in the m-commerce industry. SMS services offered by banks provide more benefits to customers as well as the banks themselves. The SMS mobile payment protocol is implemented in two ways one is application-based, and the other is Peer-to-Peer (P2P).

- **Application-based:** It is used in mobile banking, the customer and bank can perform SMS communications with the mobile application. To perform SMS banking, initially, both payer and payee register his/her mobile numbers with the respective banks. After that, the bank provides a mobile money identifier (MMID) and mobile personal identification number (MPIN) to customers to allow SMS banking. The payer uses a mobile device to create and send an SMS containing payer MMID, payee MMID, amount, and mobile number of the payee to the bank address over the mobile network operator via the SMS gateway through global systems for mobile communications (GSM). The payment gateway switch provides a settlement of payments between payer and payee using secure SSL/TLS networks. Each message exactly fits in 160 bytes. The related literature on SMS-based using application is [24, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73].

- **Peer-to-Peer (P2P):** The P2P SMS-based mobile payments can be used as a protocol that transfers messages between applications, such as message server and the "short message service center (SMSC) of the GSM service provider over an IP link. This link can be a leased line or the Internet. The related literature on SMS-based using P2P protocol is [74, 75, 76].

2.4.2.2 USSD

The "unstructured supplementary service data (USSD)" protocol is a capability of a GSM network which can be used to support wireless communication services and location-specific services, and this service allows mobile banking transactions using basic feature mobile phones; there is no need to have a mobile Internet data facility for using USSD based mobile banking. It is envisioned to provide financial penetration and inclusion of underbanked societies in mainstream banking services.
For transferring data between a mobile device and USSD application, a user enters a shortcode on a mobile device. The user who entered the code is standardized, and content is described for each different service. For example, the content includes the USSD code, payee account no., payer account no., amount, beneficiary mobile number, etc. Initially, the customer registers the mobile number with the respective bank. The customer sends a USSD request to a USSD gateway using a shortcode. The USSD gateway service provider communicates with the GSM network through the SS7 protocol. Then the USSD gateway creates a session and sends session information to a respective application. The application sends this data back to the gateway in the standard format. The USSD gateway creates a flash message or USSD message and sends it back to a respective customer. Banks communicate with the payment server at the back-end for each request made by the user. USSD works with any mobile phone since the coded commands are entered in the same way as a phone number (e.g., *123#1234567890#). The related literature on USSD-based payments is in references [77, 78, 79, 80].

2.4.2.3 IVR

Interactive Voice Response (IVR) is a method that allows a customer to pay for the services availed from a merchant, using a phone. Either a mobile phone or a land-line can be used for transacting through the IVR. IVR service can be used to pay for a host of products and services, be it ordering movie tickets or airline tickets, home delivery of food or even vacation bookings. The customer can also buy physical goods through the DirecPay IVR system [81]. IVR system is currently configured in English but will soon enable merchants to offer this service in multiple languages.
The related literature on remote payments based on channels is shown in Figure 2.8.

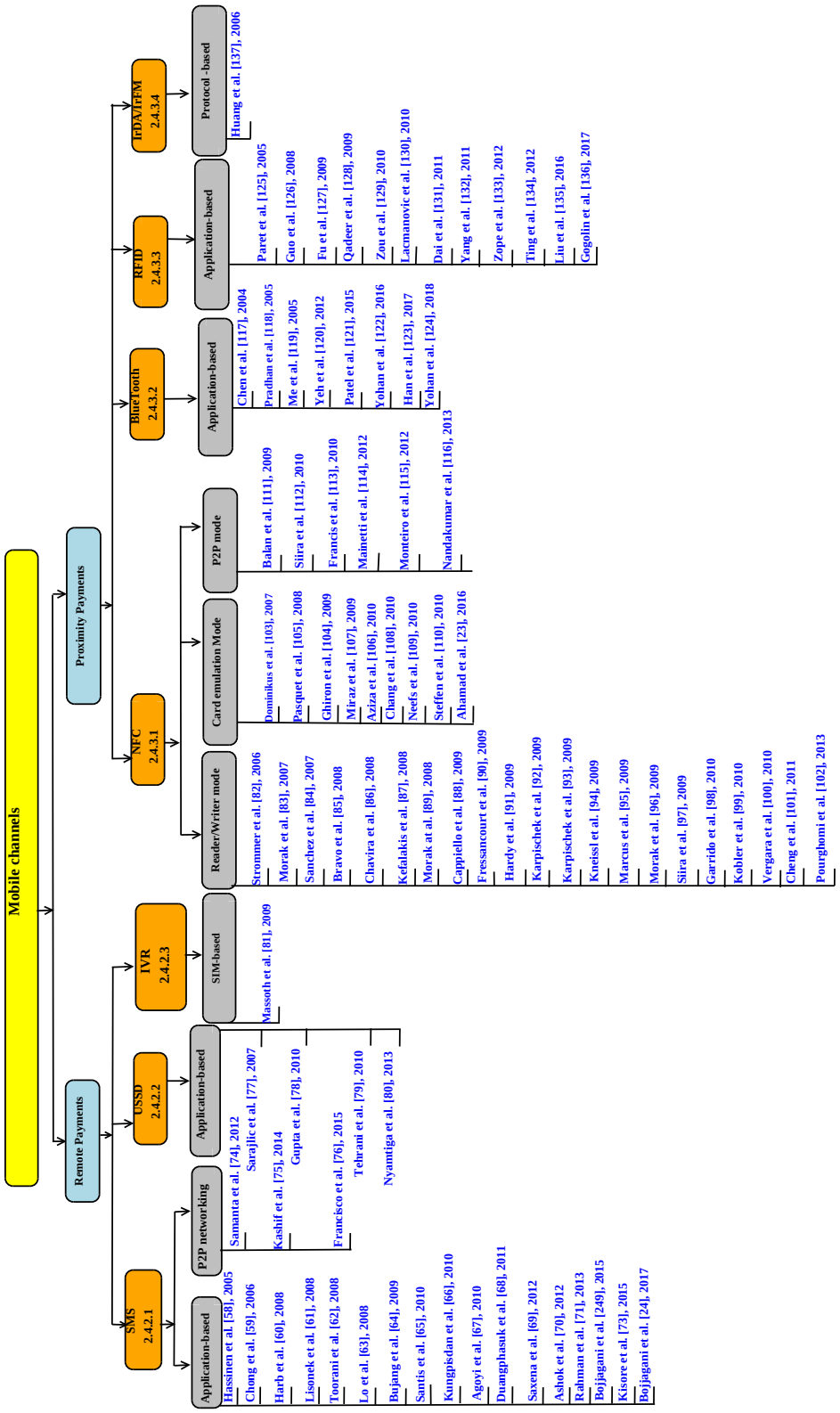


Figure 2.8: Survey of literature on mobile channel-based payments

2.4.3 Proximity payments

For this class, payments are made using a mobile device which is locally communicating, (e.g., via NFC, RFID, IrDA, Bluetooth, etc.,) with ATM/POS machines. Proximity payments are used by mobile devices to pay for parking, motorway tolls, vending machines, point-of-sale items, and so on. The related literature on proximity payments based on channels is shown in Figure 2.8.

2.4.3.1 NFC

Near Field Communication (NFC) is a technology that enables short-range communication between NFC devices, e.g., an enabled phone at one end, and another NFC enabled mobile phone, an NFC reader, or an NFC tag at the other end. NFC operates in three modes, reader/writer, card emulation mode, and peer-to-peer.

- **Reader/Writer mode:** In this mode, communication occurs between an NFC mobile and an NFC tag. The primary intent of the communication is either writing or reading from or to an NFC tag. This can further be divided into two different modes: one is the reader mode, and the other is the writer mode. In reader mode, the data contained in the NFC tag is read by the mobile device; whereas in writer mode, the mobile phone writes data into an NFC tag. The related literature on reader/writer mode using NFC payments is [82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102].

- **Card emulation mode:** For this mode, an NFC mobile functions as a contactless smart card. Using this mode, e-payments, e-ticketing, data or money transfers, etc. are enabled. Examples of an emulated contactless smart card are loyalty cards, debit cards, credit cards, and so on. The related literature on card emulation mode using NFC payments is [23, 103, 104, 105, 106, 107, 108, 109, 110].

- **Peer-to-Peer mode:** Using this mode, any information can be shared between two NFC mobile devices. Because both devices have integrated power, both use its energy by being in active mode. In this mode, the communication between NFC devices occurs in a bidirectional half duplex mode, meaning that when one NFC mobile is sending, the other has to listen and can start sending data after

the other finishes. The related literature on P2P mode using NFC payments is [111, 112, 113, 114, 115, 116].

2.4.3.2 Bluetooth

Bluetooth is one type of wireless communication technology. It is widely used on mobile devices such as smartphones, PDAs, mobile phones, tablets, etc. Bluetooth enables new payment paradigms, called proximity payments, by allowing quick connection to devices for performing mobile payments. The related literature on BlueTooth-based mobile payments is [117, 118, 119, 120, 121, 122, 123, 124].

2.4.3.3 RFID

Radio Frequency Identification (RFID) technology provides communication via radio waves to share information between an RFID reader and an electronic RFID tag, commonly attached to an object and is primarily used for tracking and identification. RFID uses electromagnetic waves for data transmission, involving different ranges depending on the magnetic field and frequency. A small integrated circuit is built into the RFID tags, which can store a small amount of data as well as miniature applications. The related literature on RFID-based mobile payments is [125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136].
RFID tags are of two types: one is the active tags and the other is the passive tags.

- **Active RFID tags:** Active RFID tags have their internal power source to power the integrated circuit that generates the outgoing signal. Compared to passive, active RFID tags are more reliable. These tags can conduct a session with an RFID reader at longer distances but are more costly.

- **Passive RFID tags:** These tags require no internal power supply. They contain an integrated circuit and an embedded antenna. These passive tags are powered by the incoming signal over a radio frequency. Passive tags can communicate within a range of 10 feet depending on the chosen antenna design, size, and radio frequency.

2.4.3.4 IrDA/IrFM

There is a widespread distribution of millions of mobile devices such as mobile phones and PDAs with "infrared data association (IrDA)" ports to consumers worldwide. Rapid advancements in both device technology and users' capabilities are the foundations for a new digital payment system, which promises to change the traditional method of payments. By utilizing standard IrDA indoor infrared technology, "infrared financial messaging (IrFM)" is a financial messaging specification with great prospects for future digital payment systems. The related literature on application-based IrDA/IrFM is [137].

2.4.4 Devices

In this category of payments, secure element (SE) is needed to provide secure storage and execution of mobile payments. SE is a combination of software, hardware, protocols, and interfaces [138] embedded in a mobile device that enables secure storage. The related research on the device-based transactions is shown in Figure 2.9.

2.4.4.1 QR code

A secure quick response-payment (QR-Pay) system for mobile devices was proposed by [139]. Their proposed framework consists of various steps for making a payment: (i) Initialization, (ii) Shop's payment information creation, (iii) Customer payment information confirmation, (iv) Customer payment approval, and (v) Payment approval confirmation. With their framework, the customer can initiate payment with the merchant, but although the customer may have access to the smartphone with an Internet connection, the merchant shop requires a terminal with a display and Desktop PC with an Internet. The proposed system provides a secure channel for transmission of payment data by using a QR-Code that demonstrates the payment data, which is revealed in a window at the shop. Thus, a customer can scan the QR-Code with the help of a camera attached to the smartphone. The entities for the QR-Code system are (i) Customer, (ii) Merchant's shop, (iii) Payment gateway, (iv) Issuing bank and (v) Acquiring bank. The related literature on QR-based mobile payments is [139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150].

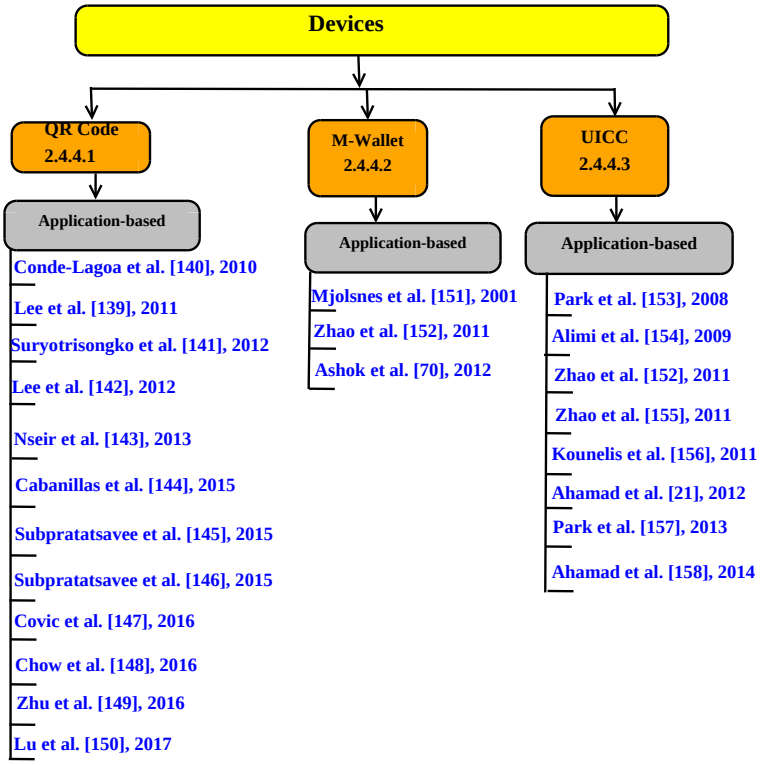


Figure 2.9: Survey of literature on mobile device based payments

2.4.4.2 M-Wallet

The mobile wallet is mobile payment application software that resides on the smartphone and contains information about the user (including his/her bank account number, customer ID, credit card details, etc.) that allow the user to make payments. Mobile wallets can retain the transaction history of credit or debit card payments.

The M-wallet payment introduced by [70] is an alternative to card payments and token-based payments. M-wallet payments consist of various entities: Customer, merchant, and M-wallet system. Any customer can use the M-wallet to purchase any service or product from the merchant, but before doing so, he/she must register the mobile number with the system by sending an SMS to the M-wallet system. After receiving the number from the user, the M-wallet system generates a PIN number and sends it to the respective customer, which can be changed later by the customer. Now M-wallet creates a virtual account for the customer with an initial zero balance. From this mo-

ment onward, the customer can perform deposits, transfers, recharge their balance, and perform other merchant payments. The related literature on M-wallet payments is [151, 152].

2.4.4.3 Universal Integrated Circuit Card (UICC)

This is one of the physical smart cards upon which the "subscriber identity module (SIM)" or "universal subscriber identity module (USIM)" is implemented. The UICC-based SE is compatible with all smart card standards and can host various applications issued by several providers. The UICC ensures the security features of data integrity, privacy, authentication, and confidentiality.
The related literature on UICC-based payments is [21, 152, 153, 154, 155, 156, 157, 158].

2.4.5 Based on payment value

In this category, current mobile payment methods in use or under test can be classified by considering how money transfer is structured [57, 159, 160]. The related research based on the payment value is shown in Figure 2.10.

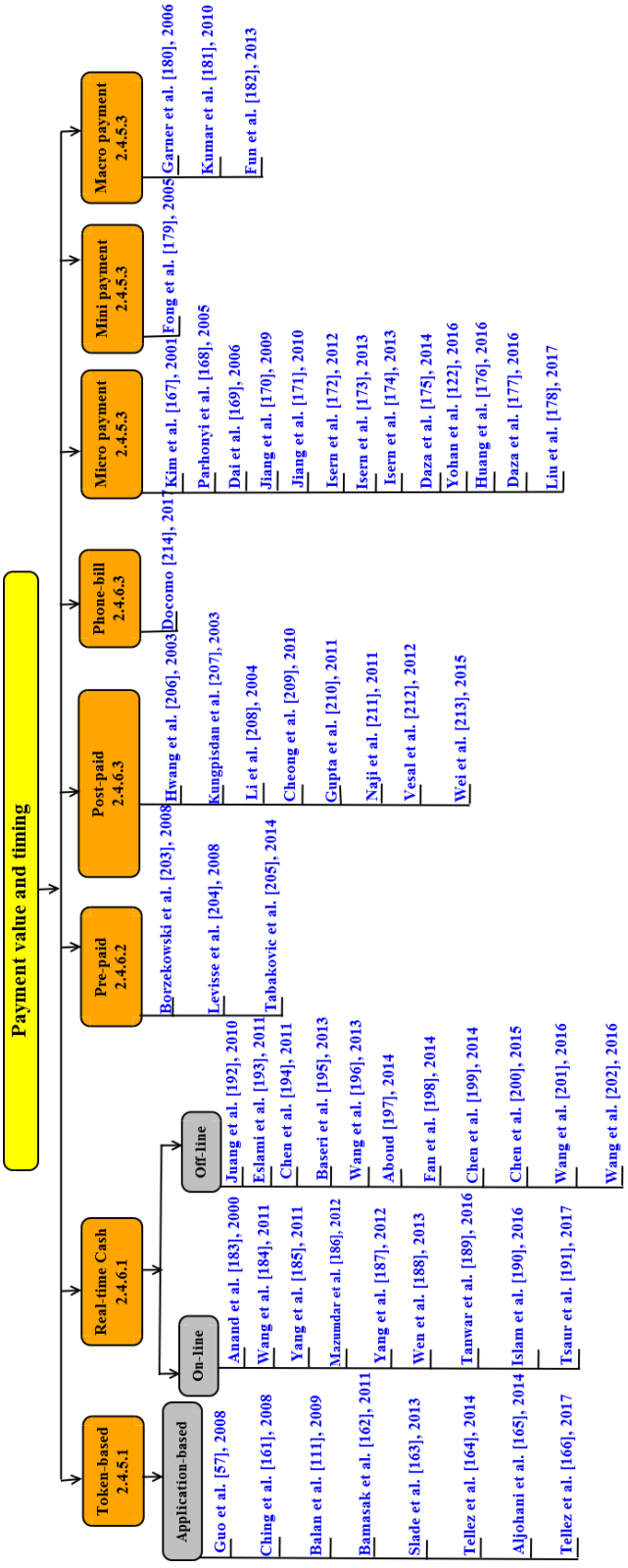


Figure 2.10: Survey of literature on mobile payment value and timing

2.4.5.1 Token-based (E-coins)

Using electronic tokens is the alternative to maintaining an account for each customer. Electronic token-based payments are another way to maintain an account for each customer. A token is given by the bank to the customer for performing a transaction. A token is a medium of exchange representing some monetary value. Whenever a customer initiates a transaction, it is necessary that the customer convert his/her real currency into an electronic token format. The request is forwarded to issuer bank for deducting the solicited amount from a customer account. Now the merchant collects all the tokens from customers and forwards these to an acquirer bank to redeem the money that will be credited to the merchant's account. Furthermore, e-tokens are intangible and can be anonymous, so the customer does not need payment authorization from the bank for each transaction. Hence these payment methods are suitable for handling micro-payments. There is no central authority for producing and maintaining e-coins, because each company controls their ratio relationship with real money, thus preventing double spending and circulation among their partners. Examples are Fair-Cash and M-token. The related literature on token-based is [57, 111, 161, 162, 163, 164, 165, 166].

2.4.5.2 Account-based

Every customer is associated with a specific account that an "Internet payment provider (IPP)" maintains. Debit and credit are exchanged during a transaction. The customer is then periodically billed and pays the balance of the account to the IPP [20]. Owing to their high charges, traditional account-based payment methods are generally not suitable for handling transactions of very low value (i.e., micro payments).

2.4.5.3 Micro, mini, and macro payments

For this category, the payments are classified based on value and are divided into three categories, which are shown in Table. 2.6. These are: micro payments, mini payments, and macro payments.

Table 2.6: Differences between micro, mini, and macro payments

Parameter	Micro-payments	Mini-payments	Macro-payments
Value range	Under $2.	Between $2 to $20	Greater than $20
Applications	Token based payments	M-ticketing,	Credit card based, digital cash, and electronic check
Channel	Internet	Mobile, Internet and POS etc.,	Mobile, Internet and POS etc.,
Security	Off-line payment verification	Off-line payment verification	Both on-line and off-line payment verification
Literature work	[122, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178]	[179]	[180, 181, 182]

2.4.6 Based on timing of payment

Mobile payments can be performed at different times [20]. The related research based on the timing of payment is shown in Figure 2.10.

2.4.6.1 Real-time payment

In this payment method, the client uses the service in real-time or a cash-like payment schedule, which involves the funds immediately credited to the merchant account (same as cash). Electronic wallets are an example. Real-time payments can be categorized into two types one is on-line and other is an off-line payment.

- **On-line payments:** Occurs when a customer's funds are transferred to your payment account right after your customer confirms the payment. On-line customers often will pay through: (i) Mobile bank apps, (ii) PayPal, (iii) Credit cards, and (iv) Debit cards. The related literature on on-line payments is [183, 184, 185, 186, 187, 188, 189, 190, 191].

- **Off-line payments:** Indicates that money is transferred at a later date. For example, a customer deposits a check into a bank account or a payment given by phone. Applications are wire transfers, phone orders, payment to the delivery person, checks, etc. The related literature on off-line payments is [192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202].

2.4.6.2 Pre-paid payments

It is most general payment method for third-party service providers as well as mobile network operators. The customer can pay according to an amount in the debit

account. For these payment methods, the customer amount is deducted before the desired product is received; after successive purchases made by the customer, the balance of the account decreases proportionately. An alternative payment method for this pre-paid category is called pay-now, which occurs when the transaction is made directly between the payee and issuer bank (who holds the central account of the payer), for example, a money transfer from one account to another account. The related literature on pre-paid payments is [203, 204, 205].

2.4.6.3 Post-paid

The theme of post-paid payments is "to buy first and pay later". The best example of this category is credit card-based payments. The customer or a credit card holder can purchase items through online or off-line shopping, then pays the full amount at the end of the statement cycle, or they can pay interest on the outstanding balance. The related literature on post-paid is [206, 207, 208, 209, 210, 211, 212, 213].

2.4.6.4 Phone-bill

In this method, mobile payment settlement is made via phone bill for common mobile payment. An example is NTT DoCoMo's i-mode, which uses its portal to provide this type of payment method [214]. Mobile payments via phone bill have an advantage since banks, and credit card companies are excluded to some degree to simplify the process, which implies a cost reduction but requires the reliability of the mobile operators' bills. The overall distribution of articles based on their classification of mobile payment methods is shown in Figure 2.11.

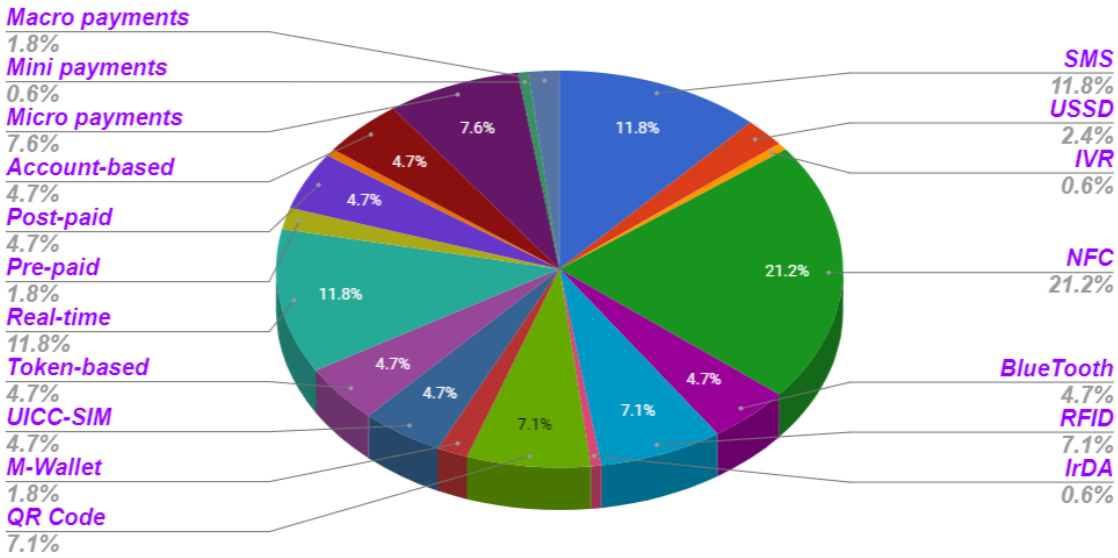


Figure 2.11: Distribution of the articles based on their classification of mobile payment channels, devices, payment value and timing of payment

2.5 Security Issues in Mobile Payments

Nowadays, mobile payments are more popular, and mobile devices are increasingly becoming targets for adversaries and hackers, so security is one of the challenging and major issues in the field of mobile payments. Because mobile payments can run under non-secure environments, trust regarding mobile commerce applications will be limited. A mobile payment application developer must know the critical issues in security and the threat models. In this survey, we consider the following parameters as security measurements for designing the mobile payment protocols and these parameters are needed for an application developer to run in the attacker-free environment such as:

2.5.0.4.1 Security, privacy and trust in MPS: The security and privacy of confidential information are one of the primary concerns in acceptance of these mobile payment systems. It plays the vital role to achieve success for the transaction security properties of confidentiality, authentication, integrity, non-repudiation, and authorization. The financial information flows between the various entities involved in MPS such as customer, bank's servers, payment gateway, and merchant are never compromised. On the other hand, trust in MPS is discussed in Chapter 1 and Section 1.1.1 as

goals and expectations of various participants. The related literature on security, privacy, and trust on MPS is in references [5, 72, 215, 216, 217, 218, 219, 220, 221, 222].

2.5.0.4.2 PKI/WPKI: A Wireless public key infrastructure (WPKI) is a system that authorizes a customer's digital identity over a wired or wireless medium. It does so by combination of public and private key pairs with the individual's identity credentials. These keys are generated with a cryptographic algorithm and shared by a certificate authority (CA) that links them to the customer's unique identity. The CA stores this information in a database and issues digital certificates, which include the public key or information about the public keys, to verify the customer's identity. M-PKI consists of the following entities: (i) User, (ii) Certificate authority, (iii) Certificate management system, (iv) PKI directory, and (v) Certificate repository. The association among these entities and how user obtains the certificate from CA as shown in Figure 2.12. The related literature on PKI/WPKI is in references [223, 224, 225, 226, 227, 228, 229, 230, 231, 232].

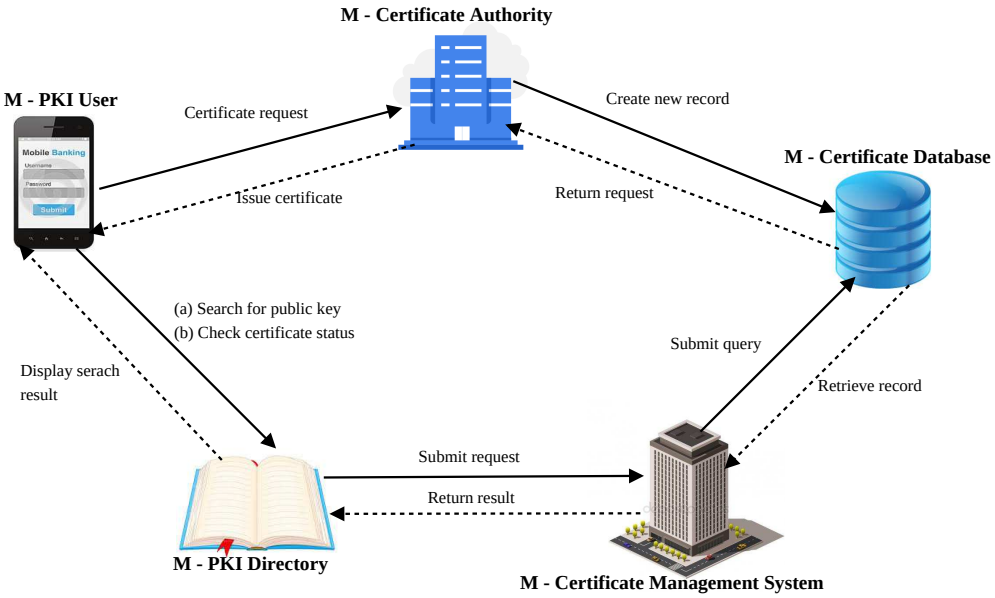


Figure 2.12: M-PKI Architecture

2.5.0.4.3 Proxy certificates: Security issues related to the usage of mobile devices in performing Transactions to which their owners have to be bound, such as payments, are of utmost importance if this kind of agents is to be used in electronic commerce. If this binding is achieved employing digital signature techniques, this means agents have to carry the owner's private key to the host where they sign documents. It exposes the key to attacks because it is copied outside a protected environment. To overcome the above key to attacks, a mechanism, called proxy certificates, that avoids the need for the agent to have access to the user's private key for digitally signing documents. The related literature on proxy certificates is in references [233, 234, 235, 236].

2.5.0.4.4 Lightweight cryptography: The primary purpose of lightweight cryptography is to provide security for devices with limited resources. These devices with limited resources implement lightweight ciphers which are reliable and require low power and low computations. The lightweight cipher should be designed with fast encryption speed and minimal use of resources. Lightweight cryptography is a cryptographic algorithm or protocol tailored for implementation in constrained environments including RFID tags, UICC/SIM, NFC payments, mobile payments, sensors, contactless smart cards, health-care devices and so on. The related literature on lightweight cryptography is in references [18, 207, 227, 237, 238, 239, 240, 241, 242].

2.5.0.4.5 Formal verification methods: Owing to the rapid growth of Internet usage, network users, and payment services, the development of security protocols has become necessary, and their formal verification is required. Designing security protocols is one of the more challenging issues; these protocols must not only support communication but also must satisfy the security goals. Because people tend to trust security protocols, the correctness and validation of these properties are highly desirable. Currently, many security experts are facing difficulties related to the strength of cryptographic algorithms employed; when designing any security protocol, it is mandatory to consider all possible behaviors of hypothetical attackers, including violations of the protocol rules, and any potential forgery of messages. In the protocol description, security attributes and intruder knowledge are defined formally. The detailed discussion of formal verification methods is described in Chap-

ter 1 and Section 1.2.3. The related literature on formal verification is in references [21, 23, 24, 25, 158, 174, 215, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252].

2.5.0.4.6 SET/iKP: The Secure Electronic Transaction (SET) protocol has been proposed by a consortium of credit card companies and software corporations to guarantee the authenticity of e-commerce transactions and the confidentiality of data. The related literature on SET/iKP is in references [4, 126, 164, 243, 245, 253, 254, 255, 256, 257, 258, 259, 260, 261].

2.5.1 Cryptography techniques in mobile payment system

Mobile payment systems must satisfy the transaction security properties [262] and [263], which are confidentiality, authentication, data integrity, authorization, non-repudiation, etc. These transaction security properties must satisfy the end-to-end security [264], and cryptography plays a central role in securing mobile payments. Cryptographic techniques are essential tools for creating a secure environment while using mobile payment systems over insecure networks and open environments. Therefore, any mobile payment system architecture design should include cryptography techniques. Table 2.7 lists the types of cryptography algorithms implemented based on technology or modes of a channel used for mobile payment systems. The related research on security issues is shown in Figure 2.13.

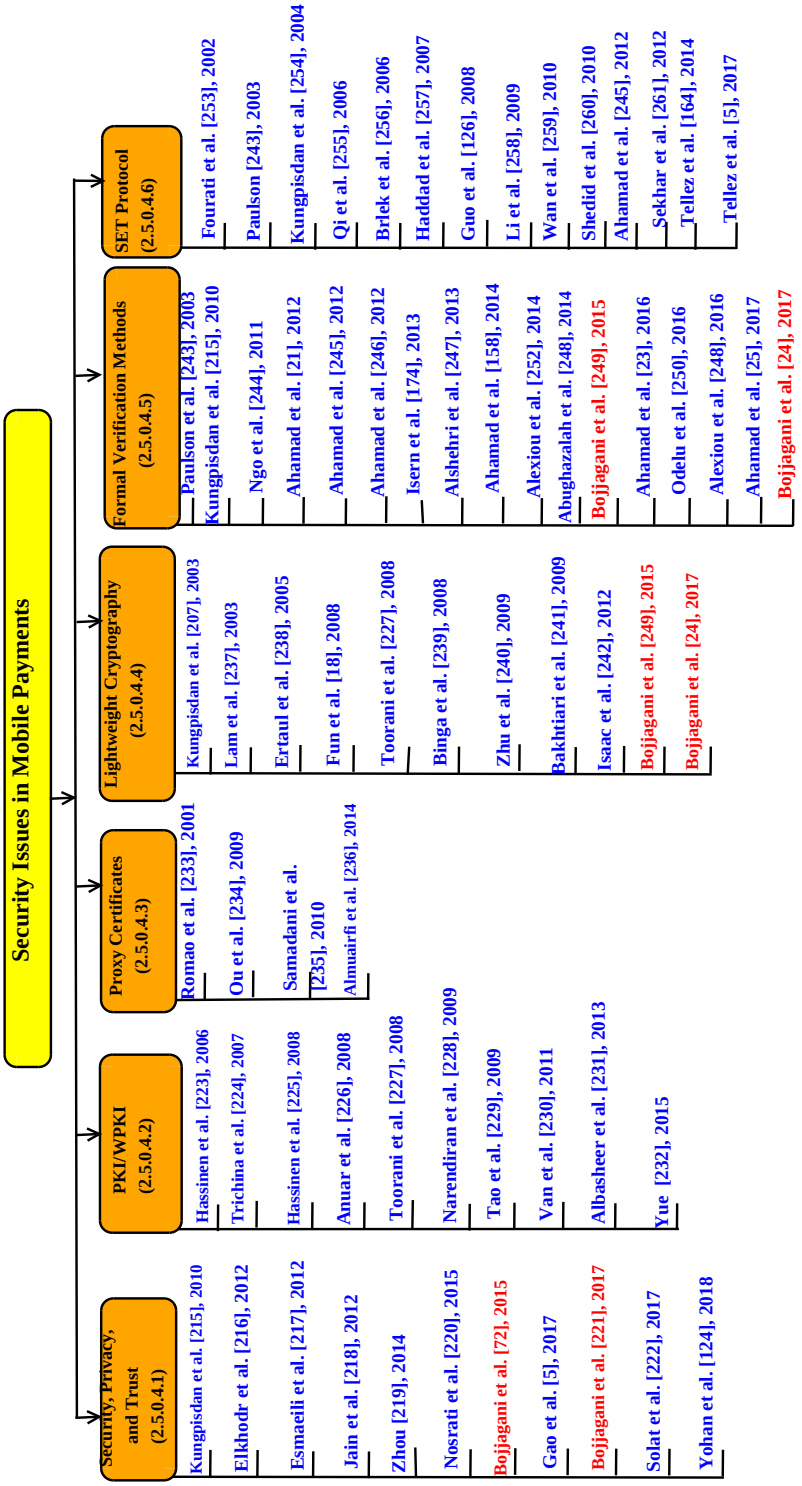


Figure 2.13: Survey of literature on security issues in mobile payments

2.5.1.1 Symmetric cryptography

For encryption and decryption of plain-text using the same key, known as symmetric cryptography or secret key cryptography, secret data or information is shared between the sender and receiver. The examples of symmetric encryption algorithms are data encryption standard (DES): key length 56 bits, Triple DES: key length 168 bits, advanced encryption standard (AES): key length 128-256 bits, international data encryption algorithm (IDEA): key length 128 bits, and Rivest cipher (RC5): key length 128–256 bits.

2.5.1.2 Asymmetric cryptography

In this type of cryptography, each party has a pair of keys, one is used for encryption of the plain-text, and the other is used for decryption of the cipher-text. In the encryption process, the sender uses the receiver's public key and sends to the receiver, then the receiver decrypts the message using his/her private key. In asymmetric cryptography, the user is authenticated person or not with the help of certificates. Some examples of asymmetric encryption are Rivest Shamir and Adleman (RSA), Elliptic Curve Cryptography (ECC), Digital Signature Standard (DSS), ElGamal, etc. Table 2.7 shows the distribution of approaches based on classification involving cryptography algorithms.

Table 2.7: Distribution of articles based on classification of cryptography algorithms

Type of cryptography	Name of the algorithm	Approches
Symmetric Key Algorithms SSMS[62], SOMP_SMS[68],SOMP1_SMS[66],SET_SPIN[258],mFerio[111],Tellez[4], Kungpisdan[254] Tellez_Secure[164],Tellez_Mobile[166],UICC_Park[157],On-line_Sign[186],E-Cash_Anand[183],Lightweight[207] Micropayment_Chaos[170]	DES	SET[260],UICC_Aegis[153]
	Triple DES	SecureSMSPay[60],UICC_Zhao[155],UICC_Aegis[153]
	AES	SMSSec[63],SMS[73],NFC_mCoupons[103], Dhwani[116],NFC_IDA-Pay[114],UICC_Zhao[155],UICC_Aegis[153]
	RC5	RFID_Bill[133]
	Blowfish (cipher)	SafeSMS[58]
Asymmetric Key Algorithms QR[148],SET[253],SET_Verify[243],SET_Flaw[256],NFC_Health[95],NFC_mCoupons[103],RFID_SIM[127],Tellez[4] Tellez_Secure[164],fairCASH[161],Tellez_Mobile[166],UICC_Alimi[154],UICC_Park[157],E-Cash_Anand[183] Credit_Trans[210],Lightweight[207],Micropayment_Off-line[169]	RSA	SMS_Encrypt[61],SMS_Security[67],SEESMS[65],SMSSec[63],SecureSMS[75],SET[260],SSMCP[257] NFC_Key[101],NFC_IDA-Pay[114], RFID-SIM1[129],Account_Sekhar[261],UICC_Aegis[153],E-cash[189] E-cash_Trapdoor[184] E-cash_Scheme[187],Off-line_E-cash[195],Improved_Off-line[196],Offline_Date[198] Offline_Cryptanalysis[199],Off-line_Feng[201],E-Cash_Revoking[200],Micropayment[172]
	ECC	SMS_Bank[24], SecureSMS[59], Ssmbp[249],SMS_Security[67],SSMS[62],SEESMS[65],SSMCP[257] WSETP[255],Bluetooth_local[119], Bluetooth_Micro[122],UICC_Ahamad[158], ElectronicCash_[191],E-cash_Pair[190], E-Cash_Novel[194],RO-cash[192],CreditPayment_Secure[212]
	DSS	SMSMobBank[24],SEESMS[65], DS_SMS[69],SET_Formal[245],Super_SET[259],NFC_Cloud[102] NFC_Biometric[23],Bluetooth_local[119], UICC_Ahamad[21],E-payment[188],MobileBank_Pay[185], Creditcard_On-line[206]
	ElGamal	SMS_Security[67],Untraceable_Off-line[193],Off-line_Robust[202]
	Diffie-Hellman	QR_Secure[149],Bluetooth_Secure[120],E-Cash_Novel[194],CreditPayment_Secure[212]
Other Approaches Visual Cryptography Scheme (VCS); Visual Secret Sharing (VSS); A5/1 Algorithm; Formal Verification: Model Checking, BAN Logic;	Miscellaneous	MEPS[71],SMS[64],SMSPAY[76],SMPP_SMS[74],USSD[80],Survey[79],Access[77],QR_VCS[150] QR_Secure[143],QR_Web[147],QR_Mob[144],IVR[81],M-wallet[70],M-wallet1[151],NFC_Alzheimer's[85],NFC_Shop[88] NFC_Social[90],NFC_IoT[98],NFC_Exploring[91],NFC_Maintenance[93],NFC_Mobile[92],NFC_Supply[87] NFC_Touch[94],NFC_LocaTag[99],NFC_Clinical[96],NFC_Home[83],NFC_Data[89],NFC_Multi[84] NFC_Wiki[97],NFC_Application[82] NFC_AAL[100],NFC_TaggyNet[106],NCASH[108], NFC_Ticketing[104] NFC_Higher[107],NFC_Tapango[109],NFC_SmartTouch[105],NFC_Auto[110],NFC_Multi[112],NFC_Relay[113] NFC_Credit[115],Bluetooth_Mobile[118],Short-range[117],Bluetooth_Beacon[123],Paytooth[121],RFID[131], RFID_Smart[128],RFID_QUICK[135],RFID_Study[132],RFID_Commerce[126], RFID_Inventory[134] IrFM[137],SET_Guo[57],Wallet_Token[165], Explore[162],Adoption[163],UICC_Secure[156], Debitcard_Use[203] Prepaid_Mobile[204],CreditCard_One-Time[208],CreditCard_DSC[213],Creditcard_Naji[211],CreditCard[209]

2.5.1.3 Other approaches

In this category we have identified the approaches as shown in Table 2.7 using different schemes for encryption and decryption, such as the following:

• **Visual Cryptography:** Visual cryptographic scheme (VCS) is a generalization of secret sharing and was first proposed by Naor and Shamir in 1995 [265]. An original halftone image is divided into n shared images and each shared image is printed on a transparent film, such that any films stacked together can restore the original image. However, for the case of fewer than k films, the original image cannot be restored. This method mainly uses the human visual system of color approximation principle. Therefore, the secret image restoration can be implemented by a simple film superposition without any password requirement [150].

• **Visual Secret Sharing (VSS) [213]:** VSS is a technique that hides the secret first and then partitions it to multi-parties for sharing purposes. This technique can be dated back to 1979, when Shamir and Blakley provided two different (t, n) secret sharing mechanisms, respectively. This technique first divides a secret into n parts and partitions the, n parts to n individuals. When $(k \geq t)$ individuals stack their parts together, the original secret will be disclosed. Compared with traditional secret sharing, VSS makes use of visual information instead of complex algorithms to improve applicability and efficiency.

• **Model Checking:** It is a promising mathematical method that can automatically verify some properties of the model of a system. This method makes it possible to prove the correctness of systems by using temporal logics to specify system properties. For model checking verification, we need an automata model to demonstrate the sequential change of a system's processes from one state to another under the action of transition [64]. Apart from this, the process of automata will form the basis of the operational models that have been used to specify the behavior of systems to be validated.

• **A5 Algorithm:** The security issues that exist in the GSM are mainly cryptographic issues related to encryption and authentication algorithms. The A5 algorithm commonly used for encryption in GSM has been reverse-engineered [80] and the A3/A8 authentication algorithm has been demonstrated to contain several flaws that make it vulnerable to attack. These security vulnerabilities in the GSM make the technologies that utilize SMS services susceptible to attacks if they do not properly protect

their data. Security issues in mobile transactions through GSM services include SIM attacks and SMS attacks.

2.5.2 Security threats in mobile payments

Many mobile payment systems lack the security and transaction properties discussed in Section 2.5. As a result, mobile devices and payment systems are increasingly becoming targets for attackers. Most of the mobile payments use the same mode of application based on communication technologies such as SMS, GSM, Bluetooth, NFC, RFID, etc., and security concerns are usually ignored while developing these apps. As a developer of mobile payment app, one should strictly focus on security aspects [266] during design and implementation. Figure 2.14 depicts the that taxonomy of vulnerabilities affecting an m-payment system.

Figure 2.14: Taxonomy of vulnerabilities affecting an M-payment system

2.5.2.1 Threat, risk and vulnerability

In this section, we describe how these vulnerabilities, attacks, risks, and threats can create damage to a supposedly secure system. To elaborate on these threats, first, we clarify various definitions [138].

- **Vulnerability:** Is a weakness in a system which allows an adversary to conduct malicious actions that threaten its information assurance.
- **Threat:** Is a possible danger that may allow an unfair benefit to the attacker who can cause harm by making use of the vulnerability.
- **Risk:** An adversary may cause severe damage to a system (potential harm) that may arise after the realization that a threat is, in fact, a risk.

2.5.2.2 Categories of threats

The threats in mobile payment systems are classified into three categories [221] such as (i) Application level threats, (ii) Communication level threats, and (iii) Device level threats. The more detail on these various level threats is explained in Chapter 5.

- **Application level threats:** These threats are mainly caused by the app developer. During designing and coding of the mobile payment application, the developer should concentrate on secure code practices.
- **Communication level threats:** While payment transactions are communicating with web servers, adversaries may exploit vulnerabilities to capture confidential information. These risks may occur because of: compromised access points and routers, monitored Wi-Fi and other network devices of cell towers, proxies, etc.
- **Device level threats:** In this case, the adversary analyzes how and where confidential information is stored in on mobile file systems and artifacts.

2.6 Analysis of SLR Results

In this section, we analyze our research questions and map them into three categories (i) mobile payment, (ii) mobile payment stakeholders, and (iii) security infrastructure and goals. In each category, we identify the research gaps and focus our analysis on associated articles. For example, in the stakeholders' phase, we have addressed the following topics: consumer adoption of mobile payments, and the study of strategic issues in multi-perspective frameworks and ecosystems.

2.6.1 Distribution of articles based on mobile payment stakeholders

Mobile payment adoption criteria and participants involved in mobile payments have traditionally attracted much awareness in academia. Additionally, these studies have included slight contributions to the understanding of the problems occurred that have hindered the developments of mobile payments over the years. The study addresses why payment initiative studies fail to reach remote places, merchants, and consumers. The recent research on multilevel mobile payment platforms at the ecosystem level address the research questions of RQ2 and RQ3, mentioned in Section 2.1. Table 2.8 provides the mobile payment stakeholders research results. In this phase we classify the research topics based on the themes of articles in four areas:

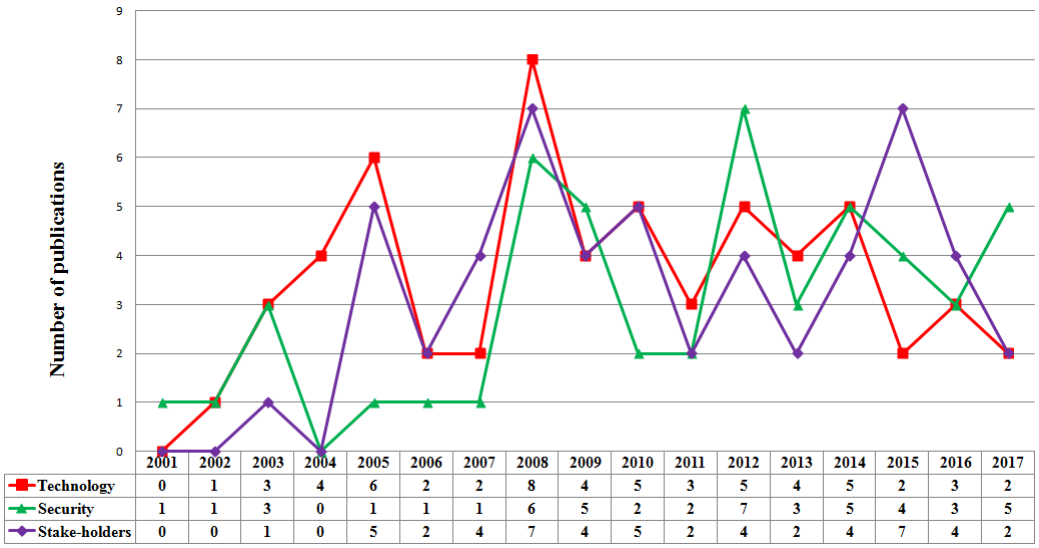
- Adoption of mobile payments
- Stakeholders in mobile payments
- Study of strategic issues
- Multi-perspective framework for mobile payment systems

Ondrus et al. [267] categorized mobile payments in to three perspectives: (i) Market, (ii) Actors, and (iii) Issues.

- Market: In this perspective, the market identifies a particular need of consumers and merchants. It investigates the ability to build and maintain a profitable relationship with customers.

- Actor: This perspective examines the roles and expectations of the stakeholders, the structure of the ecosystem, and the business and competitive conditions.
- Issues: This perspective deals with various uncertainties related to the future scope of mobile payments, for example. mobile payment sizes and physical forms of devices.

Zmijewska et al. [268] proposed a "Reshaping the framework for analyzing the success of mobile payment solutions". This framework contains mainly two perspectives: one is focused on customer and merchant adoption of mobile payments, and the other is an infrastructure perspective. For infrastructures, the author focuses on stakeholder collaboration, business models, and regulation issues, and it is claimed that both perspectives are equally important and depend strongly on each other.
In 2008, the number of articles on mobile payment technology or multi-perspective frameworks for mobile payment systems has moderately increased as shown in Figure 2.15.



	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Technology	0	1	3	4	6	2	2	8	4	5	3	5	4	5	2	3	2
Security	1	1	3	0	1	1	1	6	5	2	2	7	3	5	4	3	5
Stake-holders	0	0	1	0	5	2	4	7	4	5	2	4	2	4	7	4	2

Figure 2.15: Progress of mobile payment research year-wise w.r.t. technology, security, and stake-holders category

Tomi Dahlberg et al. [269] published an article called "Lost opportunity why has dominant design failed to emerge for the mobile payment services market in Finland?".

Their theoretical framework was constructed by combining the results of earlier research on the dominant designs of inter- and intra- business organizations and the findings of the finished case. The article also discusses standardization of research, and identifies the key players or actors involved in the firm and outlines the standards for users as well as developers. Yoris et al. [7] developed a framework to determine the appropriate stakeholders of mobile payment ecosystems and applicable theory based on their various roles, through analysis of the market, consumers, merchants, business processes, technology producers, and regulators. Ondrus et al. [270] applied various theories and levels of analysis to Swiss initiatives that had failed. This dynamic analysis provided insight into why mobile payment systems had failed and provided future directions for mobile payment diffusion. Hedman et al. [271] described how digital payment innovations influence the payment ecosystem and discussed new trends for digitalization. Their conclusion was that "The digitalization creates a new arena for competition that will require new collaboration forms among involved stockholders". Kazan et al. [272] constructed a framework to analyze the strategies for mobile payment ecosystems of current market actors. In their work, they analyzed three factors: homing and switching costs, network effects, and creating viable mobile payment platforms as essential factors.
Gannamaneni et al. [273], studied how to "unveil factors explaining the failure of past mobile payment platforms". They identified reasons for failure in four cases of mobile payments platforms: South Korea (Moneta), Switzerland (Postfinance), Spain (Mobipay), and Germany (Paybox) happening at different times. As a result, they observed that these cases share many of the same failure factors:

- Lack of technology standard
- Little or no collaboration among stakeholders
- Low value-added for merchants and consumers compared to existing payment solutions

In summary, various multi-perspective frameworks for mobile payment systems have evolved over the decades. Figure 2.16 depicts a pie chart showing the number of articles as well as the percentage of the pool of relevant publications targeting a research

topic, i.e., distribution of the articles based on their mobile payment stakeholders, constituting 30% relevant publications on adoption of mobile payments, 30% on stakeholders in mobile payments, and 25% of the relevant papers have studied strategic issues, whereas 15% of the relevant articles use multi-perspective frameworks for mobile payment systems.

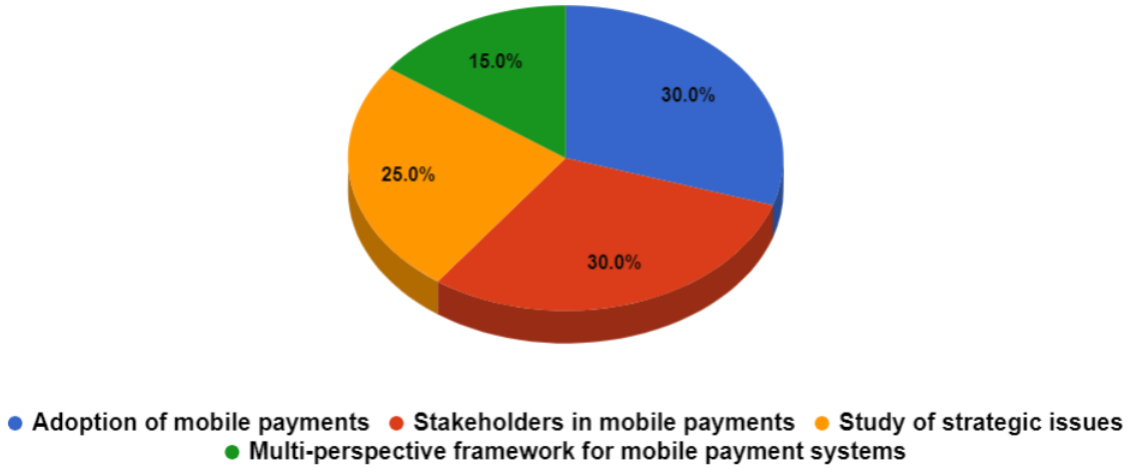


Figure 2.16: Distribution of the survey publications-based on the mobile payment stakeholders

2.6.2 Distribution of articles based on technologies in mobile payments

The technology-based articles for mobile payments have a close relationship with the security aspects. Most of the relevant publications on technology do not address security concerns, and for this reason, the corresponding publications on technology have become increasingly one-sided or unidimensional. However, we address the issues in mobile payment security domain in the next subsection. Recent research on technology and technology environment addresses the research question of RQ1, mentioned in Section 2 of this thesis. The relevant publications on the classification of mobile payment systems is provided in Figures 2.8, 2.9, 2.10, and 2.13. The Table 2.8 suggests the research gaps we identified during the survey and addresses the research questions

mentioned in Section 2.1. For the technology environment we mainly focus on the following six areas listed below:

- Analysis of mobile payment models
- Approaches to mobile payment systems
- Approaches to protocols for secure mobile payment transactions
- Approaches to tools or mechanisms for mobile payment transactions
- Approaches to biometric security in mobile payments
- Approaches to payment architectures and technological environments

Antovski et al. [274] described a framework for mobile payments technology, supported protocols, and the software environment. They primarily focused on what factors influence the introduction of successful mobile payment systems, and with these factors, they evaluate whether J2ME technology is suitable for developing mobile payment systems. Kungpisdan et al.[254] proposed a secure account-based payment protocol for wireless networks. In their approach, the protocol employs cryptography operations based on the symmetric key. It involves lightweight security and requires lower computation for all engaging entities than existing payment protocols. The protocol also formally verified the means to achieve the goals of payment protocol and how to satisfy the transaction security properties provided by public-key based payment protocols, such as secure electronic transaction (SET) protocol and internet keyed payments (iKP). Gao et al.[275] proposed a peer-to-peer wireless payment system, known as P2P-Paid. The P2P-Paid system provides both online and mobile peer-to-peer payment functions. They provided a P2P-Paid system overview on system functional features, system architecture, and applied technologies. In 2008, the number of relevant publications on mobile payment technology and technology environment frameworks for mobile payment systems had moderately increased as shown in Figure 2.15. Godbole et al.[276] described a secure and efficient protocol for mobile payments. Conducting electronic payments on mobile devices is a difficult task because of lack of interoperability, standardized protocols, and security aspects of developing a mobile payment infrastructure. They analyzed the proposed protocol and showed it to be more efficient

than other schemes, and argued that is especially applicable for micropayments in SEcure MObile Payment Service (SEMOPS).

Ahamad et al. [22, 23] developed frameworks on biometric-based mobile payment protocols. Their research was built on UICC, the payments originating from the mobile application to the bank web server. Yang et al. [277] covered issues on biometric templates with security protection and how to enable seamless integration with existing password-based payment systems. In summary, various technologies for mobile payments have been proposed over the decades. Figure 2.17 depicts a pie chart showing the number of articles as well as the percentage of the pool of relevant articles targeting a research topic i.e., distribution of the articles based on their technology in mobile payments: 8.2% on analysis of mobile payment models, 24.6% on approaches to mobile payment systems, 14.8% on approaches to protocols for secure mobile payment transactions, and 16.4% of the relevant publications employed approaches on tools or mechanisms for mobile payment transactions, whereas 11.5% of the relevant publications employ approaches on biometric security in mobile payments, and finally, 24.5% on approaches to payment architectures and technological environments.

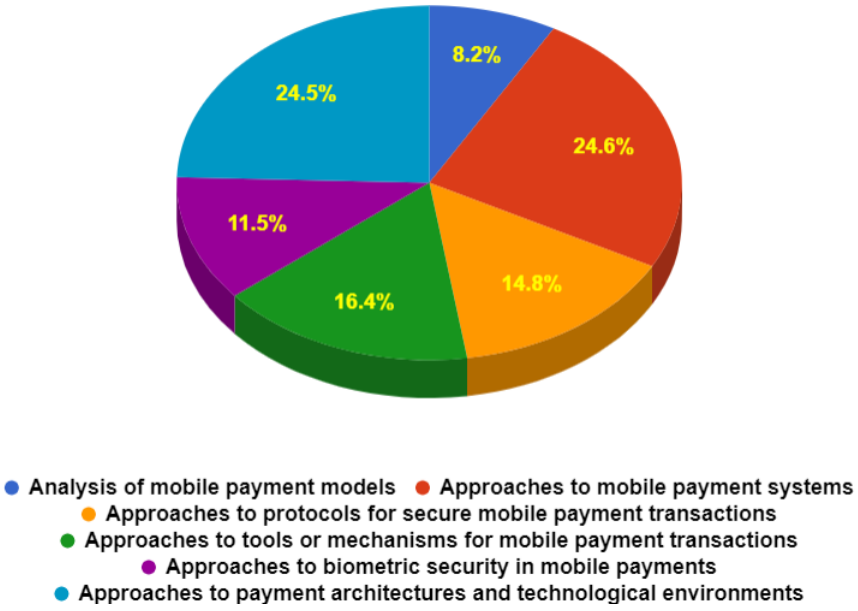


Figure 2.17: Distribution of the survey publications-based on the technologies in mobile payments

2.6.3 Distribution of articles based on security issues in mobile payments

Lack of security and transaction properties in mobile payments will cause severe limits to m-commerce applications. In this section, we mainly focus on various approaches to security, which address RQ3 and RQ4 mentioned in Section 2.1. Table 2.8 lists security issues in the mobile payments phase. In this phase we classify the research topics based on the themes of articles into five areas:

- Approaches to security, privacy, and trust for mobile payments
- Approaches to PKI/WPKI-based mobile payments
- Approaches to proxy certificates in mobile payments
- Approaches to lightweight cryptography
- Approaches to formal verification of protocols, architectures, and tools for mobile payments and mobile banking

Table 2.8: Gaps identified in three phases of literature survey on mobile payment systems

Name of the phases	Name of the research topic	Research works
Mobile payment stakeholders	Adoption of mobile payments	[5, 6, 9, 52, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291]
	Stakeholders in mobile payments	[5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 287, 292, 293, 294, 295]
	Study of strategic issues	[7, 9, 12, 13, 17, 225, 272, 273, 296, 297, 298, 299, 300, 301, 302]
	Multi-perspective framework for mobile payment systems	[7, 13, 267, 268, 269, 270, 271, 303, 304, 305, 306]
Technologies in mobile payments	Analysis of mobile payment models	[10, 14, 294, 305, 307]
	Approaches on mobile payment systems	[164, 180, 274, 275, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318]
	Approaches on protocols for secure mobile payment transactions	[207, 254, 261, 276, 319, 320, 321, 322, 323, 324]
	Approaches on tools or mechanisms for mobile payment transactions	[237, 302, 325, 326, 327, 328, 329, 330, 331, 332]
	Approaches on biometrics using in mobile payments	[22, 23, 211, 277, 333, 334, 335, 336]
	Approaches on payment architectures and technological environment	[225, 242, 296, 324, 325, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346]
Security issues in mobile payments	Approaches on Security, privacy, and trust for mobile payments	[5, 72, 215, 216, 217, 218, 219, 220, 221, 222]
	Approaches on PKI/WPKI based mobile payments	[223, 224, 225, 226, 227, 228, 229, 230, 231, 232]
	Approaches on proxy certificates in mobile payments	[233, 234, 235, 236]
	Approaches on lightweight cryptography	[18, 207, 227, 237, 238, 239, 240, 241, 242]
	Formal verification & validation of protocols, architectures, & tools	[21, 23, 24, 25, 158, 174, 215, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252]

In summary, various security issues in mobile payments have evolved over the decades. Figure 2.18 depicts a pie chart showing the number of articles as well as the percentage of the pool of relevant articles targeting a research topic, i.e., distribution of the publications based on security issues in mobile payments: 20.0% on approaches to security and trust for mobile payments, 20.0% on PKI/WPKI-based mobile payments, 8.0% on proxy certificates in mobile payments, 18.0% of the relevant publications have employed approaches to lightweight cryptography, whereas 34.0% employ approaches to formal verification of protocols, architectures, and tools for mobile payments and mobile banking. The overall distribution of articles based on three categories is shown in Figure 2.19.

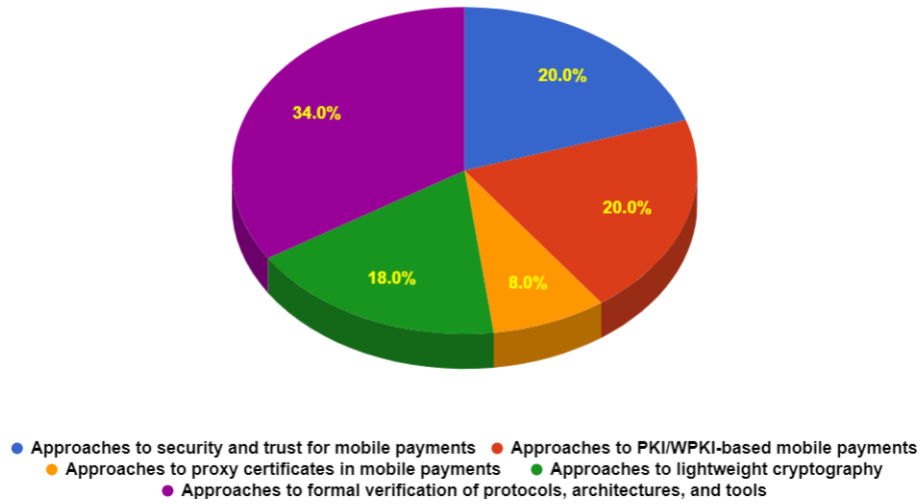


Figure 2.18: Distribution of the survey publications-based on the security issues in mobile payments

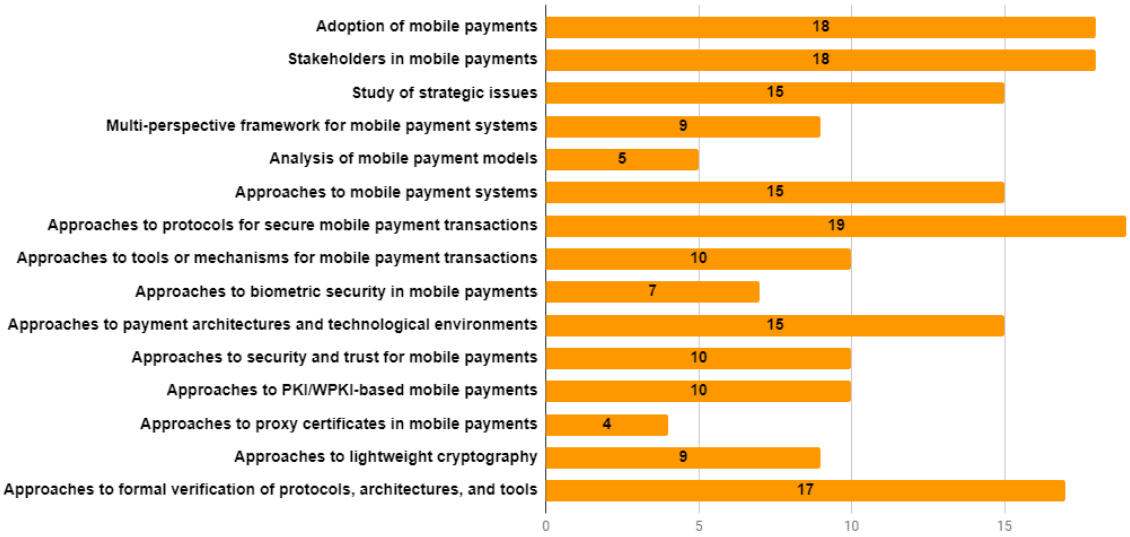


Figure 2.19: Overall distribution of relevant publications according to the research domain classification

2.7 Discussion and Future Directions

After analyzing the information collected through this SLR for mobile payment research, we present the following discussions for each phase.

2.7.1 Discussion on stakeholders in mobile payments

Dahlberg et al. [302, 347] suggested various recommendations for mobile payment adoptions from the perspective of the consumer for the successful development of mobile services and to indicate the contextual meaning of adoption factors. There are specific gaps finding adoption, in fulfilling the theoretical basis for adoption, and conducting empirical data collection in real-life payment scenarios. The approaches to mobile payment implementation use software, type of algorithm, the platform, security, and trust mechanisms described. However, these approaches typically do not include adoption research to validate the preferences or exceptions of stakeholders. This indicates that security and trust mechanisms should consider adoption research.

2.7.2 Discussion on technology in mobile payments

- To increase the performance of mobile payment transactions most of the approaches use elliptic curve cryptography (ECC). These involve lightweight cryptography and provide higher security with lower overhead.

- The bring your own device (BYOD) concept refers to the policy of allowing organizational-employees to bring their own devices (smartphones, laptops, tablets, etc.) in the workplace. Use of the devices to access privileged organizational data and applications for improving productivity gains and cost benefits enables employees to use their technology in the workplace, but increases security risks [166].

2.7.3 Discussion on security in mobile payments

- Generally, mobile payment transactions operate in the client server environment, the data traverse the mobile device's carrier network (Wi-Fi access point) and the Internet. An adversary initiates attack to intercept or capture confidential information while it is being transmitted.

- As mobile payment applications generally do not consider security requirements such as confidentiality, data integrity, non-repudiation, authorization, etc., the company or trusted party who conducts payments using mobile applications must regularly verify the security requirements or conduct vulnerability assessment in depth as proposed in this thesis.

2.7.4 Benefits for practitioners and researchers

This SLR classifies approaches to new technologies, requirements, expectations of stakeholders, and security issues involved in mobile payments. The relevant publications on the areas mentioned above provided a reasonable amount of information, and reduced the time required to search for relevant studies and solutions. We present a detailed analysis of the mobile payment research work, which should be useful to practitioners, researchers, mobile payment application developers, security testers, third-party organizations, bankers, and consumers interested in the mobile payment field.

2.8 Summary

The primary objective of this systematic literature survey was to obtain a holistic view of the state-of-the-art research regarding stakeholders, technologies, and security in the mobile payments context. In this study, we have identified 316 primary studies, which cover a spectrum of research approaches varying in artifacts, terminology, descriptions, and activities. Finally, beyond these differences, we carefully analyzed approaches and eliminated redundant topics, e.g., application context, focus, and goals. We extracted approaches based on research challenges and summarized our study into three main categories: stakeholders, technology classification, and security issues in mobile payments. These primary studies have contributed with technology frameworks (28%), protocols (12%), methods (8%), tools (7%), approaches (4%), and others (4%).
This chapter presented a systematic literature review to classify and characterize research efforts in this area. In this study, we have identified various gaps in each phase that could be used to improve the criteria for mobile payments, and provides recommendations that will shape the focus of future research efforts.

Chapter 3

Secure End-to-End SMS-based Mobile Banking Protocol

The short message service (SMS) provides a channel of communication for banking in m-commerce and mobile payments. Kungpisdan et al. [254] proposed a model of an account-based mobile payment protocol, using symmetric-key approach, and it requires a lot of computations for key generation between client and merchant. Hassinen et al. [58] developed SafeSMS-end-to-end encryption for SMS messages. In his proposed method, the secret password is shared between the sender and receiver using application. The application stores messages sent and received and maintains records of name, phone number, and the password for each contact. If an adversary stole the device, then all sensitive information is disclosed. Hashemi et al. [355] have proposed a protocol for secure m-payment in mobile devices; the protocol explains the features of the interaction between parties such as SMS gateway and SMS center. The entities are dealing with m-payment transactions that are secured by using advanced encryption standard; it is used for sharing the keys between the customer and the bank. However, these keys are distributed when the customer registers the service for the first time, but session key for transaction updates was not discussed in his method. Toorani et al. [62] proposed "Secure SMS messaging protocol for the M-payment systems", by ECC that ensures the properties of the cryptography such as message authentication, integrity, and confidentiality.

Narendrian et al.[356] developed a framework for mobile banking protocol using X.509, but this protocol could not protect from replay and man-in-middle attacks.

Mary et al. [67] develops SMS security: an asymmetric encryption approach. Which evaluates various encryption techniques such as RSA, ElGamal, and elliptic curve with the inputs of random SMS messages of various sizes for analyzing their encryption and decryption time. Alfredo et al. [65] proposed a framework for efficient, secure SMS; it is an application-based framework running on the platform of Java. The messages are exchanged between two peers in a network and encryption of communication channel is done by the ECIES and RSA algorithms. But this paper cannot address the problems associated with the fund transfer between customers.

3.1 Introduction

In the "Global System for Mobile communications (GSM)", the mobile operator provides communication between the Mobile Station (MS) and the Base Transceiver Station (BTS) for transmission of data, which is optionally encrypted with weak A5 stream cipher vulnerable against rainbow attack. The data is stored as plain text at "Short Message Service Center (SMSC)" [59] of the mobile operator. The attacker can use duplicate BTS to pursue "Man-in-the-Middle (MitM)" attack, as there is no mechanism of mutual authentication presently in mobile network.

Security issues in SMS banking

1. The attackers can intercept the mobile network traffic and capture the unencrypted SMS packet to sniff or acquire sensitive information of users and can tamper information by altering the receiver's (or beneficiary) data with its own.

2. An adversary can perform replay attack on captured SMS message without the knowledge of the user, which can lead to an unintended transaction.

3. Sent SMSs are stored in an outbox of the mobile device of the customer which can reveal MPIN into a user with unauthorized access.

4. The customer SMSs are stored at SMSC of the mobile telecom operator and third-party SMS gateway before forwarding to a bank, which enables them to track or identify user activities and reveal sensitive information to illegitimate personnel.

The goal of the chapter is to resolve problems identified in existing works and the security threats which are discussed above to provide end-to-end application layer security features and feasible services that bear as simple as possible. Also, security services should not decrease the performance of applications involving resource utilization, memory, and bandwidth limitations. A "transaction level security (TLS)" must ensure an end-to-end mechanism providing confidentiality, integrity, and non-repudiation. To provide end-to-end security for mobile devices, our proposed framework supports WPKI.
To achieve end-to-end security with fewer resources consumption, (i.e., regarding less communication and computation overheads) is to provide a secure lightweight using ECC that is fit in the current existing SMS infrastructure.

3.1.1 Organization of chapter

The remaining sections of this chapter are organized as follows. Section 3.2 describes the proposed protocol. Section 3.3 discusses the security analysis of the proposed approach and formal verification results of the SMS-based mobile banking protocol. Section 3.4 provides the performance analysis and a comparative analysis of the current state of the art. Section 3.5 describes the implementation, and concluding remarks are provided in Section 3.6.

3.1.2 Contributions

In this chapter, we introduce a new SMS-based mobile banking protocol with formal verification. The proposed framework allows a payer to make payment to a bank through an encrypted format. This protocol allows digital signatures, hash functions, and symmetric and asymmetric algorithms between the participating entities. It provides a bridge between payer and payee via a payment gateway using the concept of an elliptic curve combining digital signature and encryption functions in mobile devices. This protocol is free from all possible attacks such as Man-in-the-Middle [348], replay, and reflection attacks as verified under the environments of the AVISPA and Scyther tools. Furthermore, the proposed SMS-based mobile banking protocol is compatible with existing SMS message infrastructures.

3.2 Proposed scheme

This chapter proposes a secure mobile banking protocol based on SMS. This framework uses the ECIES encryption mechanism and ECDSA for digital signatures. The model uses a mobile application as a component in the design of the proposed model. Some essential components are used in the model for encryption and signing, which are as follows:

- ECDSA: It is used for digital signature of a message. The signature generation and verification of the ECDSA algorithm available in Appendix B.4.
- ECIES: SMS messages are encrypted with public key and decryption using private key using the ECIES algorithm which is addressed in Appendix B.6.
- AES: It is a symmetric key for encryption and decryption of messages.

The notations throughout this chapter are listed in Table 1.

3.2.1 Participants in SMS banking and initial conditions

The participants involved in the SMS-based mobile banking protocol are the Payer or Remitter (PR), Payee or Beneficiary (PE), Payment Gateway (PG), Issuing bank (IB), Acquiring bank (AB), Mobile Network Operators (MNO), and SMS gateways (SMSC). In the current existing SMS banking infrastructure, the user uses a mobile device to send and create SMSs to the bank's target address over MNO's. The MNO route and forward these messages to SMS gateway to the respective banks. Bank collects these SMSs for delivery to the bank server through a leased line "hypertext transfer protocol secure (HTTPs)" connection. The PG provides a secure network for interbank transactions as shown in Figure 3.1.

Initial conditions

- Both PR and PE are registered customers for SMS banking.
- Every PR will have his/her own certificate provided by a certification authority (CA).

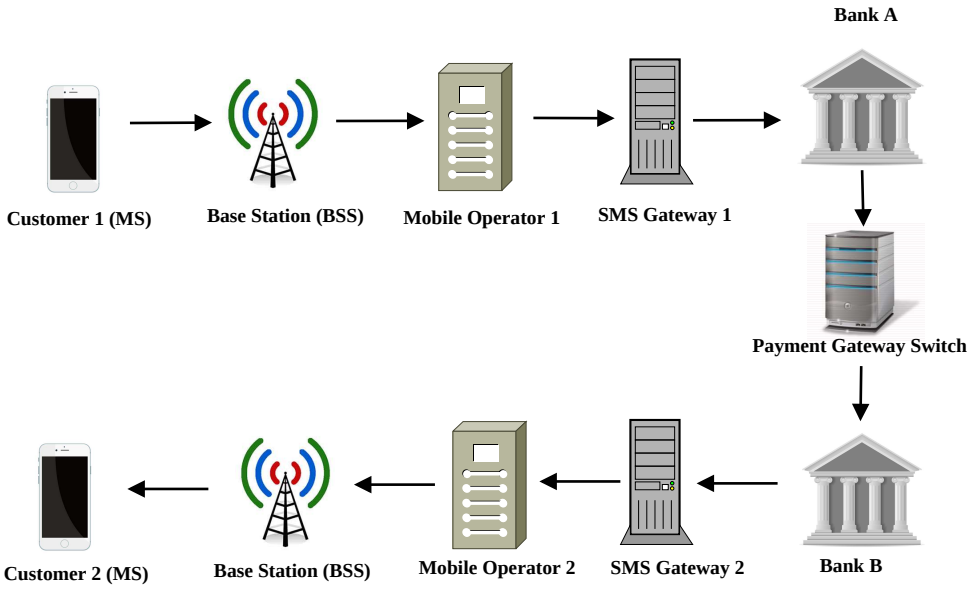


Figure 3.1: SMS banking architecture

- Each SMS-based mobile bank payment application will have its own WPKI certificate issued and maintained by the CA.
- Participants involved in the SMS protocol possess public keys and their own certificates

Payer (PR) and Payee (PE) registration

- PR & PE registered users for SMS-based mobile banking service with the IB and AB respectively.
- PR receives a "mobile money identifier (MMID)" and MPIN from the IB.
- PE link his/her mobile number with their account in the AB.
- PE receives the MMID from the AB.
- Use the SMS service for mobile banking.

3.2.2 Detailed protocol

The SMS-based protocol consists of three phases such as (i) Authentication phase, (ii) Transaction phase, and (iii) Notification phase as described in the following text.

(i). Authentication phase

Before initiating a transaction, the payer or remitter must be registered with the issuing bank (IB), then open the SMS bank application and send the signature information encrypted with issuing bank's public key. The IB receives the message and decrypts it with the private key, then verifies the signature generated by the payer using his private key, as shown in Table 3.1. The sequence of steps is described as follows.

Step 1 : $Payer\ (PR) \rightarrow Issuer\ Bank\ (IB) : \left\{M1, SIGN_{PR}^{IB}(M1)\right\} K_{IB}^{+}$,
where $M1 = \{TS_{PR}, N_{PR}, BI, ID_{PR}\}$

Step 2 : $Issuer\ Bank\ (IB) \rightarrow Payer\ (PR) \left\{\ M2, SIGN_{IB}^{PR}(M2)\right\} K_{PR}^{+}$,
where $M2 = \{SK_{ib}, ID_{PR}, N_{PR}, N_{IB}, TS_{IB}, TS_{PR}\}$

Step 3 : $Payer\ (PR) \rightarrow Issuer\ Bank\ (IB) : \left\{M3, SIGN_{PR}^{IB}(M3)\right\} SK_{ib}$,
where $M3 = \{PI, TS_{IB}, N_{IB}\}$

The transaction flow between entities is shown in Figure 3.2. The sequence of steps is shown in Figure 3.3 and the above steps are represented in Algorithm 1.

(ii). Transaction phase

In this phase, once the IB validate the PR data, the PR sends payment information (PI) to the IB. The IB routes the transaction to a PG through a secure bank network as shown in Table 3.2. The sequence of steps is described as follows.

Step 4 :$Issuer\ Bank\ (IB) \rightarrow Payment\ Gateway\ (PG)\ \{MAC0, M4)\ \}\ SK_{ip}$
where $MAC0 = \{hash(SK_{ip}, M4)\}$ where $M4 = \{PI, BI, SK_{ip}, N_{IB}, TS_{IB}\}$
(a) Check $BAL_{PR}\ (M1_{PR})$ for low balance
if $BAL_{PR} < Amt_{PR}$ send SMS_B ("Balance is Low");
end if
else
(b) Transaction is routed through secure network.
exit.

(iii). Notification phase

In this phase the participants (PR, PE) receive notification messages regarding the payment transaction status.

Step 5 : $Payment\ Gateway\ (PG) \rightarrow Acquirer\ Bank\ (AB)\ \{MAC1, M5\ \}\ SK_{pa}$
where $MAC1 = \{hash(SK_{pa}, M5)\}\ , M5 = \{(PI, BI, SK_{pa}, N_{PG}, TS_{PG})\}$
/*In this step the PG sends a message to the AB. if all verifications are successful, the amount is debited from payer's account, and an amount is credited to the payee's account, and this payment successful information is transmitted to both payer and payee by their respective banks. Otherwise, the bank sends an alert message.*/
if (verification hash $(SK_{pa}, M1))$
(a) SHOW ("Transaction completed with transaction ID") display on PR mobile screen.
else
(b) PR receive transaction unsuccessful as alert message ALERT ("verification problem");
Step 6 : $Acquirer\ Bank\ (AB) \rightarrow Payee\ (PE) : PE_{receive} SMS_{AB}$
display message $SMS_{AB}\ (PI_{PR}, ID_{PE}, TID)$
In this step, the acquirer bank sends a successful payment notification through SMS to the payee with the transaction ID.

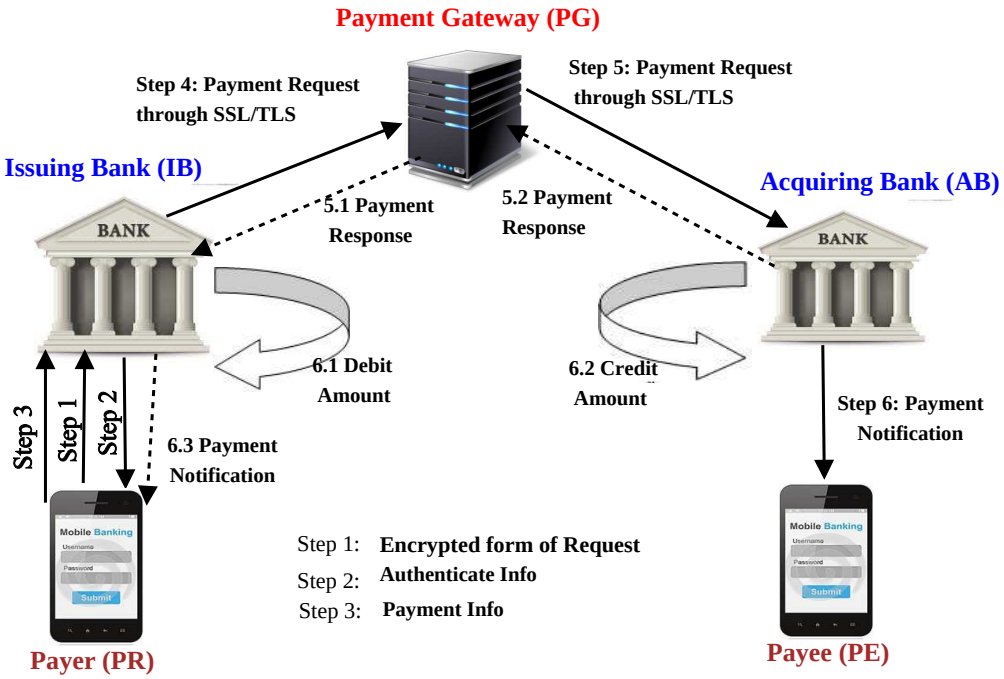


Figure 3.2: Messages flow between participants in SMS-based mobile banking protocol

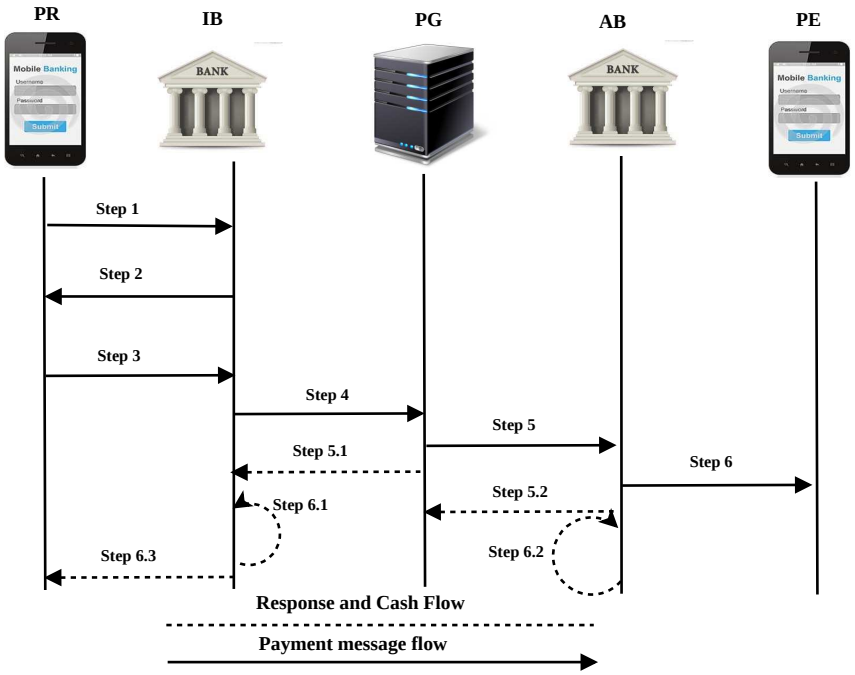


Figure 3.3: Sequence diagram of SMS-based mobile banking protocol

Table 3.1: Authentication phase

PR	IB
1. PR inputs the following ID_{PR}, bank information (BI), generates a secret number (random nonce) with a time stamp TS_{PR} value based on this data PR constructs a $M1 = \{ID_{PR}\|\|BI\|\|N_{PR}\|\|TS_{PR}\}$ 2. PR computes signature with the help of private key generation using ECDSA. ECIES component uses the public key of issuer bank to encrypt the customer's message and send it to the IB. $\mathrm{m}_1=\{M1, SIGN_{PR}^{IB}(M1)\}K_{IB}^{+}$ →	
	1. IB decrypt using private key of K_{IB}^{-} and verify the data integrity and authentication of PR. $if(Verification(SIGN_{PR}^{IB}(M1)) = 1)$ /* Data integrity and Authentication of PR is not compromised */ else /* Data integrity and Authentication PR is compromised */ exit. 2. IB sends encrypted message with the payer's public key of K_{PR}^{+}. With the inputs of session key SK_{ib}, nonce and time stamp value placed in Message2. $M2 = \{SK_{ib}\|\|N_{IB}\|\|N_{PR}\|\|TS_{IB}\|\|TS_{PR}\|\|ID_{PR}\}$ 3. IB construct a signature and encrypt the message using PR public key which is shown below m_2 $\mathrm{m}_2=\{M2, SIGN_{IB}^{PR}(M2)\}K_{PR}^{+}$ ←
3. PR decrypt using private key of K_{PR}^{-} and verify the data integrity and authentication of IB. if (Verification $(SIGN_{IB}^{PR}(M2)) = 1)$ /* Authentication and integrity of IB is not compromised */ else /* Data integrity and Authenticity of IB is compromised */ exit. 4. PR sends payment information to the IB, using shared secret key of SK_{ib}. The message named as Message3 m_3 $M3 = \{PI\|\|TS_{IB}\|\|N_{IB}\}$ $\mathrm{m}_3=\{M3, SIGN_{PR}^{IB}(M3)\}SK_{ib}$ →	
	4. IB decrypt using session key of SK_{ib} and verify data integrity and authentication of PR. if (Verification $(SIGN_{PR}^{IB}(M3)) = 1)$ /* Data integrity and Authenticity of PR is not compromised */ else /* Data integrity and Authenticity of PR is compromised */ exit.

Algorithm 1 PR and IB with SMS signing and encryption.

Objective: To provide secure end-to-end encryption between PR & IB.
Input: Payer (PR) construct message with the SMS mobile application;
Output: IB receives SMS of Gamma;
// PR enter his message in SMS mobile application
$M1_{pr} \leftarrow \{MPIN_{pr}, ID_{pr}, MMID_{pe}, MOB_{pe}, AMT_{pr}\}$;
$\left\{SIGN_{PR}^{IB}(M1)\right\}$; // Generate Signature by X and verified to be bank A
$C_{pr} \leftarrow ENC_{K_{ib}^{+}}(M_{PR})$; // Generate Encrypted SMS
$SMS_{pr}\,(C_{pr}, MOB_{pr})\ to\ SC_{IB}$;
// IB receives SMS_{pr}
$M1_{pr} \leftarrow DEC_{K_{IB}^{-}}(C_{pr})$ // SMS decryption with private key of Issuing Bank;
$\{MPIN_{pr}, ID_{pr}, MMID_{pe}, MOB_{pe}, AMT_{pr}\} \leftarrow M1_{pr}$;
// Verification of Data integrity, Authenticity and Non-repudiation of the Payer;
if Verification ($SIGN_{PR}^{IB}(M1)$)=1 **then**
    // Data integrity and authenticity of PR is not compromised
    IB encrypts data with the PR public key of K_{pr}^{+}. The message consists of session key of SK_{ib} for further payment transactions and send this message as $M2 = \{SK_{ib}, N_{ib}, N_{pr}, TS_{ib}, TS_{pr}, ID_{pr}\}$;
    IB constructed a signed message and verified by PR: $\left\{M2, SIGN_{IB}^{PR}(Beta)\right\}$;
    $C_{ib} \leftarrow ENC_{K_{pr}^{+}}(M_{IB})$ // IB generated encrypted SMS
    $SMS_{ib}\,(C_{ib}, M2)$; // Send to PR
else
    // Data integrity and authenticity of PR is compromised
    Send SMS_{ib} (“unable to identify") to MOB_{pr};
end if
PR receives SMS_{ib} from IB;
$Beta_{pr} \leftarrow DEC_{K_{pr}^{-}}(C_{ib})$; // Decrypt SMS using private key of payer
$\{Sk_{ib}, N_{ib}, N_{pr}, TS_{ib}, TS_{pr}, ID_{pr}\} \leftarrow M2_{pr}$;
// Verify the data integrity authentication of the IB
if (Verification $(SIGN_{IB}^{PR}(M2)) = 1$) **then**
    // Data integrity and Authentication of PR is not compromised
    PR sends payment information to the IB, using session key SK_{ib} called a $M3 = \{PI, TS_{ib}, N_{ib}\}$;
    Signature generated by PR and verified to be IB: $\left\{Gamma, SIGN_{PR}^{IB}(M3)\right\}$;
    $C_x \leftarrow ENC_{SK_{ib}}(M3)$; // Bank A generated encrypted message and send to Payer
end if

Table 3.2: Transaction phase

IB	PG	AB
1. The IB is having the inputs of payment information (PI) and bank information (BI). IB Generates a random nonce secret number with time stamp of TS_{IB} and send this information as $M4 = \{PI\|\|BI\|\|N_{IB}\|\|TS_{IB}\|\|SK_{ip}\}$ 2. IB Computes hashing as $MAC0 = \{hash(SK_{ip}, M4)\}$ 3. IB encrypt message with the shared session key of SK_{ip} and send to payment gateway. $\xrightarrow{(MAC0,M4)SK_{ip}}$		
	1. Message from issuer bank decrypts using the SK_{ip} and gets original message M4 and Verify the identity of issuer bank. if (Verification(MAC0)=1) /* data integrity, authentication and confidentiality of M4 are verified successfully */ then goto step 2 else exit. 2. If all verifications are successful then PG constructs $M5 = \{PI\|\|BI\|\|N_{PG}\|\|TS_{PG}\|\|SK_{pa}\}$ 3. computes hashing as $MAC1 = \{hash(SK_{pa}, M5)\}$ 4. encrypt message M5 with the shared session key of SK_{pa} and send to acquirer bank. $\xrightarrow{(MAC1,M5)SK_{pa}}$	
		1. Message from the payment gateway decrypts using the SK_{pa} and gets original message M5 and Verify the identity. if (Verification(MAC1) = 1) /* data integrity, authentication, and confidentiality of M5 are verified successfully */ then goto step 2 else exit. 2. If all verifications are successful then AB sent payment notifications to both payee and payment gateway.

3.3 Security Analysis of Proposed Approach

In this section, the security requirements verification of our proposed framework is analyzed both formally and informally.

3.3.1 Informal security analysis of proposed approach

Our proposed framework can prevent various possible attacks, as described in the following text.

3.3.1.1 Man-in-the-Middle attack (MitM)

Let us consider that an adversary $\mathcal{A}$ successfully taps into a message during the authentication phase. Assume that an adversary $\mathcal{A}$ generates a new nonce N_{AV} and creates its own current Timestamp TS_{AV}. However, $\mathcal{A}$ does not know decryption key of the issuer bank or issuer's bank or the identity of a payer, it is difficult for an adversary $\mathcal{A}$ to alter the message $M1 = \{ID_{PR}||BI||N_{PR}||TS_{PR}\}$ as $M1 = \{ID_{PR}||BI||N_{AV}||TS_{AV}\}$. Similarly, it is also difficult to modify the message from issuer's bank to the payer, because $\mathcal{A}$ does not know the private key of the payer. Hence from the above discussion, our proposed scheme can also overcome MitM attacks.

3.3.1.2 Replay attack

Let us assume an adversary $\mathcal{A}$ who intercepts network traffic and captures a message. The captured message start to retransmits it, possibly as a masquerade attack during the phases of authentication and transaction resends this message $M1 = \{ID_{PR}||BI||N_{PR}||TS_{PR}\}$ after some time to the IB and PG. Generally, it is not easy to create false sessions between the entities because replay messages are easily detected through the validation of Timestamp TS_{PR} and N_{PR} fresh nonce messages.

3.3.1.3 Wi-Fi sniffing, victim's node capture attack

Suppose an adversary $\mathcal{A}$ physically captures any node, such as a PG, either selectively or randomly from a target field in the network from which a valid node wishes to access confidential data. The PG and AB communication with secret session key is

computed by $MAC1 = \{hash(SK_{pa}, M5)\}$. Now an adversary is able to compromise SK_{pa} of the captured node PG alone. Finally, $\mathcal{A}$ can communicate with the node using only bogus data. Note that uncaptured nodes can still communicate securely with the authenticated users. From the discussion, we claim that capturing of any single node does not leak any confidential data about other uncaptured nodes. Suppose that not only can adversary $\mathcal{A}$ capture a node, but he is also able to capture individual packets with any node. In each established session for communication, all messages are unique because of fresh nonce and timestamps used in messages and are encrypted with session keys, so the adversary will find it very difficult to compromise secure communications between users and other nodes.

3.3.1.4 Impersonation attack

An attacker tries to steal and impersonate all sensitive data of a PR such as ID_{PR}. Adversary $\mathcal{A}$ then knows the seven digit "MMID, and MPIN" from the authenticated PR, but he/she unable to rebuild the message, like $m_1 = \left\{M1, SIGN^{IB}_{PR}(M1)\right\} K^{+}_{IB}$. Even though an attacker $\mathcal{A}$ can construct a message and send it to the IB. Then IB responds accordingly, but this represents an unbreakable problem for $\mathcal{A}$ (to decrypt the message). As a result, an impersonation attack fails in our approach.

3.3.1.5 SQL-Injection attacks

Our proposed framework is tested using static analysis [72]. In this testing mechanism, the attack surface shows no content providers and no services exported. Hence, an adversary $\mathcal{A}$ will find it difficult to perform malicious operations on SQLite databases or any file stores. The test results of Secure-SMS are shown in Figure 3.4. It is clear that our proposed framework protects from SQL-Injection attacks and information leakage. We tested our framework with the help of the Drozer tool [1].

3.3.1.6 Phishing attack

Mobile browser-based banking also suffers from phishing attacks. The phishing attack has been a most common threat in which many people receive spam or fake e-mails

[1] www.mwrinfosecurity.com/products/drozer

Figure 3.4: Static testing of our proposed SMS banking protocol

which try to pose or mimic some organization and try to gain access to sensitive information to commit fraud. In practice, the victims receive phishing emails which redirect them to a web page which is created by the attacker but looks similar to the real website of victim's service provider, such as a bank or any other financial institution. In our proposed mechanism we overcome the phishing attack with the following steps.

- The payer accesses the fake uniform resource locator (URL) of the bank through a phishing e-mail in their browser.
- The bogus website immediately opens a session on the real bank website to impersonate a user and to commit fraud. Because the bogus website has no information regarding decryption, the bogus site has no other choice but to mimic the user's site. The message alpha contains information about payer, bank, nonce, and timestamp $m_1 = \left\{M1, SIGN_{PR}^{IB}(M1)\right\} K_{IB}^{+}$.
- The real bank verifies the signature. Additionally, the bank verifies that the authentication information is correct through the credentials of a payer, and the payer's URL.
- If the comparison is successful, the bank constructs a message $m_2 = \left\{M2, SIGN_{IB}^{PR}(M2)\right\} K_{PR}^{+}$. Otherwise, any mismatch between the real and bogus URLs informs the bank that a problem exists.

As a result, we protect our scheme from real-time phishing attacks.

3.3.2 Formal verification of proposed protocol using AVISPA tool

Many methods have been developed to prototype a network protocol and verify the security properties and correctness. One of the most valid and reliable approach is model checking [349]. The security protocols analyzed with the model checking is a formally based approach for focus on searching for incorrect traces. It follows the standard "Dolev-Yao model" [350] in which the adversary $\mathcal{A}$ is assumed to have control over the network. This state space is explored employing an algorithm to determine a path which leads to a state where the system's properties do not hold. It has various benefits over conventional approaches using logical reasoning, testing, and simulation. A detailed explanation of AVISPA-HLPSL is presented in the Appendix A.1.

- The number and scope of current security protocols under development are outpacing the human ability to carefully analyze and prove or validate them (i.e., users' ability to check the properties of the protocol against every possible single execution trace, but some hidden errors remain in the protocol itself). These errors are explored by using model checking with the AVISPA tool [31].
- The AVISPA tool is completely automated, robust, expressive, and easily usable and can be integrated into the protocol development and standardization process.
- AVISPA has already proved successful in modeling and verifying a large number of widespread security protocols [351]. In our work, we focus on verification of secure mobile SMS-based payment protocols.

3.3.2.1 SMS Framework simulation results

Once the SMS framework is written in the "high level protocol specification language (HLPSL)" specification, for compilation and validation purposes the AVISPA-HLPSL is converted into an "intermediate form (IF)" using the compiler "HLPSL2IF". After translation, we used AVISPA integrated with four back-ends, and out of the four, the first three: SATMC, CL-AtSe, and OFMC indicated that our SMS framework is safe. The last back-end, of TA4SP, reports as "INCONCLUSIVE".

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /opt/avispa-1.1/testsuite/results/smsframework.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 51.82s
  visitedNodes: 8192 nodes
  depth: 13 plies
```

Figure 3.5: AVISPA OFMC back-end result

- AVISPA OFMC back-end reports the SMS framework is safe, which is shown in Figure 3.5.
- AVISPA CL-AtSe back-end verifies that the SMS framework is safe.
- AVISPA SATMC back-end reports that our framework is safe; results are provided in Figure 3.6.
- AVISPA TA4SP back-end reports that certain rules are not fired, so it does provide verification.

From the above results, our proposed framework for SMS is essentially secure and protects from network attacks such as MitM, replay, etc. No revealed threats were found, and security goals were achieved.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
PROTOCOL
  smsframework.if
GOAL
  %% see the HLPSL specification..
BACKEND
  SATMC
COMMENTS
STATISTICS
  attackFound               false     boolean
  upperBoundReached         true      boolean
  graphLeveledOff           1         steps
  satSolver                 zchaff    solver
  maxStepsNumber            30        steps
  stepsNumber               1         steps
  atomsNumber               0         atoms
  clausesNumber             0         clauses
  encodingTime              0.04      seconds
  solvingTime               0         seconds
  if2sateCompilationTime    0.7       seconds
ATTACK TRACE
  %% no attacks have been found..
```

Figure 3.6: AVISPA SATMC back-end result

3.3.3 Formal security analysis of proposed approach using Scyther

Scyther supports an input language known as "software protocol description language (SPDL)" [32] which is based on a C or Java-like syntax. A detailed explanation of Scyther-SPDL is presented in the Appendix A.2. The goals of the SMS-based mobile banking protocol for formal verification [352] should be:

3.3.3.1 Message confidentiality

User messages exchanged between parties should be kept secret. The formalization notation for message confidentiality is
$(\forall N_{PR} message \in \{claim_PR(PR, Secret, npr)\})$ Notation (1)
$(\forall key \in \{claim_PR(PR, Secret, Prpin)\})$ Notation (2)

3.3.3.2 Uniqueness of symmetric keys

To achieve confidentiality, the payer and payee must ensure that the secret key is never shared with any other person. The formalization notation for the uniqueness of the symmetric keys is as follows
$(\forall key \in \{claim_PR(PR|B, Secret, skb)\})$ Notation (3)

3.3.3.3 No theft or stolen service

To achieve this, the bank never shares any transaction operations with untrusted agencies, and only communicates with the authorized user.
$(\forall message \in \{claim_PR(B, Secret, nb)\})$ Notation (4)
From the above notation statements (1), (2), and (3), we can conclude that the parties involved in the proposed model of remitter and bank are authorized parties for exchanging secure messages for payment transactions; other entities are also involved, but for theoretical purposes we have shown only two entities. The protocol is verified using Scyther [353] and the SPDL code is shown in (Appendix section: Figure A.7). The results of the Scyther tool one is claim verification and the other automatic claim verification as shown in Figure 3.7, and Figure3.8, respectively.
The Scyther tool provides a good "graphical user interface (GUI)" that facilitates trouble-free evidence or verification and allows easy understanding of a protocol. Besides, attack graphs are generated whenever threats are found. In our proposed scheme all possible claims are valid, and no revealed threats were found, and hence security goals are achieved.

Scyther results : verify

Claim				Status		Comments
SMS	PR	SMS,PR1	Secret skb1	Ok		No attacks within bounds.
		SMS,PR2	Secret npr1	Ok		No attacks within bounds.
		SMS,PR3	Secret nb1	Ok		No attacks within bounds.
		SMS,PR4	Secret Prpin	Ok		No attacks within bounds.
		SMS,PR5	Niagree	Ok		No attacks within bounds.
		SMS,PR6	Nisynch	Ok		No attacks within bounds.
	IB	SMS,IB1	Secret nb1	Ok		No attacks within bounds.
		SMS,IB2	Secret npr1	Ok		No attacks within bounds.
		SMS,IB3	Niagree	Ok		No attacks within bounds.
		SMS,IB4	Nisynch	Ok		No attacks within bounds.
	PG	SMS,PG1	Secret nb2	Ok		No attacks within bounds.
		SMS,PG2	Secret np1	Ok		No attacks within bounds.
		SMS,PG3	Niagree	Ok		No attacks within bounds.
		SMS,PG4	Nisynch	Ok		No attacks within bounds.
	AB	SMS,AB1	Secret np1	Ok	Verified	No attacks.
		SMS,AB2	Secret na1	Ok	Verified	No attacks.
		SMS,AB3	Niagree	Ok	Verified	No attacks.
		SMS,AB4	Nisynch	Ok	Verified	No attacks.
	PE	SMS,PE1	Secret na1	Ok	Verified	No attacks.
		SMS,PE2	Secret skab	Ok	Verified	No attacks.
		SMS,PE3	Niagree	Ok	Verified	No attacks.

Figure 3.7: SMS framework using “Verification claim”

Scyther results : autoverify

Claim				Status	Comments
SMS	PR	SMS,PR7	Secret skb1	Ok	No attacks within bounds.
		SMS,PR8	Secret nb2	Ok	No attacks within bounds.
		SMS,PR9	Secret npr2	Ok	No attacks within bounds.
		SMS,PR10	Secret npr1	Ok	No attacks within bounds.
		SMS,PR11	Secret nb1	Ok	No attacks within bounds.
		SMS,PR12	Alive	Ok	No attacks within bounds.
		SMS,PR13	Weakagree	Ok	No attacks within bounds.
		SMS,PR14	Niagree	Ok	No attacks within bounds.
		SMS,PR15	Nisynch	Ok	No attacks within bounds.
	IB	SMS,IB5	Secret skpe	Ok	No attacks within bounds.
		SMS,IB6	Secret skib	Ok	No attacks within bounds.
		SMS,IB7	Secret skbp	Ok	No attacks within bounds.
		SMS,IB8	Secret skb1	Ok	No attacks within bounds.
		SMS,IB9	Secret nb2	Ok	No attacks within bounds.
		SMS,IB10	Secret nb1	Ok	No attacks within bounds.
		SMS,IB11	Secret np2	Ok	No attacks within bounds.
		SMS,IB12	Secret npr1	Ok	No attacks within bounds.
		SMS,IB13	Alive	Ok	No attacks within bounds.
		SMS,IB14	Weakagree	Ok	No attacks within bounds.
		SMS,IB15	Niagree	Ok	No attacks within bounds.
		SMS,IB16	Nisynch	Ok	No attacks within bounds.

Figure 3.8: SMS framework using “Automatic claim”

3.3.4 Formal security analysis of SMS using BAN logic

Now we analyze the proposed framework for SMS using BAN logic [39] who extended the formal verification of BAN logic for protocols working in a secure environment.

3.3.4.1 Assumptions for confidentiality and authentication keys

To attain the protocol goals, formal assumptions must hold, as stated by the formal proofs, where "E" is a set of participating entities consisting of payer, bank, and CA. These assumptions specify asymmetric keys (both public and private) for parties to achieve one of the security features of authentication.

A1: Certification Authority believes that $K^+_{PR}, K^+_{IB}, K^+_{CA}$ are the public keys for payer, issuing bank, and certification authority.

$(\forall E \in \{Payer, Issuing\ Bank, Certification\ Authority\} \overset{K^+_{PR}}{\rightarrow} E)$

A2: $(E \in \{Payer, Issuing\ Bank, Certification\ Authority\}\, PR\ believes \overset{K^+_{CA}}{\rightarrow} CA)$. PR and IB, believe that K^+_{CA} is the public key for the CA.

A3: $(\{Payer, Issuing\ Bank\}\, believes\ Pr \overset{SK_{IB}}{\rightleftharpoons} B)$ The PR and IB believe that SK_IB is a shared session key, in other words, it is a symmetric key that is not known to anybody except the valid users: PR and IB.

A4:$(\{Payer, Issuing\ Bank\}\, believes\ PR \overset{mpin}{\rightleftharpoons} IB)$ The PR and IB believe that (m-pin) is a secret mobile permanent identification number that is not known to anyone except parties PR and IB.

3.3.4.2 Assumptions about the freshness of the message

Message freshness to avoid replay attacks and ensure timeliness of the messages

A5: Pr believes fresh (n_{pr}), B believes fresh (n_b). Every participating entity believes that the nonce is fresh and that no one has used the same nonce for two different transactions.

A6: Pr believes fresh (ts'_{pr}), B believes fresh (ts'_b), where $ts'_{pr}ts'_b$ indicates the valid time periods of payer and bank certificates. Each participating entity believes that certificates are within the valid period only.

3.3.4.3 Assumptions about a trusted third party

To specify the level of trust between parties, the "certification authority (CA)" plays a role as a trusted third party for correctly certifying the principals.
A7: $(\forall E, Q \in \{payer, issuing\ bank, certificate\ authority\}$ E believes that the CA controls $\overset{K^+_{CA}}{\rightarrow}\ CA)$. Each participant trusts that the CA for correctly certifies other entities.

3.3.4.4 Detailed security evaluation of the proposed protocol

In the following, we analyze the security goals of the proposed scheme to resist specific possible threats or attacks.
Step 1:$Payer(PR) \rightarrow Issuer\ Bank(IB)$:
$\left\{M1, SIGN^{IB}_{PR}(M1)\right\} K^+_{IB}$
where $M1 = \{ID_{PR},\ BI,\ N_{PR},\ TS_{PR}\}$
IB receives $\left\{M1_{PR},\ SIGN^{B}_{PR}(M1)\right\} K^+_{IB}$ from a payer and performs decryption with his secret key, per the above stated assumptions A1, A2, and A6.
IB decrypts K^-_{IB} using his secret key and obtains $\left\{M1, SIGN^{IB}_{PR}(M1)\right\}$ so from this, IB believes that $\left\{M1,\ SIGN^{IB}_{PR}(M1)\right\}$. Expression (1)
The bank validates the payer's certificate by using a certificate validation procedure, as given in cryptography theory and practice by Stinson [354]:
1. Verifying non-repudiation, message integrity, and authentication of the certificates by verifying the certificate authority signature.
2. Verifying the validity period of the certificate.
After the certification validation procedure is successful, the bank believes that Pr said $SIGN^{B}_{PR}(AI_{PR})$. Expression (2)
Step 2 : The IB construct M2 and send to PR. $Issuer\ Bank(IB) \rightarrow Payer(PR)$:
$\left\{\ M2,\ SIGN^{PR}_{IB}(M2)\right\} K^+_{PR}$,
where $M2 = \{SK_{ib},\ N_{IB},\ TS_{IB}, ID_{PR},\ BI,\ N_{PR},\ TS_{PR}\}$
indicates that PR receives $\left\{\ AI_B,\ SIGN^{PR}_{B}(AI_B)\right\} K^+_{IB}$, from IB and decrypts the message with his secret key, per the above assumptions A1, A2, and A6
Pr decrypts K^-_{PR} using his secret key and obtains $\left\{\ M2, SIGN^{PR}_{IB}(M1)\right\}$.
Pr believes that $\left\{\ M2,\ SIGN^{PR}_{IB}(M2)\right\}$. Expression (3)
The bank validates the payer's certificate by using a certificate validation procedure as

given in cryptography theory and practice by Stinson [354]:
i. Verifying non-repudiation, message integrity, and authentication of the certificates by verifying the certificate authority signature.
ii. Verifying the validity period of the certificate.
After the certification validation procedure is successful, Pr believes that B said $\left\{ M2, SIGN_{IB}^{PR}(M2) \right\}$. Expression (4)
Step 3 : $Payer(PR) \rightarrow Issuerbank(IB) : \left\{ M3, SIGN_{PR}^{IB}(M3) \right\} SK_{ib}$
where $M3 = \{PI,\ N_{IB},\ TS_{IB}\}$
where $PI = \{Payee_Id,\ Amt,\ TS_{PR},\ N_{PR},\ N_B\}$
IB decrypts $\left\{ M3,\ SIGN_{PR}^{IB}(M3) \right\} SK_{ib}$ using the shared symmetric key SK_{ib} between PR and IB obtains $\{PI_{PR}\}$.
IB believes PR said PI. Expression (5)
IB believes the fresh nonce from payer N_{PR}. Expression (6)
IB believes the fresh time stamp TS_{PR}. Expression (7)
PR and IB believe the session key shared between payer and issuing bank. Expression (8). From the above expression statements (1), (2), (3), (4), (5), (6), (7), and (8), we can conclude that payer and bank believe that the certificate and the symmetric and asymmetric keys are valid.

3.4 Performance analysis

We compare our proposed work with the existing solutions presented in [58, 60, 61, 62, 254, 355, 356], [65, 68, 244], and [63] in terms of attack resilience. We mentioned some of the attacks in the previous subsection, regarding the most well-known attacks such as replay, node capture, man-in-the-middle, impersonation, and SQL-injection, which cover parts of the attacks resiliency summary in Table 3.3 which shows security features supported by our scheme and other schemes. From this table, our scheme provides important key features compared to other schemes. In our proposed method, session key updates and other security features such as authentication and non-repudiation are possible. Our scheme has undergone formal security verification using AVISPA, and Scyther tools, whereas other mechanisms did not undergo this verification. The

proposed application can be deployed on mobile devices because it incorporates basic and essential features of security, flexibility, and interoperability.

Table 3.3: Summary of resilience to the attacks

Schemes /Attack	**[254]**	**[58]**	**[355]**	**[60]**	**[61]**	**[62]**	**[356]**	**[65]**	**[68]**	**[244]**	**[63]**	Proposed scheme
Prevention of Replay	yes	no	yes	yes	no	yes	yes	yes	yes	yes	yes	yes
Secure against MIT-M	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Resilient against Password guessing	yes	no	yes	yes	no	yes	yes	yes	yes	yes	yes	yes
Resists node capture	yes	no	yes	yes	no	yes	no	no	yes	yes	yes	yes
Protect from impersonation	yes	no	yes	yes	no	yes	no	yes	no	yes	yes	yes
Prevention of forgery	yes	yes	no	no	no	yes	yes	yes	yes	yes	yes	yes
Supports session key updates	yes	no	no	no	no	no	no	yes	yes	no	yes	yes
Prevention of key privacy and insider	yes	no	yes	yes	no	yes	no	yes	yes	yes	yes	yes
Secure against brute-force	no	yes	no	no	yes	no	yes	no	no	yes	no	yes
Resilient against SQL-Injection	no	no	no	no	no	yes	yes	no	no	yes	no	yes
Prevents Eavesdropping	no	no	yes	yes	no	yes	no	yes	no	yes	yes	yes
Formal Verification by Scyther, AVISPA	no	no	no	no	no	no	no	no	no	no	no	yes

3.4.1 Performance comparison of ECC, RSA, and other cryptography schemes

As the wireless industry explodes, it faces a growing need for security. Applications in sectors of the economy such as healthcare, financial services, and government depend on the underlying security already available in the wired computing environment [357]. Both for secure (authenticated, private) Web transactions and for secure (signed, encrypted) messaging, a full and efficient public key infrastructure is needed. Three basic choices for public key systems are available for these applications:

- "Rivest−−Shamir−−Adleman (RSA)" systems
- "Diffie−−Hellman (DH)" or "digital signature algorithm (DSA)" (e.g., modulo a prime p systems)
- "Elliptic Curve Diffie−−Hellman (ECDH)" or "Elliptic Curve Digital Signature Algorithm (ECDSA)"

Currently, for the same level of resistance against the best-known attacks, the system parameters for an elliptic-curve-based system can be chosen to be much fewer than

the parameters for RSA or mod (p) systems. For example, an elliptic curve over a 160-bit field currently gives the same level of security as a 1024-bit RSA modulus or Diffie-Hellman prime. The difference becomes even more dramatic as the desired security level increases. For example, 512-bit ECC is currently equivalent in security to 15,360-bit RSA/DH/DSA. Public key protocols are used in combination with symmetric key algorithms. The overall strength of the system is the strength of the weakest link. At key lengths of 128, 192, and 256, the advanced encryption standard (AES) has made ECC systems even more attractive as a key agreement alternative. Table 3.4 lists number of the key size standards [238, 358].

Key generation for ECC outperforms RSA at all key lengths and is especially apparent as the key length increases. Because ECC does not have to devote resources to the computationally intensive generation of prime numbers, ECC can create the private/public key pair with superior speed to RSA for comparable lengths. ECC key generation time grows linearly with key size, while RSA's grows exponentially, as shown in Table 3.5. This analysis is based on references [65, 67]. Our proposed method for a secure end-to-end SMS-based approach using ECC is compared with other schemes such as [67] and [65], as shown in Table 3.6. Our framework with different ECC curves and time for generating keys is noticeably lower than existing schemes. Hence, the proposed scheme minimizes communication overhead and reduces computation complexity compared with other results, so we conclude that our ECC-based system is suitable for mobile phones, and hence for mobile payments.

Table 3.4: Comparable key sizes (in bits) for equivalent levels

Symmetric	RSA/DSA/DH	ECC curve
80	1024	160
112	2048	224
192	7680	384
256	15360	512

Table 3.5: Key generation compared with ECC and RSA

Key Length		Time (ms)	
ECC	**RSA**	**ECC**	**RSA**
160	1024	80	160
224	2048	180	7470
384	7680	640	133900
512	15360	1440	679060

Table 3.6: Comparison of key generation time with other schemes in (ms)

ECC curve	**[67]**	**[65]**	Proposed method
160	2437	-	80
224	4449	-	180
384	-	-	640
512	40317	1500	1440

3.5 Implementation

For the purpose of implementation, we have explored various technologies for mobile application development.

3.5.1 Experimental setup

A prototype framework for an SMS banking architecture was implemented on two mobile platforms, Android and J2ME (Java 2 Platform, Micro Edition) at Center for Mobile Banking (CMB).

Android mobile application software

- Eclipse Android developer tools (ADT) and integrated development environmen (IDE) embedded in android software development kit (SDK)
- Mobile Phones: Samsung Galaxy Wave-II (1 GHz Dual Core, 1 GB RAM)
- Application size of 300KB - 600KB

J2ME mobile application

- NET Beans IDE including Java ME
- Bouncy Castle lightweight-149.jar-Crypto Library(Includes the modified ECIES Algo in compliance with SECG) [1]
- On CLDC 1.0 and CLDC 1.1 models
- Mobile phones: Nokia N8, Nokia X2
- Application Size of 64 KB - 100 KB in memory

3.5.2 Message structure

For secure end-to-end SMS banking, it is essential to describe the message format. We extend the structure of banks general SMS format by including a sequence number for detection of duplication messages. The sequence number is generated on the mobile application. The header format of the SMS is given in [65], [69] but those approaches fail to provide the data structure of the SMS message. The banking message structure is comprised of multiple fields. In our mechanism, the SMS size is specified as 50 characters as shown in Table 3.7. Base64 encoding is used to represent encrypted bytes. Generic SMS banking format for a secure message is as follows:
$\{Type\ of\ Transaction\}\,\{MPIN\}\,\{UID\}$
$\{OtherDetails\}\,\{Seq_Num\}\,\{Encrypted\ Message\}$

[1] http://www.bouncycastle.org/java.html

Table 3.7: SMS framework data structure

Field	Length	Data Type
UID	6	Alphanumeric
MMID of Payee	7	Integer
Payee mobile No	10	Integer
Amt	5	Double
Transaction name	4	Char
Mobile PIN	7	Integer
Sequence num	5	Integer
Space	6	Special CHAR
Total	**50**	

3.5.3 Application demonstration and working environment

For demonstration purposes, we discuss a peer-to-peer communication model with encryption and decryption of SMS. We have used two Android phones. The configuration details are as follows.

- Mobile A acts as a customer with an SMS encryption application embedded with the public keys of bank B. We have used three ECs for examination purposes: P-160, P-224, and P-384. In Figure 3.9 represents a payer conducting a fund transfer with SMS using the curve P-224, and Figure3.10 represents encrypted SMS in an outbox of the mobile device.

- Mobile B acts as a bank with decryption application private keys, and bank B receives an encrypted message from A. In Figure 3.11 represents a payee decrypting the fund transfer SMS using the same curve, P-224, as used by the payer who sent the message, and Figure 3.12 represents the payee selecting a message for decryption from his SMS list.

- Bank B runs the decryption application and selects the private key for decryption of the encrypted SMS.

- The decrypted SMS can be viewed on the mobile application.

From the above discussion, we demonstrated SMS encryption using a public key and decryption using a private key. To develop our protocol framework, the support from various entities acting as an SMS gateway, bank, and mobile operator required. Finally, we deployed this web application on a server and linked it to a mobile operator owing a real account with any national bank. An HTTPs connection was used between a voice gateway and server. This pilot project was tested and executed in the server environment, and the details of the servers regarding configuration and encryption and decryption of SMS using ECC P-224 for funds transfer from one account to another are given is shown in Table 3.8.

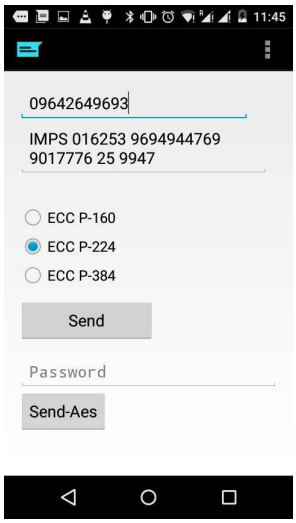


Figure 3.9: Payer sends SMS for IMPS mobile payment using ECC P-224

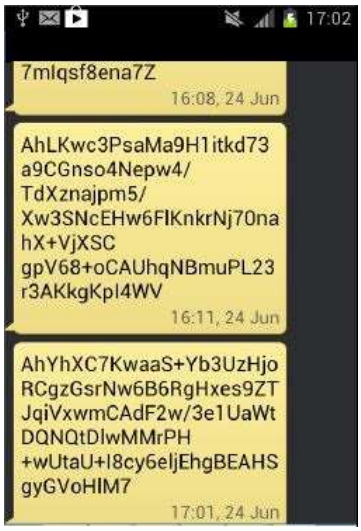


Figure 3.10: Encrypted messages in out-box

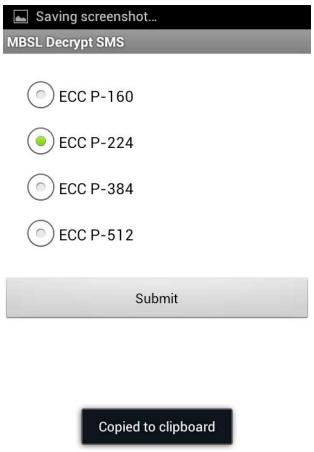


Figure 3.11: Payee select the curve for decryption

Figure 3.12: Payee select a message for decryption

	Windows Server 2008	Linux Cent OS-7
Configuration	2 * Xeon 2.4G/32G RAM	4 * Xeon 2.4G/100G RAM
Application Server	Apache Tomcat 7.0	Apache Tomcat 7.0
Database Server	MySQL 5.6	MySQL 5.6
Mobile Phone	Samsung Galaxy wave-II 1GHz Dual core, 1G RAM	Samsung Galaxy wave-II 1GHz Dual core, 1G RAM
SMS Banking	• Funds Transfer • Mini Statement • Balance Inquiry • Merchant Payment	• Funds Transfer • Mini Statement • Balance Inquiry
Sign Generation	340 (ms)	350 (ms)
Sign Verification	510 (ms)	500 (ms)
SMS Encryption Time	985 (ms)	745 (ms)
SMS Decryption Time	680 (ms)	562 (ms)
No. of SMS Handshake	4	4

Table 3.8: Comparison of performance of SMS security payments on different server environments

3.6 Summary

In this chapter, we have proposed a novel solution for a secure SMS banking protocol. The proposed model provides end-to-end encryption between customer and bank through a mobile application. The proposed framework prevents the storage of private key or customer-sensitive data in an application. To reduce the communication and computational overheads (i.e., storage space and complexity), The customer capable of performing banking transactions using SMS protocol with less complexity and size, we proposed the optimal elliptic curve P-224 for a single confined encrypted message. Also, we formally verified the security properties of the SMS-based mobile banking protocol using model checking tools of AVISPA, Scyther, and theoretical verification using BAN logic. Hence, our proposed framework is withstanding various attacks and easily suitable for the current existing SMS infrastructure because, in our protocol, the size of the SMS does not exceed 160 characters.

Chapter 4

Secure End-to-End Proximity NFC-based Mobile Payment Protocol

Near Field Communication (NFC) is one of the fast-growing technologies related to proximity-based mobile payments. Cappiello et al. [88] have designed a touch-based remote grocery shopping process using RFID and NFC's and tested their framework with an explicit focus on customers touch-based interaction and evaluated the performance using a limited and heterogeneous group of 25 potential customers. (They performed a comparative analysis with existed web-based grocery shopping test and found some limitations). Based on problems identified during analysis they developed touch-based process, which overcomes the problems and combines the benefits of existing virtual shopping with merchandising comfortably delivered to the customer's home. They did not discuss on security analysis aspects during the data transfer using NFC and RFID technologies. Karpischek et al. [92] described a framework on an NFC-based product information system for retailers using Mobile Sales Assistant (MSA) based on NFC and the Electronic Product Code (EPC). The goals and assumptions of the MSA are helping the shop assistants to provide quicker better and superior data and raising the customer's satisfaction. But how MSA could be evaluated at the management level of retailers, recommendations for cross-selling, security aspects in authentication, and privacy issues were not mentioned.

Pourghomi et al. [102] have proposed a framework on NFC payment application. Which is an NFC cloud wallet model to support a whole transaction solution. Using NFC cloud wallet model, a user with an NFC enabled mobile device can pay through

his/her cell phone in a secure way. The whole idea of NFC cloud wallet is that every time the user makes a purchase using payment application, the login credentials of the customer is downloaded into the mobile device (secure element) from the cloud. Once transactions are completed, cloud automatically updates the customer's account. In their approach, the security and privacy issues mainly depend on the cloud which is operated by mobile network operators (MNO), and it is difficult to maintain in a multi-party protocol. Saminger et al. [361] have proposed a framework for ticketing systems using NFC devices. In their approach, validating tickets using reader/writer mode and implementation using light-weight protocols are presented. The design of the m-ticketing system using inverse mode consists of an NFC enabled mobile phone and reader/writer application and server infrastructure. The literature survey on NFC-based mobile payments is shown in Table 4.1.

Table 4.1: Classification of the NFC literature

S.No	NFC operating mode	Approaches
1	Reader/Writer mode	[82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 361]
2	P2P mode	[111, 112, 113, 114, 115, 116, 362, 363]
3	Card Emulation mode	[23, 103, 104, 105, 106, 107, 108, 109, 110, 112, 364, 365]

4.1 Introduction

The vast evolution and rapid development of information technologies are improving the ways of conducting business transactions. Several improvements have been achieved in online transactions, and there are no-constraints of time and place. Presently, customers are showing more interest to purchase goods and make payments using mobile devices. NFC is one of the most popular emerging technologies used in mobile payment systems. NFC is most promising and easy to quickly implement on any platform, and provides good profit and returns for mobile payment stakeholders such as [7] mobile device makers, network operators, banks, merchants, and IT industries. NFC devices used for proximity payment provide short-range communication at high frequency and low bandwidth between two compatible devices within a 10-cm range. Because of its simplicity [359, 360], NFC can be readily integrated with mobile devices, providing an enormous number of use cases including ticketing, access keys for offices, hotels, and houses, loyalty services, peer-to-peer amount/data transfers, credit cards, debit cards, smart advertising, and set up services.
The requirements and essentials of some NFC devices are as follows.

- NFC-enabled mobile phone: NFC-enabled mobile phone, which is commonly known as NFC mobile, is the most important NFC device. Nowadays, the NFC technology embedded in mobile devices is becoming a significant opportunity for the use and wider acceptance of mobile payments and the NFC ecosystem.

- NFC tags: An NFC tag is a passive RFID tag with no integrated power source; however, a little amount of information can be stored on an NFC tag. To give power to an NFC tag, the customer/merchant touches it with a mobile device or an NFC reader. The mobile device or the NFC reader acts as an active device that generates an RF signal. The NFC tag boots up immediately, and the information is transferred or shared according to the pre-loaded algorithm.

- NFC readers: The most general example of an NFC reader is a contactless point of sale (PoS) terminal. The contactless PoS facilitates NFC-enabled payment when an NFC-enabled mobile device touches the NFC reader.

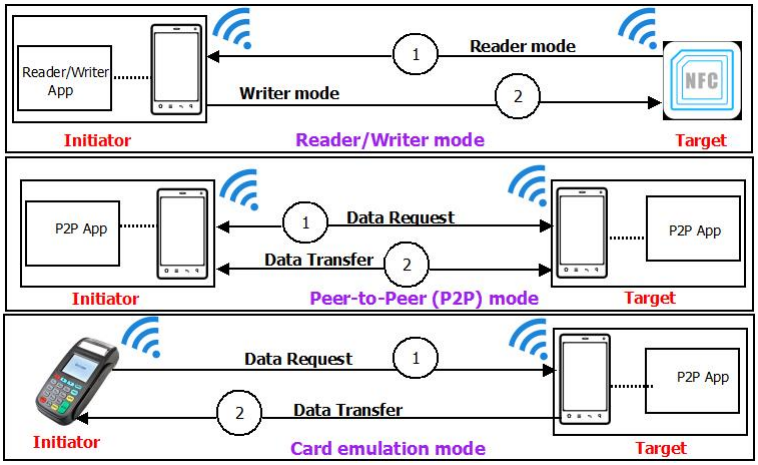


Figure 4.1: NFC operating modes

The NFC is operated mainly in three modes:
(i) Reader/Writer mode: In this mode, communication of data/amount transferred takes place between an NFC mobile phone and an NFC tag. It is further divided into two modes: reader and writer modes. In the reader mode, the NFC mobile phone acts as an initiator that reads data from an NFC tag having the requested data as shown in Figure 4.1. In the case of the writer mode, the initiator (mobile phone) writes the data to the NFC tag. Every time the data are written to the NFC tag, it is updated with the latest and all previous data are overwritten. The maximum data rate possible in this mode is 106 kbps. (ii) Peer-to-Peer mode: Two NFC-enabled mobile devices communicate with each other for sharing any information such as text messages, digital photos, contact records, virtual business records, or any other type of information. In this mode, the information is sent over a bidirectional half-duplex channel; i.e., when one mobile device is transmitting, the other one receives and starts to send information after the first one finishes. Both devices have integrated power, and each one operates using its own energy by being in the active mode. The maximum data rate possible for this mode is 424 kbps.
(iii) Card emulation mode: In this mode, an NFC-enabled mobile phone acts as a contactless smart card. The examples of emulated contactless smart cards are debit cards, credit cards, and loyalty cards. An NFC reader generates its own RF field to initiate the communication. A smart card chip embedded in a mobile device is connected to the antenna of the NFC module. When the customer touches the NFC reader with the

mobile device, the NFC reader initiates the communication. The typical applications for this mode include ticketing, mobile payments, smart environment, access control, loyalty, identity services, and credit card and debit card payments.
In this study, we mainly focus on three areas, namely NFC technology, NFC payment app development, and formal verification of security of NFC payment app.

4.1.1 Organization of the chapter

Our proposed scheme is illustrated in Section 4.2. Section 4.3 provides a detailed analysis of the security of the proposed protocol. Performance analysis is discussed in Section 4.4. In Section 4.5, we present the implementation details of our approach. Section 4.6 summarizes of the chapter.

4.1.2 Contributions

- We have developed an NFC-based mobile payment protocol with the help of ECC. It is operated in the read/write mode for mobile banking and mobile payments, and it is secure against tag cloning and tag impersonation attacks.
- The developed model uses lightweight cryptography of ECC based NFC for signature generation, and verification using elliptic curve integrated encryption scheme (ECIES).
- The proposed protocol is verified and validated with the BAN logic.
- We have simulated our proposed protocol with the AVISPA tool and results of backends are shown to be safe.
- The proposed framework is shown to be secure against numerous security threats using formal and informal security analyses.

4.2 Proposed protocol

In this chapter we propose a secure end-to-end NFC-based mobile payment protocol. Some essential components are used in the model for encryption and signing, which are as follows:

- ECIES: This is used for encryption using public key and decryption using private key.
- ECDSA: It is used for digital signature of a message.
- AES: It is a symmetric key for encryption and decryption of messages.

The notations used for this proposed protocol scheme are listed in Table 1.

4.2.1 Entities in NFC payment protocol and Initial conditions

The participating entities involved in the proposed framework are customers, merchants, acquiring bank (AB), issuing bank (IB), and payment gateway (PG). A wide range of stakeholders are potentially involved in each NFC implementation [138] including mobile network operators (MNO), NFC chipset manufacturers, security element manufacturers, etc.

Initial conditions

- Customer and merchant both are registered with banks for using the NFC payments.
- Every customer has his/her digital certificate distributed by a certification authority (CA).
- Each customer has his/her own wireless public key infrastructure (WPKI) certificate, and it is stored as a secure element (SE).
- All parties involved in the protocol possess public keys and their own certificates.

4.2.2 Detailed protocol

Before making the payment using NFC, the customer (C) should register his/her account with the IB. The IB creates an account and sends a message containing ID_C. The merchant (M) also registers his/her account with the AB that sends a message ID_M and AB public key K_{AB}^{+}. The merchant possesses his/her own PoS machine and reader/writer application. The transaction flow between parties is shown in Figure 4.2. The proposed protocol is divided into three phases (i) customer registration, (ii) authentication & transaction phase, and (iii) notification phase.

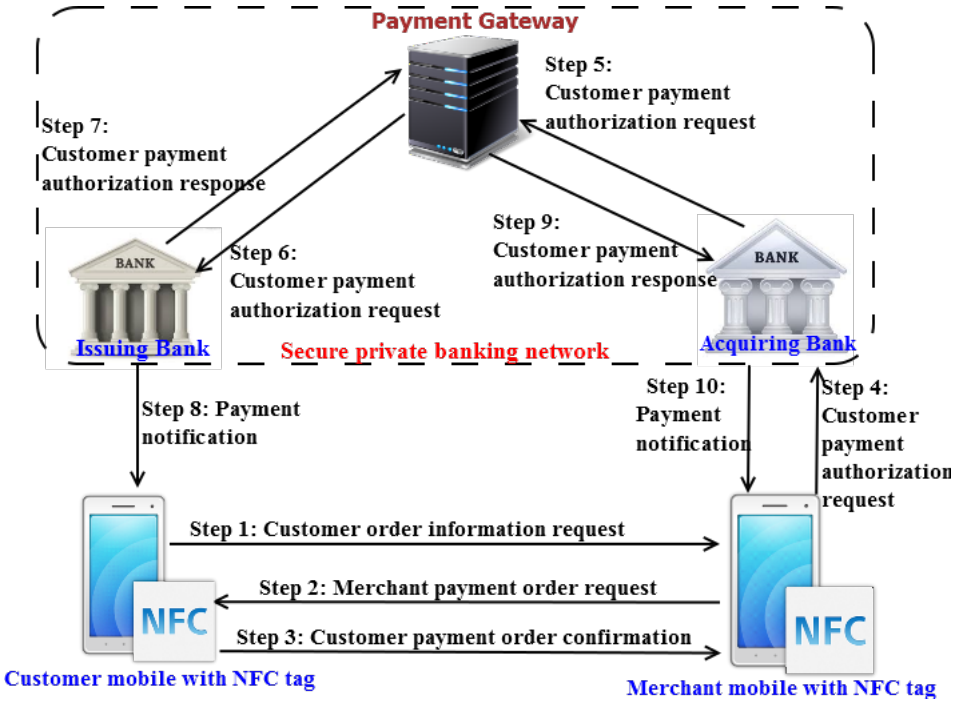


Figure 4.2: NFC-based mobile payment protocol architecture

(i) Customer registration phase: When a customer wants to buy the products and perform the transaction with a merchant, he/she should make sure that the mobile device has NFC function turned ON. The customer downloads the NFC mobile banking application and enters his/her account number and mobile number into the app and sends $\left\{ MESSAGE1, SIGN_M^C(MESSAGE1) \right\} K_M^+$ from his/her NFC-enabled the mobile device to NFC enabled POS. The following steps are necessary to perform NFC-based payments.

Step 1: $Customer(C) \rightarrow Merchant(M)$: $\left\{ MESSAGE1, SIGN_M^C(MESSAGE1) \right\} K_M^+$

where MESSAGE1 = $\{ID_C, N_C, T_C\}$

Step 2: $Merchant(M) \rightarrow Customer(C)$: $\left\{ MESSAGE2, SIGN_C^M(MESSAGE2) \right\} K_C^+$

where MESSAGE2 = $\{POI_M, Amount_M, TID_M, N_M, ID_M, T_C, N_C\}$. The merchant decrypts message1 using his/her private key and constructs message2 with the NFC-enabled POS, which scans the ordered items in the shopping cart and generates the payment ordered information POI_M containing the

price of the product, transaction ID, and a nonce of the merchant. From decrypted message1, the merchant obtains the details of the customer and generates $\left\{ MESSAGE2, SIGN_C^M(MESSAGE2) \right\} K_C^+$.

Step 3:

$Customer(C) \rightarrow Merchant(M) : \left\{ MESSAGE3, SIGN_M^C(MESSAGE3) \right\} K_M^+$

where MESSAGE3 =

$\left\{ POI_M, Amount_M, TID_M, N_M, TID_C, Amt_C, ID_C, N_C, T_C, ID_M, (PI)_{K_{c,ib}} \right\}$

where PI = $\{POI_C, Amt_C, N_C, T_C, ACC_C, MOB_C\}$

After receiving MESSAGE2 from the merchant, the customer decrypts $\left\{ MESSAGE2, SIGN_C^M(MESSAGE2) \right\} K_C^-$ using his/her secret key and receives MESSAGE2 and checks for authentication, and the integrity of the payer.

if(Verification $(SIGN_C^M(MESSAGE2)) = 1)$

/* Data integrity and Authentication of merchant is not compromised */

else

/* Data integrity and Authentication of merchant is compromised */

exit.

After successful verification and validation, the customer constructs the MESSAGE3 with POI_C includes his account details with a registered mobile number for NFC payments with issuing bank and sends $\left\{ MESSAGE3, SIGN_M^C(MESSAGE3) \right\} K_M^+$ to the merchant.

(ii) Authentication & Transaction phase: Once the customer's confidential data and payment orders are received by the merchant, the transaction phase is executed. The merchant sends these details to the AB using the public key of the AB (K_{AB}^+). The sequence of steps is illustrated as follows.

Step 4: $Merchant(M) \rightarrow AcquirerBank(AB) : \left\{ MESSAGE4, SIGN_{AB}^M(MESSAGE4) \right\} K_{AB}^+$

The merchant decrypts $\left\{ MESSAGE3, SIGN_M^C(MESSAGE3) \right\} K_M^-$ using his/her secret key and obtains MESSAGE3. Next, the merchant constructs another message and sends it to AB

where MESSAGE4 = $\left\{ POI_M, Amount_M, TID_M, N_M, ID_M, (PI)_{K_{c,ib}} \right\}$

Step 5 :

$AcquirerBank(AB) \rightarrow PaymentGateway(PG) \{ MAC0, MESSAGE5 \} SK_{AP}$

where MAC0= $\{hash(SK_{AP}, MESSAGE5)\}$ where MESSAGE5 =

$\left\{POI_M, Amount_M, SK_{AP}, TID_M, N_M, (PI)_{K_{c,ib}}\right\}$

The AB decrypts the merchant information of MESSAGE4 with the private key of $\left\{MESSAGE4, SIGN_C^M(MESSAGE4)\right\} K_{AB}^-$. The AB verifies all received information from the merchant because merchant authorization is necessary the details of verification as shown in Algorithm 2. The AB validates the merchant certificate by using the signature validation procedure proposed in [354]. If all verifications are found to be successful, then the AB forwards MESSAGE5 to the PG using a secure private bank network (PBN).

Step 6 :

$PaymentGateway(PG) \rightarrow IssuingBank(IB) \{MAC1, MESSAGE6\} SK_{GI}$

where MAC1= $\{hash(SK_{GI}, MESSAGE6)\}$

where MESSAGE6 = $\left\{POI_M, Amount_M, SK_{GI}, TID_M, N_M, (PI)_{K_{c,ib}}\right\}$

(iii) **Notification phase:** In this phase, the intended parties have received the transaction success/failure notifications, and the IB verifies the customer account which is higher than the purchased order amount as shown in by using Algorithm 3.

Step 7 :

$IssuingBank(IB) \rightarrow PaymentGateway(PG) \{MAC2, (MESSAGE7)\} SK_{IP}$

Now the IB decrypts MESSAGE6 from the PG and verifies the authorization of payment information (PI), which contains information about the customer. The IB verifies the received information from the merchant and the PI information from the customer. If all the verifications are successful, then the IB constructs MESSAGE7 and sends to it PG and the customer. where MESSAGE7= $\{TID, Amount, ID_M, Success/Failure\}$

Step 8 :

$IssuingBank(IB) \rightarrow Customer(C) \left\{MESSAGE8, SIGN_C^{IB}(MESSAGE8)\right\} K_C^+$

IB informs the customer about the transaction success/failure.

Step 9 :

$PaymentGateway(PG) \rightarrow AcquirerBank(AB) \{MAC3, MESSAGE9\} SK_{PA}$

where MAC3= $\{hash(SK_{PA}, MESSAGE9)\}$ where MESSAGE9= $\{TID, Amount, ID_M, Success/Failure\}$

Step 10 :

$AcquirerBank(AB) \rightarrow Merchant(M) \left\{MESSAGE10, SIGN_M^{AB}(MESSAGE10)\right\} K_M^+$

where MESSAGE10= $\{TID, Amount, ID_M, Success/Failure\}$

Algorithm 2 Authentication verification by Acquirer Bank (AB)

Input: $\{M_{ID}, POI_M, TID_M, Amount_M\}$
Output: Acquiring Bank authorizes the merchant ID and payment order information (POI)

if $(M_{ID} = Authorized)$ **then**
  // Proceed to Payment //;
else
  // Unauthorized Merchant //;
  if $((POI_M == POI_C)\ AND\ (Amount_M == Amt_C))$ **then**
    Return 1 True
    // Payment order verification successful //;
  else
    // Payment order verification unsuccessful //;
    if $(TID_M == TID_C)$ **then**
      Return 1 True
      // Return Transaction ID verification successful //;
      if $(N_M == N_C)$ **then**
        Return 1 True
        //Return Nonce verification successful;
      else
        Return 0 False
        // Return Nonce verification unsuccessful //;
      end if
    else
      Return 0 False
      // Return Transaction ID verification unsuccessful //;
    end if
  end if
end if

Algorithm 3 Authentication verification by Issuing Bank (IB)

Input: $\{BAL_C, POI_M, TID_M, Amount_M\}$
Output: Issuing Bank authorizes the payment order information
if $(BAL_C < Amt_C)$ **then**
    // IB send SMS to Customer ("Balance is Low") //;
else
    // Transaction is routes through secure network //;
    if $(POI_M == POI_C \; AND \; Amount_M == Amt_C)$ **then**
        Return 1 True
        // Payment order verification successful //;
    else
        // Payment order verification unsuccessful //;
        if $(TID_M == TID_C)$ **then**
            Return 1 True
            // Return Transaction ID verification successful //;
            if $(N_M == N_C)$ **then**
                Return 1 True
                // Return Nonce verification successful //;
            else
                Return 0 False
                // Return Nonce verification unsuccessful //;
            end if
        else
            Return 0 False
            // Return Transaction ID verification unsuccessful //;
        end if
    end if
end if

4.3 Security Analysis of Proposed Approach

In this section, we analyze our proposed protocol with regard to both formal and informal security features. To estimate the security strength, the validation of our protocol is needed.

4.3.1 Informal security analysis of proposed approach

Our proposed framework can withstand various security attacks, as illustrated in the following propositions.

Proposition 1. *The proposed protocol is secure against replay attacks.*

Proof. In a replay attack, an adversary $\mathcal{A}$ tries to store and resend a captured message or a packet to the victim (server/client). Now let us assume an adversary $\mathcal{A}$ who intercepts the traffic and captures a packet and tries to resend it to the victim, possibly for impersonating during login, and replays the message $MESSAGE1 = \{ID_C||ACC_C||N_C||T_C\}$ using various open source tools. However, resending the messages to create and hijack the sessions between the parties is difficult because replay messages are easily identified by the validation of time stamps T_C and N_C and fresh nonce messages. □

Proposition 2. *The proposed protocol is robust against SQL injection attacks.*

Proof. In our proposed protocol implemented and used in Android mobile devices, the reader/writer app does not allow permissions [366] or restrict access to all content providers. Because of this reason, the adversary $\mathcal{A}$ cannot perform any SQL injection attacks on the database [24]. In our approach, we perform static analysis on both the reader and writer apps as shown in Figure 4.3 and Figure 4.4. From the diagrams, we know that our proposed framework is free from SQL injection. We test our protocol with the help of the Drozer tool [1]. □

Proposition 3. *The proposed protocol protects against Man-in-the-Middle attack (MitM).*

Proof. One of the most powerful of active network attacks is the MitM attack. Because of this attack, many associated attacks like phishing, masquerade, improper session handling, and account lock-out [221] emerge and are executed by the adversary. When an adversary performs a MitM attack on an organization, the network traffic will be intercepted or redirected to the attacker, causing severe damage to the organization. Let us consider that an adversary $\mathcal{A}$ taps the message and analyzes the packet information.

[1] www.mwrinfosecurity.com/products/drozer

```
dz> run app.package.attacksurface com.idrbt_riti.nfc_reader_mbank
Attack Surface:
  2 activities exported
  0 broadcast receivers exported
  0 content providers exported
  0 services exported
    is debuggable
dz> run app.provider.info -a com.idrbt_riti.nfc_reader_mbank
Package: com.idrbt_riti.nfc_reader_mbank
  No matching providers.

dz> run scanner.provider.finduris -a  com.idrbt_riti.nfc_reader_mbank
Scanning com.idrbt_riti.nfc_reader_mbank...

No accessible content URIs found.
dz> run scanner.provider.injection -a  com.idrbt_riti.nfc_reader_mbank
Scanning com.idrbt_riti.nfc_reader_mbank...
Not Vulnerable:
  No non-vulnerable URIs found.

Injection in Projection:
  No vulnerabilities found.

Injection in Selection:
  No vulnerabilities found.
dz> run app.service.info -a com.idrbt_riti.nfc_reader_mbank
Package: com.idrbt_riti.nfc_reader_mbank
  No exported services.
```

Figure 4.3: Static testing of our proposed NFC-reader application

```
dz> run app.package.attacksurface com.idrbt_riti.nfc_writer_mbank
Attack Surface:
  1 activities exported
  0 broadcast receivers exported
  0 content providers exported
  0 services exported
    is debuggable
dz> run app.provider.info -a com.idrbt_riti.nfc_writer_mbank
Package: com.idrbt_riti.nfc_writer_mbank
  No matching providers.

dz> run scanner.provider.finduris -a  com.idrbt_riti.nfc_writer_mbank
Scanning com.idrbt_riti.nfc_writer_mbank...

No accessible content URIs found.
dz> run scanner.provider.injection -a  com.idrbt_riti.nfc_writer_mbank
Scanning com.idrbt_riti.nfc_writer_mbank...
Not Vulnerable:
  No non-vulnerable URIs found.

Injection in Projection:
  No vulnerabilities found.

Injection in Selection:
  No vulnerabilities found.
dz> run app.service.info -a com.idrbt_riti.nfc_writer_mbank
Package: com.idrbt_riti.nfc_writer_mbank
  No exported services.
```

Figure 4.4: Static testing of our proposed NFC-writer application

Now assume that an adversary $\mathcal{A}$ creates a fresh nonce N_{AV} and makes its current time stamp value of TS_{AV}. However, $\mathcal{A}$ does not know the decryption key of any party such as a merchant, and the IB for obtaining the identity of a customer. Without knowing both the decryption key of the merchant and the identity of a customer, it is difficult for an adversary $\mathcal{A}$ to modify the message $MESSAGE1 = \{ID_C||ACC_C||N_C||T_C\}$ as $MESSAGE1 = \{ID_C||ACC_C||N_{AV}||T_{AV}\}$. Similarly, it is also difficult to alter the message from the merchant to the customer, because $\mathcal{A}$ does not know the private key of the customer. Hence from above discussion, our proposed protocol can also overcome MitM attacks. □

Proposition 4. *The proposed protocol defends Wi-Fi sniffing, victim's node capture attack.*

Proof. In this attack, an attacker $\mathcal{A}$ tries to capture any node physically, like PG, either randomly or selectively from a victim's field in the network from which a reasonable node wishes to access secure information. The AB and PG communicate using a secret session key computed by MAC0= $\{hash(K_{AP}, MESSAGE5)\}$. An adversary $\mathcal{A}$ can compromise K_{AP} of that captured node AB alone. Finally, an adversary can communicate with a node, only with spurious information. An important point to be noted is that uncaptured nodes can be still able to communicate securely with the genuine users. From the discussion, we claimed that the interception of traffic and captured information of any victim's node does not leak any other secure information about other uncaptured nodes. Attacker $\mathcal{A}$ can capture a node as well as capture typical packets with any node. Once the login is performed by an authorized person and a session for communication is established, all messages containing time stamps and fresh nonces are encrypted with session keys, making it challenging for the attacker to compromise secure communication between the users and other nodes. □

Proposition 5. *The proposed protocol is secure against NFC TAG cloning and tag impersonation.*

Proof. Tag cloning and tag impersonation are one of the most challenging tasks in NFC/RFID communication. When NFC communication is unauthenticated, an adversary tries to elicit sensitive data and modify information on tags. For example, the user-sensitive information stored on the reader from a stolen reader can provide the required credentials for gaining access to NFC/RFID tags and backend systems. In

our proposed protocol, an adversary finds it difficult to construct the message because each message is unique with a time stamp and fresh nonce messages. Even though an attacker steals credentials from the valid user and constructs the message sent to the merchant, the merchant decrypts the initial message and reconstructs the message sent to the respective customer. In such an exchange, $\mathcal{A}$ is unable to decrypt the message. As a result, impersonation attack fails when our approach is adopted. Moreover, each message is unique, encrypted, and signed, and hence the adversary again fails when he/she attempts to tag cloning. □

4.3.2 Formal security analysis using BAN logic

In this section, we analyze the security issues of our proposed approach. Initially, we validate our protocol using the BAN logic. Ban logic [39] consists of rules on beliefs based on logic, used to verify the nonce and time stamps for avoiding the replay attack and validate the protocol to achieve authenticity and other security features. The notation we use for logical verification is shown in Table 1. Before we apply the BAN logic to our proposed approach, we mentioned assumptions, rules, and goals of our scheme. The following section described on Hypotheses: Assumptions made to analyze the proposed scheme.

4.3.2.1 Hypotheses: The assumptions about nonce and time stamps freshness

These assumptions are used for the freshness of the messages and avoiding replay attacks so that the protocol ensuring timeliness of the messages.
H1: Nonce messages: Each participating entity believes that any nonce it generates is fresh. In other words, the same nonce is never used for two different transaction execution instances of the protocol.

- $C_{BAN}| \equiv \#n_c$; C_{BAN} believes fresh ($\#n_c$), or C_{BAN} believes ($\#n_c$) is true.
- $M_{BAN}| \equiv \#n_m$; M_{BAN} believes fresh ($\#n_m$), or M_{BAN} believes ($\#n_m$) is true.
- $IB_{BAN}| \equiv \#n_i$; IB_{BAN} believes fresh ($\#n_i$), or IB_{BAN} believes ($\#n_i$) is true.

H2: Timestamp messages: T_c', T_m', T_i' these timestamps are used in digital certificates. Every participating entity believes that certificates are within the validity period.

Whereas $\#T_c, \#T_i, \#T_m$ are timestamps generated by customer, issuer, and merchant which ensures timeliness of the messages.

- $C_{BAN}| \equiv T'_c$; C_{BAN} believes fresh (T'_c), or C_{BAN} believes (T'_c) is true.
- $M_{BAN}| \equiv T'_m$; M_{BAN} believes fresh (T'_m), or M_{BAN} believes (T'_m) is true.
- $IB_{BAN}| \equiv T'_i$; IB_{BAN} believes fresh (T'_i), or IB_{BAN} believes (T'_i) is true.

4.3.2.2 Hypotheses: The assumptions about secret keys

H3: Certification Authority (CA) believes $\left(\forall E \in \{C, M, AB, IB\} \overset{K_C^+}{\mapsto} C\right)$
CA believes that all participating entities using public keys of "$K_C^+, K_M^+, K_{AB}^+, K_{IB}^+$". The above keys are used to encrypt the data for communication between entities.
H4: Customer and Issuing bank believes $C \overset{K_{c,ib}}{\mapsto} I$.
The customer and IB believes that $K_{c,ib}$ is a session key shared between the customer and issuing bank and is only known to C and IB.
H5: Sign generation and verification. $C| \equiv M \overset{SIGN_M^C(MESSAGE1)}{\leftrightharpoons} C$
The signed message generated by the customer and verified to be a merchant. The merchant (M) decrypts the signature using the public key of customer and recovers MESSAGE1 using his private key of K_M^-. The Merchant believes Customer (C) said MESSAGE1.

4.3.2.3 Hypotheses: The assumptions about secure private bank network

Acquiring Bank (AB) believes that Issuing Bank and Payment Gateway (PG) data is a secure and timely channel from IB and PG, i.e., the messages of outgoing O_{ib} and O_{pg} are known, and these are sent recently by IB and PG.
Acquiring Bank (AB) believes that Issuing Bank and Payment Gateway (PG) data is a secure and timely channel from IB and PG, i.e., the messages of outgoing In_{ib} and In_{pg} are known, and these are sent by IB and PG.
H6: AB believes $\overset{O_x}{\prec} \left\{ \begin{matrix} IB \\ \\ PG \end{matrix} \right\}$ AB believes secure and timely O_x, where x is between IB and PG.

H7: IB believes $\overset{O_x}{\prec} \left\{ \begin{matrix} AB \\ PG \end{matrix} \right\}$ AB believes secure and timely O_x, where x is between AB and PG.

H8: AB believes $\overset{In_x}{\prec} \left\{ \begin{matrix} IB \\ PG \end{matrix} \right\}$ AB believes secure and timely In_x, where x is between IB and PG.

H9: IB believes $\overset{In_x}{\prec} \left\{ \begin{matrix} AB \\ PG \end{matrix} \right\}$ IB believes secure and timely In_x, where x is between AB and PG.

4.3.2.4 Hypotheses: The assumptions about trusted third party

It specifies the level of trust between parties. The certification authority (CA) plays a role as a trusted third party for correctly certifying the principals.

H10: CA believes $\left(\forall E, Q \in \{C, M, AB, IB, CA\} \overset{K^{+}_{CA}}{\mapsto} CA \right)$

Each participant trusts the certification authority for correctly certifies other entities.

Rule 1 "Message meaning rule": $\frac{C_{BAN} | \equiv C_{BAN} \overset{K}{\longleftrightarrow} M_{BAN}, C_{BAN} \triangleleft \{S\}_k}{C_{BAN} | \equiv M_{BAN} | \sim S}$

The description of the rule is, if C_{BAN} believes that the shared the key K with M_{BAN}, and C_{BAN} sees the message $\{S\}$ encrypted with key K, C_{BAN} believes that M_{BAN} once said S.

Rule 2 "Nonce verification rule": $\frac{C_{BAN} | \equiv \#(S), C_{BAN} | \equiv M_{BAN} | \sim S}{C_{BAN} | \equiv M_{BAN} \equiv S}$

The description of the rule is, if P_{BAN} believes S is fresh and Q_{BAN} once said S, P_{BAN} believes Q_{BAN} believes S.

Rule 3 "Jurisdiction rule": $\frac{C_{BAN} | \equiv M_{BAN} \Rightarrow S, C_{BAN} | \equiv M_{BAN} \equiv S}{C_{BAN} | \equiv S}$

The description of the rule is, if C_{BAN} believes that M_{BAN} had jurisdiction right to S and believes M_{BAN} believes S, C_{BAN} believes S.

To attain the superior security features, the proposed framework should achieve following security goals as described in Table 4.2.

Table 4.2: The goals described to analyze the proposed framework.

Goal 1 $C\mid \equiv K_M^+$	**Goal 5** $C\mid \equiv M\mid \sim ID_C$
Goal 2 $C\mid \equiv IB\mid \equiv K_{c,ib}$	**Goal 6** $M\mid \equiv C\mid \sim ID_M$
Goal 3 $M\mid \equiv C\mid \equiv T_C$	**Goal 7** $C\mid \equiv IB\mid \sim K_{c,ib}, K_C^+$
Goal 4 $C\mid \equiv M\mid \equiv N_M$	

Based on the hypothesis assumptions and rules of the BAN logic as described above, we validate our proposed protocol which can fulfill the goals and follow the rules with clear explanation as follows:

1. To validate the goal 1, the customer C_x believes on the truth of confidential data on K_M^+.

   - $$\frac{M\mid \equiv C \Rightarrow K_C^+, M\mid \equiv C\mid \equiv K_C^+}{M\mid \equiv K_C^+}$$

   The above derivation is described as, if M_x believes that C has jurisdiction over K_C^+ then M_x trusts C on the truth of K_C^+. Hence, we achieve Goal 1.

2. The customer (C) and Issuer Bank (IB) trusts on the belief on confidential information shared using key $K_{c,ib}$

   - $$\frac{C\mid \equiv IB \Rightarrow K_{c,ib}, C\mid \equiv IB\mid \equiv K_{c,ib}}{C\mid \equiv K_{c,ib}}$$

   The above derivation described as, if C believes that IB has jurisdiction over $K_{c,ib}$ then C trusts IB on the truth of $K_{c,ib}$ Hence, we achieve Goal 2.

3. Verification of freshness of customer's time stamp T_C by Merchant M (using Rule 1 and Rule 2):

   - $$\frac{M\mid \equiv C \overset{K_M^+}{\leftrightharpoons} M, M \triangleleft \langle T_C \rangle_{K_M^+}}{M\mid \equiv C\mid \sim T_C}$$ (Based on Rule 1)

The above derivation described as, if M believes the K_M^+ is shared with C and sees $< T_C >_{K_C^-}$, then M believes C once said T_C

- $\dfrac{M| \equiv \#(T_C), M| \equiv C| \sim T_C}{M| \equiv C| \equiv T_C}$ (Based on Rule 2)

The above derivation described as, if M believes that the time-stamp T_C is fresh and C once said T_C, then M believes C believes T_C. Hence, we achieve Goal 3.

4. Verification of freshness of merchant's nonce N_M by the customer C (using Rule 2):

   - $\dfrac{C| \equiv \#(N_M), C| \equiv M| \sim N_M}{C| \equiv M| \equiv N_M}$ (Based on Rule 2)

   The above derivation described as, if C believes that the nonce N_M is fresh and M once said N_M, then C believes M believes N_M. Hence, we achieve Goal 4.

5. Verification of customer identity ID_C by the merchant M:

   - $\dfrac{C| \equiv M \overset{K_M^+}{\leftrightharpoons} C, C \triangleleft \langle ID_C \rangle_{K_M^+}}{C| \equiv M| \sim ID_C}$ (Based on Rule 1)

   That is, if C believes the encrypted message with public key of K_M^+ merchant M and sees $< ID_C >_{X_M}$, then M believes C once said ID_C. Therefore, we achieve Goal 5.

6. Verification of merchant identity ID_M by the customer M:

   - $\dfrac{M| \equiv C \overset{K_C^+}{\leftrightharpoons} M, M \triangleleft \langle ID_M \rangle_{K_C^+}}{M| \equiv C| \sim ID_M}$ (Based on Rule 1)

   That is, if M believes the encrypted message with public key of K_C^+ merchant C and sees $< ID_M >_{X_C}$, then C believes M once said ID_M. Hence, we achieve Goal 6.

7. Verification of the shared key $K_{c,ib}$ between Customer (C) and issuing bank IB:

   - $\dfrac{C| \equiv IB \overset{K_C^+}{\rightleftharpoons} C, C\triangleleft < K_{c,ib} >_{K_C^+}}{C| \equiv IB| \sim K_{c,ib}}$ (Based on Rule 1)

   That is, if C believes the secret $K_{c,ib}$ is shared with IB and sees $< K_{c,ib} >_{K_C^+}$, then C believes IB once said K_C^+. Hence, we achieve Goal 7.

4.3.3 Formal security analysis of proposed protocol using AVISPA tool

This section, describes the AVISPA tool [31, 34, 35] used for simulating our protocol. The implementation codes of the protocol have been written in high-level protocol specification language (HLPSL), which is given in Appendix A.3. The tool simulation results of our proposed protocol by using the AVISPA tool demonstrate that the protocol is secure against passive and active attacks, including MitM and replay attacks.

4.3.3.1 AVISPA simulation results of our NFC framework

Our proposed NFC protocol is written in the HLPSL; for compilation purpose, it is converted into an intermediate form (IF) using the translator HLPSL2. AVISPA uses four backends, and the first three are safe against active and passive attacks. The results of AVISPA for the four backends are:

- "On-the-fly Model-Checker: OFMC" ensures that the NFC framework is safe, as shown in Figure 4.5.
- "CL-based Attack Searcher: CL-AtSe" verifies that the NFC approach is safe.
- "SAT-based Model-Checker: SATMC" reports that our protocol is safe, and results are provided in Figure 4.6.
- "Tree-Automata-based Protocol Analyzer TA4SP" reports that certain rules are not fired, and that verification is not performed.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/opt/avispa-1.1/testsuite/results/nfc.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.05s
  visitedNodes: 4 nodes
  depth: 2 plies
```

Figure 4.5: AVISPA OFMC back-end result

From the above results, it is ensured that our proposed protocol for NFC is safe and secure against active and passive network attacks. No revealed attacks were found. Hence, we achieved our security goals.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
PROTOCOL
  nfc.if
GOAL
  %% see the HLPSL specification..
BACKEND
  SATMC
COMMENTS
STATISTICS
  attackFound               false     boolean
  upperBoundReached         true      boolean
  graphLeveledOff           1         steps
  satSolver                 zchaff    solver
  maxStepsNumber            30        steps
  stepsNumber               1         steps
  atomsNumber               0         atoms
  clausesNumber             0         clauses
  encodingTime              0.06      seconds
  solvingTime               0         seconds
  if2sateCompilationTime    2.5       seconds
ATTACK TRACE
%% no attacks have been found..
```

Figure 4.6: AVISPA SATMC back-end result

4.4 Performance Analysis

We have compared our proposed protocol with the existing approaches reported in [102, 103, 114, 115, 364, 365, 367] for attack resilience. We stated some of the most well-known attacks in Section 4, and those are SQL-injection, replay attack, MitM, etc. Our proposed protocol provides important security features compared with other approaches are shown in Table 4.3.

Table 4.3: Comparison of proposed protocol (PP) and summary of resilience to various attacks

SF	**[102]**	**[114]**	**[103]**	**[364]**	**[365]**	**[363]**	**[367]**	PP
SF1	Y	Y	Y	Y	Y	Y	Y	Y
SF2	Y	Y	Y	Y	Y	Y	Y	Y
SF3	Y	Y	Y	Y	Y	Y	Y	Y
SF4	-	Y	-	Y	Y	Y	Y	Y
SF5	N	N	N	N	N	-	N	Y
SF6	Y	Y	Y	Y	Y	Y	Y	Y
SF7	Y	-	-	N	Y	-	-	Y
SF8	Y	Y	Y	Y	-	Y	Y	Y
SF9	Y	Y	Y	Y	Y	-	Y	Y
SF10	N	N	N	Y	-	N	Y	Y
SF11	Y	Y	Y	Y	Y	-	Y	Y
SF12	N	N	N	N	N	N	N	Y

SF1:Secure against MIT-M, SF2:Resilient against Password guessing, SF3:Prevention of Replay, SF4:Prevention of forgery, SF5:Resilient against SQL-Injection, SF6:Prevents Eavesdropping, SF7: Supports session key updates, SF8:Protect from impersonation, SF9:Prevention of key privacy and insider, SF10: Resists node capture, SF11: Secure against brute-force, SF12: Formal Verification by BAN Logic, AVISPA; Y: yes, N: no, PP: Proposed Protocol; SF: Security Features.

In this NFC-based mobile payment system model, for encryption and decryption, we have used ECIES and for signature generation and verification we have used the ECDSA algorithm. The advantage of ECC is that it is more efficient (in both time and space) and so it is preferred in resource-constrained devices such as wireless and hand-held devices. In ECC, fewer bits are needed for the same level of security compared to the "Rivest−−Shamir−−Adleman (RSA)" algorithm [358, 367]. Here, we have used a 224-bit key size of ECC, which gives the same level of security as the 2048-bit key size of RSA. The key generation time is also less compared with RSA as shown in Table 4.4.

4.4.1 Performance comparison

In Table4.5, we have compared the computation cost of the proposed NFC-based mobile payment protocol scheme and other related schemes [23, 365, 367, 368, 369]. To analyze the performance of each related scheme, we first define some useful notations as follows: T_n stands for the nonce, T_h stands for the hash function, T_{xor} stands for the EX-OR operation, T_{ae} stands for the asymmetric encryption, T_{ad} stands for the asymmetric decryption, and T_{se} stands for the symmetric encryption, T_{sd} stands for the symmetric decryption for using various cryptographic algorithms. In our proposed protocol, we have used ECIES for encryption and decryption and ECDSA for signature generation and verification. The execution time as considered in [370, 371] for cryptographic operations are listed in Table 4.6. The cryptographic operations performed by various entities of the customer (C), merchant (M), PG and banks (AB/IB). The login phase requires
$2T_s$+$2T_{ae}$+$2T_v$+$2T_{ad} \approx$ ((2 *0.34) + (2* 0.18) + (2 *0.51) + (2 * 0.18) = 2.42) in (s) associated with the customer. The transaction phase requires $2T_v$+$2T_{ae}$+$2T_{ad}$+$2T_s$ $\approx$ ((2 *0.51) + (2* 0.18) + (2 *0.18) + (2 * 0.34) = 2.42) in seconds associated with Merchant. The notification phase requires $1T_v$+$1T_{ad}$+$2T_{se}$+$2T_h$+$1T_{sd}$+$1T_{se}$+$1T_s$+$1T_{ae}$ $\approx$ ((1 *0.51) + (1* 0.30) + (3 *0.18) + (2 * 0.0005) + (1 * 0.008) + (1 * 0.34) = 1.69) in (s) associated with banks. The total execution time for all three phases is $\approx$ 6.53 (s).

Table 4.4: Comparision of RSA and ECC concerning key size, key generation, sign generation and verification

Key Length Bits		Key Generation Time (s)		Sign Generation Time (s)		Sign Verification Time (s)	
ECC	**RSA**	**ECC**	**RSA**	**ECC**	**RSA**	**ECC**	**RSA**
160	1024	0.08	0.16	0.15	0.01	0.23	0.01
224	2048	0.18	7.47	0.34	0.15	0.51	0.01
384	3072	0.64	133.90	1.18	0.21	1.8	0.01
512	7680	1.44	679.06	3.07	1.53	4.53	0.03

Table 4.5: Performance comparison

Entity	Scheme [365]	Scheme [23]	Scheme [368]	Scheme [367]	Scheme [369]	Our Proposed
1. Customer	$2T_h$+$3T_{xor}$+ $1T_{ae}$+$1T_v$+$2T_{sd}$	$2T_s$+$2T_{ae}$ +$2T_{ad}$+$2T_v$	$1T_n$+$1T_{sd}$+$1T_{prf}$	$4T_h$	$1T_n$+$1T_{se}$+$2T_{sd}$	$2T_s$+$2T_{ae}$+$2T_v$+$2T_{ad}$
2. Merchant	$1T_{ad}$+$1T_h$+ $2T_{xor}$+$2T_{se}$	$3T_{ad}$+$2T_s$+ $3T_v$+$2T_{ae}$	$1T_s$+$1T_{sd}$+ $1T_{ad}$+$1T_{se}$	NA	$2T_{sd}$+$3T_{se}$+$1Th$	$2T_v$+$2T_{ae}$+$2T_{ad}$+$2T_s$
3. Payment Gateway	NA	$1T_v$+$1T_h$	$1T_v$+$1T_{ae}$+$1Tprf$	$2T_h$+$5T_{xor}$ +$2T_n$+$4T_{ae}$+$4T_{ad}$	NA	$1T_h$+$2T_{sd}$+$1T_{se}$
4. Banks (AB, IB)	NA	AB:$1T_h$+$1T_{ad}$ $2T_s$+$1T_{ae}$ IB:$1T_h$+$1T_s$+ $1T_{ae}$+$1T_v$	NA	NA	$1T_h$+$3T_{sd}$+ $1T_n$+$3T_{se}$	AB:$1T_v$+$1T_{ad}$+$1T_{se}$+$1T_h$ IB:$1T_h$+$1T_{sd}$+$1T_{se}$+ $1Ts$+$1Tae$
5. Phases	(i). Registration (ii). Authentication (iii). Transaction	(i). Registration (ii). Authentication (iii). Transaction	(i). Authentication	(i). Registration (ii). Authentication	(i) Registration (ii). Authentication	(i). Login (ii). Authentication & Transaction (iii). Notification
6. Type of Cryptography	NTRU	RSA	(i). Symmetric (ii). Assymetric	RSA	AES	(i). ECC (ECIES & ECDSA) (ii). AES

Table 4.6: Execution time for cryptographic operations using ECC

Notation	Operation	Time Taken (S)
T_h	One-way cryptographic hash function	0.0005
$T_{\{ae/ad\}}$	Asymmetric encryption or decryption	0.18
T_s	Sign Generation	0.34
T_v	Sign Verification	0.51
$T_{\{se/sd\}}$	Symmetric encryption or decryption	0.005

4.5 Implementation

For implementation, we have considered various technologies for NFC mobile application development.

4.5.1 Experimental setup

A prototype framework for the NFC reader/writer application development is implemented on two Android mobile phones. The list of hardware/software packages used for the development of the NFC protocol framework is shown in Table 4.7.

4.5.2 Message structure

In our proposed NFC framework, we define the data structure format for the encryption of confidential information of the customer and the merchant. Other existing frameworks fail to provide the data structure of the NFC message. For our proposed protocol, the NFC account of both the customer and the merchant as well as the bank accounts of the customer and the merchant are shown in Table 4.8.

4.5.3 Application demonstration and working environment

For demonstration purposes, we describe the reader and writer application for our proposed NFC mobile payment protocol in the following steps.

- Initially, mobile device A acts as the writer application. The customer opens the NFC writer application (WA) and fills the customer details as shown in Figure 4.7. After filling the details, the customer clicks on the ENCRYPT button. For encryption, we have used P-224, ECIES.
- The encrypted data of the customer is written onto the NFC tag, by using the WRITE button as shown in Figure 4.8
- Once the data is written onto the tag, the customer opens his/her reader application and then waves or taps the device on the corresponding NFC tag. Now, the customer enters his/her PIN and amount at the reader application (RA) and

Table 4.7: List of software and Hardware used for NFC implementation

Name Hardware/Software used for NFC	Description
NFC enable phones	Name: Samsung Galaxy wave- II
	Processor: Quad Core 1.6 GHz
	RAM:32 GB
	Operating System: Android 6.0 Marshmallow
NFC enabled POS Reader	Name: VISIONTEK GL-11
	Processor: ARM 9 32 bit Core CPU @ 400 MHz
	RAM: 256 MB
	Operating System: LINUX OS (2.6)
NFC TAG	Name: Milfare Ultralight C
	User Memory: 144 bytes
	Thickness: 0.30 mm
	Size: 35X35 mm
	Data written on Tag: NFC Forum Type 2
Server Environment	Configuration: 2 * Xeon 2.4G/32G RAM
	Application Server: Apache Tomcat 7.0
	Database Server: MySQL 5.6
NFC Application	Wrtier Appliaction Size:3.85 MB
	Reader Application Size:3.96 MB
	Public Key Certificate: X.509 V3, Hash: SHA1 ECDSA
	Encryption and Decryption: ECIES, Key Size:224 bits

Table 4.8: The NFC framework data structure

Data Structure of customer/merchant BANK Account		
Field name	**Field type**	**Field width (in bits)**
Account number (CAN/MAN)	Integer	11
Full name	Character	40
Balance (CAB/MAB)	Float	12
Data Structure of customer/merchant NFC Account		
Account number (CAN/MAN)	Integer	11
Full name	Character	40
User ID(C/M) DB	Alphanumeric	15
NFC PIN for C	Integer	4
IMEI_DB for M	Integer	18
Mobile number	Integer	10
Time stamp and Random number (TSRDB)	Date: Time: Number	15

C: Customer, M: Merchant, CAN: Customer Account Number, MAN: Merchant Account Number, CAB: Customer Account Balance, MAB: Merchant Account Balance, DB: DataBase, TSRDB: Time Stamp and Random number value which is there DataBase, RA: Reader Application, WA: Writer Application

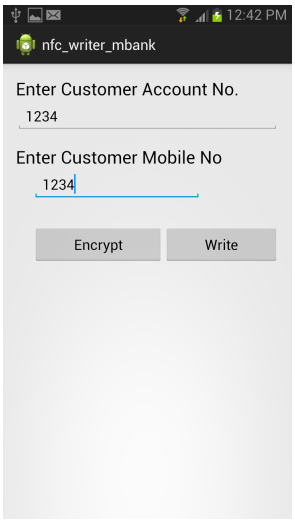


Figure 4.7: Customer fill the details in WA

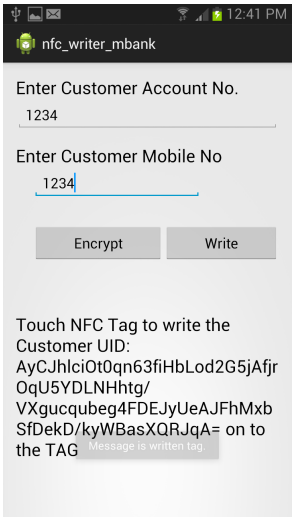


Figure 4.8: Encrypted data of customer written onto the NFC tag

clicks on the NEXT button as shown in Figure 4.9. This payment information is encrypted as shown in Figure 4.10.

- When the SEND button is clicked, the encrypted payment information and encrypted customer account details are sent to the AB server for merchant authorization. For security and reputation reasons, we do not show the server details.
- After merchant authorization, the information is sent to IB for customer authorization through the PG.
- Once a customer is authorized, the amount is debited from the customer's account and credited to the merchant's account.

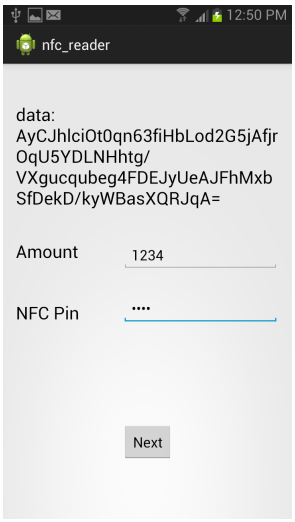


Figure 4.9: Details entered onto the RA of merchant's reader device

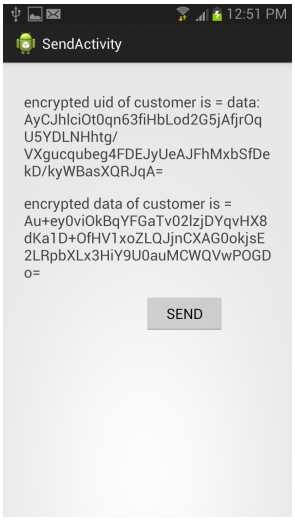


Figure 4.10: RA encrypts the payment information and merchant specific data using AB's public key

4.6 Summary

In this chapter, we have proposed a novel secure proximity-NFC protocol for facilitating mobile payments. The proposed approach is well suited for all types of payments such as mini, macro, and micro payments in end-to-end secure environments. Our approach has been analyzed informally and formally validated by using BAN logic and the model-checking tool of AVISPA. Our proposed protocol is robust against all possible listed attacks and is easily adaptable to mobile devices.

Chapter 5

VAPTAi: Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps

With the change in user preference and rapidly moving technology, users make use of mobile to handle sensitive data, such as personal, healthcare, financial, and so on, on their phones. Companies allow "bring your own device (BYOD)" policy, which results in corporate data being present on the phones. Therefore, it has become a necessity to secure information that is present on the device to protect users' privacy and also fulfill the compliance requirement mandated by the government.

This chapter primarily discusses common vulnerabilities and security-related shortcomings in Android and iOS MBAs. It helps for readers to conduct the static and dynamic analysis of Android/iOS MBAs. Also, the chapter focuses on the practical approach to conduct a vulnerability assessment and penetration testing of Android/iOS MBAs from basic to intermediate level as per industry standards. It will help mobile app developers and information security professionals to find out the commonly known vulnerabilities and ways to mitigate them.

5.1 Introduction

With extensive use of mobile phones in recent years, there is remarkable growth in mobile banking applications (MBAs). Mobile banking and payment applications are used to provide services to the customers. The banking, financial and other payment applications are relatively more sensitive to security compared to another category of non-financial applications such as social-networking applications. Development of MBAs to store and access sensitive data such as "personally identifiable information (PII)" and user credentials [249] is challenging. Any malfunctioning in MBAs leads to a greater effect in the economic wealth of not only the organization itself but also to its customers. Therefore, the developers of MBAs should ensure the security of sensitive information in three levels such as (i) stored in a device (device-level threats), (ii) communicating with servers (communication level threats), and (iii) interacting with other device applications (application level threats). As per "national institute of standards and technology (NIST)" an app vetting process comprises two main activities: app testing and app approval/rejection. The app vetting process and its related actors are shown in Figure 5.1. The roles and responsibilities of actors involved in the vetting process are discussed more detailed in NIST special publication [372]. The related research work on mobile application security is given in Table 5.1.

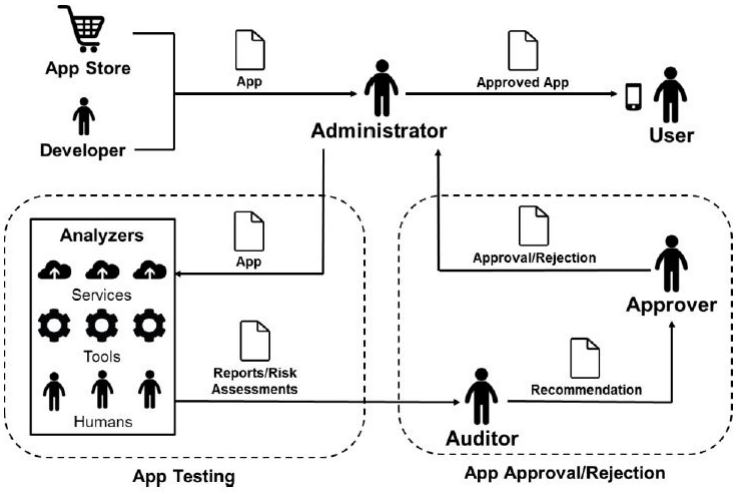


Figure 5.1: App vetting process of NIST

Table 5.1: Related work on mobile security testing

Category of Security Testing	Research Works	Type of program analysis					Platform
		Static	Dynamic	Hybrid	Automatic	Manual	
Application Level Security	iCryptoTracer [373]	Yes	Yes	Yes	Yes	Yes	iOS
	ClusteringiOS[374]	Yes	No	No	Yes	No	iOS
	PiOS [375]	Yes	Yes	Yes	No	No	iOS
	PSiOS [376]	Yes	Yes	Yes	No	No	iOS
	STAMBA [72]	Yes	Yes	Yes	No	No	Android
	Amandroid [377]	Yes	Yes	Yes	No	No	Android
	DroidJust [378]	Yes	Yes	Yes	No	No	Android
	Droidalarm [379]	Yes	No	No	Yes	No	Android
	AppCracker [380]	Yes	No	No	Yes	No	Android
Communication Level Security	Open Wi-Fi [381]	NA	NA	NA	NA	NA	iOS
	SSL:proxy [382]	Yes	Yes	Yes	No	No	iOS/Android
	Dangerous Wi-Fi [383]	No	Yes	No	No	Yes	Android
	SMV-HUNTER [384]	Yes	Yes	Yes	No	No	Android
	AndroSSL [385]	Yes	Yes	Yes	No	No	Android
	StaDyna[386]	Yes	Yes	Yes	Yes	No	Android
Device Level Security	DroidVulMon [387]	Yes	Yes	Yes	Yes	No	Android
	Xing et al.[388]	No	Yes	No	Yes	No	Android

5.1.1 Organization of the chapter

The rest of the chapter is organized as follows: Section 5.2 describes proposed threat model. Section 5.3 provides a security testing framework for the Android and iOS apps. Section 5.4 presents a case study. Section 5.5 describes results and observations. Finally, Section 5.6 has summary of the chapter.

5.1.2 Motivation and contributions

In this chapter, we have analyzed and tested five Android and three iOS MBAs (Having several millions of installations on the smartphone in total)[1]. Before performing vulnerability and penetration testing of these apps, we must be aware of the security and privacy concerns, the system architectures and business models. In many of the organizations, an app is commonly developed by a vendor or third party. Which is downloaded and installed from a Google play store or Apple app store. According to the recent survey report [389], 95% of the top 200 free iOS and Android apps exhibit at least one risky behavior. In Appthority report it is stated that 91% of iOS apps and 85% Android apps showed potential to threat on location tracking, enable address

[1]The detailed statical data of various bank apps as shown in Table 5.3

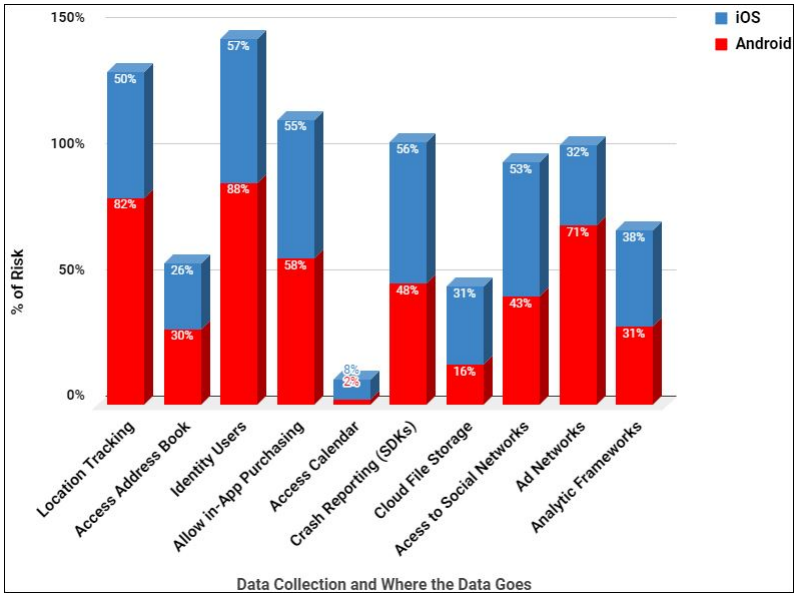


Figure 5.2: Top 10 risky behaviors: (Appthority report)

books, contact lists to be read and identify a user, etc., which is shown in Figure 5.2. In this graph, the first five risks are mentioned about user or customer's data collection, and next five risks are where the collected data goes. Appthority [389] found how secure these mobile apps (free from Google play store) are against various threats. So app developers should take care of the privacy and security related issues especially for the financial and mobile banking apps. Most existing research works are on Android apps [390] and devices which are not relevant to iOS apps or devices.
Our key contributions in this chapter, are the following:

- We have proposed a threat model to detect and mitigate various unknown vulnerabilities in Android & iOS mobile applications.
- We propose a security testing framework for identification of MitM attacks in mobile applications.
- We have analyzed the compliance of mobile banking applications to the "open web application security project (OWASP)" listed mobile security risks and presented our observations.

5.2 Proposed Threat Model

In this section, we propose a threat model as shown in Figure 5.3 which is a consistent and rigorous classification approach in determining vulnerabilities in mobile device and apps. This threat model helps to understand an attack surface and could potentially expose the device and customer confidential information and PII.

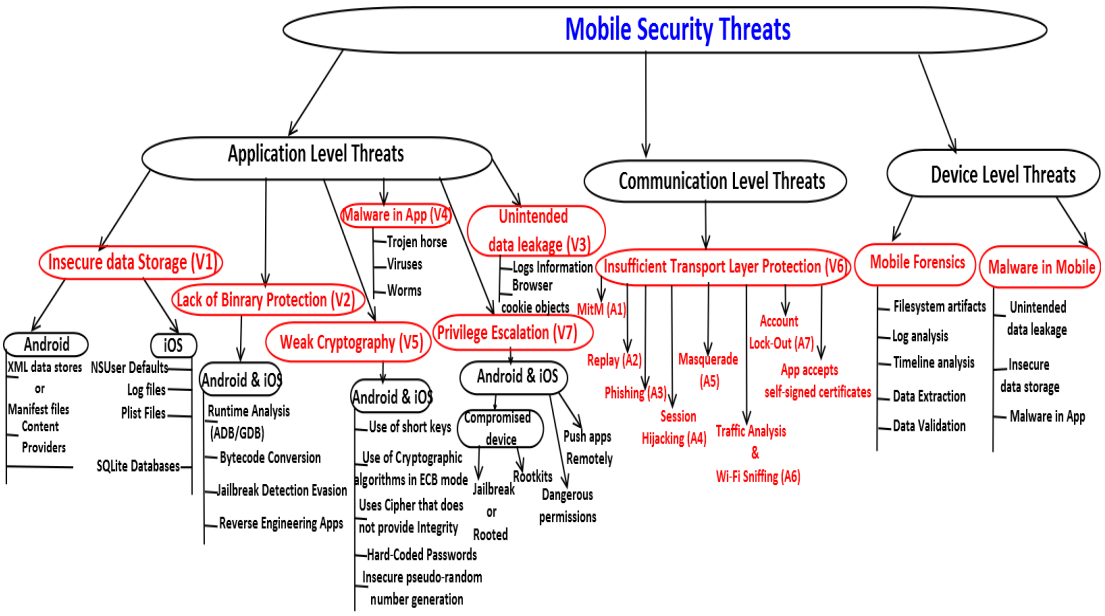


Figure 5.3: Threat model for MBAs in context

The threat types are discussed in the following sections:

5.2.1 Insecure data storage (V1)

Insecure data storage is all about storing the data insecurely on the device. Many times, an application uses simple plist files for iOS applications and XML data stores or manifest files for Android apps or unencrypted database to store sensitive information such as passwords or other user-related information. File systems are only accessible [391]. Insecure data storage vulnerability is the most frequent and, sometimes, very easy-to-find vulnerability. An application should never store sensitive data such as the users' personal, financial, or healthcare information in plain text format. Hence the attacker

uses sophisticated tools, to view sensitive data owing to insecure data storage.
If app developers are storing sensitive information, it may be non-complaint with various compliance standards such as "Payment Card Industry Data Security Standard (PCI DSS)", "Health Insurance Portability and Accountability Act (HIPAA)", and so on. Insecure data storage occurred many times when we tested MBAs use an unencrypted plist files and XML manifest files to store confidential information such as username/password or other customer-related data.

5.2.2 Lack of binary protections (V2)

Lack of binary protections is about checking protections of binary. Checking whether the application allows attackers to revere engineer the application source code. It is essential in case of application handling, as the user's sensitive data should not allow the attackers to decompile the application entirely [391]. We can also check whether binary has implemented any protection for stack smashing attacks or implemented "address space layout randomization (ASLR)" to prevent memory corruption attacks.

5.2.3 Unintended data leakage (V3)

Unintended data leakage arises when a mobile application (iOS/Android/Windows etc.,) developer places confidential data or information in a location on the mobile where it is easily accessible by other application. Thus, resulting in an unintended data leakage or side channel data leakage.
The reasons for side channel data leakage is application screenshots having customer sensitive information, device logs, pasteboards, cookies, and so on, where the application may leak confidential information.

5.2.4 Malware in apps (V4)

Due to the popularity of the Android apps in the market, the primary goal of attackers and malware developers is to exploit threats among apps by reconstructing the most popular apps on Google Play and spreading them on other third-party app stores. It results in decreasing the effectiveness of the app and deteriorates the reputation of the app-store. App malware includes vulnerabilities as Trojan horses, computer viruses,

adware [392] and other harmful programs which are specifically designed to distort or disrupt the mobile operating system and steal user credentials, business, and financial information.

5.2.5 Weak cryptography (V5)

Owing to the broken cryptographic functions or algorithms to encrypt user data on a device. An adversary successfully breaks the encrypted code or sensitive data to its original unencrypted form.

5.2.6 Insufficient transport layer protection (V6)

Insufficient transport layer protection is all about how to user's confidential information across the network. Mobile apps are frequent victims of coffee shop attacks, and if and if an application is sending sensitive data, such as credentials, access tokens, and so on, over HTTP, then any attacker sniffing over the network can easily catch or modify it.
Even if the developer is sending data over HTTPS; however, if he/her does not validate the server-side certificates, then it is vulnerable to SSL pinning such as attacks where the attacker performs man-in-the-middle (MitM) attacks with self-signed certificates. The above communication level vulnerabilities are because of the network being implemented in an inferior "secure socket layer (SSL)" setup which leads to the following attacks:

1. MitM attack (A1)
2. Replay (A2)
3. Phishing (A3)
4. Session hijacking (A4)
5. Masquerade (A5)
6. Traffic analysis and Wi-Fi sniffing (A6)
7. Account lock-out (A7)

The descriptions of A1, A2, and A3 are already described in Chapter 3 in Section 3.3. The descriptions of A4, A5, A6, and A7 are as follows:

5.2.6.0.1 Session hijacking (A4): Session hijacking or improper session handling is managing the user's session token (or OTP) insecurely. Many times, the app developers do not invalidate session tokens at user logout. So, the attacker can reuse these tokens for unauthorized logins. If an attacker can get the victim's token, he can use his credentials to log in and can assign the victim's token using a proxy to log in to the victim account.

5.2.6.0.2 Masquerade (A5): For this attack, the plain-text, and MAC addresses are combined in all packets, and these are transmitted through wireless links. An attacker can perform eavesdropping for identification of MAC addresses. The attacker is also capable of altering its MAC address. Because of this, an attacker can masquerade as any workstation or wireless station by spoofing its MAC address. An attacker does not only spoof the MAC address but also can install his Access Point, with a fake MAC address and a spoofed "service set identifier (SSID)".

5.2.6.0.3 Traffic analysis and Wi-Fi sniffing (A6): The adversary may be able to perform this with the help of access point (AP) device and capture packets during its transmission. It enables an adversary to have access to three types of data. Firstly, data related to detection of activities on the network. Secondly, data related to detection of a physical position of an access point (AP), this information is essential for an adversary. Thirdly, data about the communication protocol.

5.2.6.0.4 Account lock-out (A7): As a security measure, most of the financial service providers have a limit on the number of authentication attempts by the users. Beyond a threshold number of attempts with the wrong password, the user account gets locked in the server. An attacker can exploit this mechanism of security by attempting to log in several times with the wrong password of a particular user account until it results in an account lockout attack. It's not hard for an attacker to find whether a packet that is carrying user credentials is HTTP or HTTPs traffic.

5.2.7 Privilege escalation attack (V7)

Android permissions is an advanced security model, which is mainly operated at application level only [393], and sandboxing. It allows app developers and users to restrict to install an app to the privileges it has assigned during installation time. The exploitation of threats in the source code is hence believed to be confined within the privilege boundaries of an application's sandbox.

5.3 Security Testing Framework

Penetration testing refers to the experimental security testing for identifying the vulnerabilities invisible in applications. It is used primarily with the permission to test cryptography properties such as authentication, confidentiality, non-repudiation, etc., discover vulnerabilities in the testbed environment and examine the performance of a system in extreme conditions. Static and dynamic analysis helps to detect vulnerabilities by performing vetting (testing) of the mobile apps.

5.3.1 Static analysis of MBAs

The static analysis is an activity that involves the testing of MBAs in identifying hidden vulnerabilities with the help of humans, tools, and services, to derive risk assessments and vulnerability reports.

5.3.1.1 Android MBAs

In case of Android apps, static analysis mainly focuses on the data comprise of the "Android application package (APK)" file, including the manifest file of (AndroidManifest.xml) and the compiled code of reverse engineering files (of classes.dex). Static analysis involves determining insecure" application program interface (API)", or it might include information flow through the program to determine potentially unsafe or weak handling of program inputs. Conducting of static testing on the mobile device using various tools and services is given in Figure 5.4. To perform static testing, initially one has to install APK file on a mobile device using "Android debug bridge (ADB)" by the command of (adb install filename.apk), and via Google play store. The

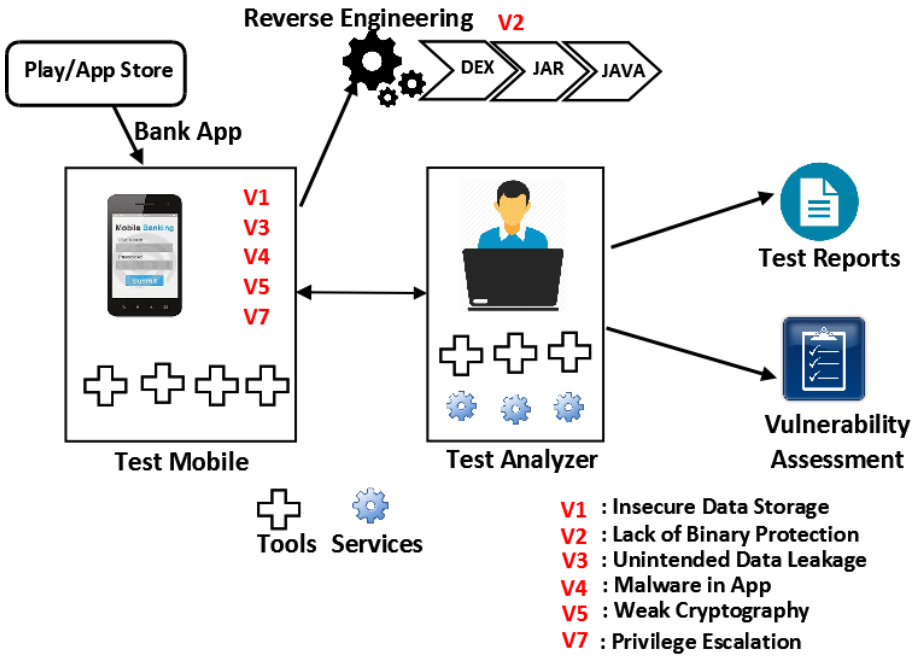


Figure 5.4: Testing process of static analysis

tools used at the application level and communication level for Android are shown in Table 5.2.

Table 5.2: List of penetration testing tools, software, and devices used for our demonstration of vulnerability detection

Type of device	Security Analysis Levels	Name of tool	Platform	Purpose of testing	Vulnerabilities
Test Mobiles: 1. SAMSUNG GalaxyS2 (Rooted) 2. iPhone 4s (Jailbroken device) 3. iPhone 6s (Non-Jailbroken) **Access Point (AP):** 1. CISCO-LinksysPUUU0M300147 **BurpSuite Hosts:** 1. HP ProBook 4540s (Android) 2. Macbook Pro A1278 (iOS)	**Application Level Security**	Drozer [394]	Android	Applications, Providers etc.,	**V1-V3, V7**
		Virus-Total [395]	Android	Malware in apps	**V1, V4**
		Apktool analyzer [396]	Android	Receivers, services etc.,	**V2-V3**
		Dex2jar [397]	Android	Mobile decryption, unpacking & conversion	**V1-V2**
		Clutch [398]	iOS	Mobile decryption, unpacking & conversion	**V5**
		baksmali [399]	Android	Static binary analysis: disassembly, decompilation	**V2**
		Keychain Dumper [400]	iOS/Mac OS	App reverse engineering tools	**V1-V2**
		class-dump-z [401]	iOS/Mac OS	Static binary analysis: information dumping	**V1-V2**
		Introspy-Android [402]	Android	Dynamic binary analysis: debugging, tracing	**V7**
		Introspy-iOS [403]	iOS	Dynamic binary analysis: debugging, tracing	**V7**
		Cydia Substrate [404]	Android/iOS	Runtime manipulation: code injection, patching	**V1, V3**
		Cycript [405]	iOS/Mac OS	Runtime manipulation: method swizzling	**V5-V6**
		idb tool [406]	iOS/Mac OS	Exploit run-time vulnerabilities in iOS apps	**V1-V3, V5-V7**
		SQLite browser [407]	Android/iOS/Mac OS	Exploit database vulnerabilities in Andrid &iOS apps	**V1, V3**
		Xcode [408]	iOS/Mac OS	Detection of vulnerabilities in plist, SQL database	**V1, V3**
		FindBugs [409]	Android	Permissions	**V1, V7**
	Communication Level Security	BurpSuite V1.7.0 [410]	iOS/Android	Interception of network	**V1-V7**
		Wireshark [411]	iOS/Android	Packet analysis	**V1-V7**
		Connectify Hotspot		HTTP proxy	
	Device Level Security	Andriller [412]	Android	Android forensic tool	**V1, V3, V5**
		BlackBag Blacklight [413]	iOS/Android	Data analysis forensic toolkit	**V1, V5**
		iPhone Backup Unlocker [414]	iOS	Analyze an iDevice encrypted backup	**V1, V5**
		Encase forensics V7 [415]	Android/iOS	Log file and event log analysis	**V1, V5**
		Slueth Kit+Autopsy Browser [416]	Android/iOS	Efficiently analyze hard drives and smart phones.	**V1, V5**

5.3.1.2 iOS MBAs

To conduct a security assessment of an iOS application, we will need to access the iDevice's filesystem and keychain data, hook debuggers to applications, and so on. It's not possible to perform iOS app penetration testing (pen-testing) with so many restrictions and this is the point where we need to jailbreak the iDevice to bypass security restrictions provided by iOS. Jailbreaking is the privilege escalation from the mobile user to root user. Jailbreaking is exploiting the iOS vulnerability and maintaining root access on the device by installing utilities on it. Jailbreaking allows us to do the following:

- Access complete filesystem on iDevice
- Install any app from outside App Store

Once the iDevice's has been Jailbroken, then we conduct static and dynamic analysis; the tools we have used for the security level of application and communication for iOS are shown in Table 5.2.

5.3.2 Dynamic analysis

For conducting MitM simulation during dynamic analysis, our testbed consists of four devices. One HP laptop serves as the compromised access point. A CISCO Linksys PUUU0M300147 is a wireless access point. One Samsung Galaxy GT-19100LKAINU smartphone with the configuration of ("Dual-core 1.2 GHz, 1 GB RAM, with Android 4.1.2 (Jelly Bean)"). Finally bank web server for implementing the MitM. In this dynamic analysis, the tester simulates the real attacker and targets the victims and web servers. The testing scenario is shown in Figure 5.5, which is the situation of MitM attack using the Burp proxy (192.168.50.1) of BurpSuite [410]. An attacker sets up a fake Wi-Fi access point when test mobile user (192.168.50.15) connects to the Internet via the access point. Hence, any network communication that takes place between any MBA, and its respective bank server now passes through the Burp Proxy. Thus, the host running BurpSuite acts as a MitM attacker. Note that Burp Proxy can generate a self-signed certificate, essentially an unauthorized one. This fake certificate is stored in the user space, instead of root space, of mobile host's key store to mimic a real attack scenario as it does not require the root access.

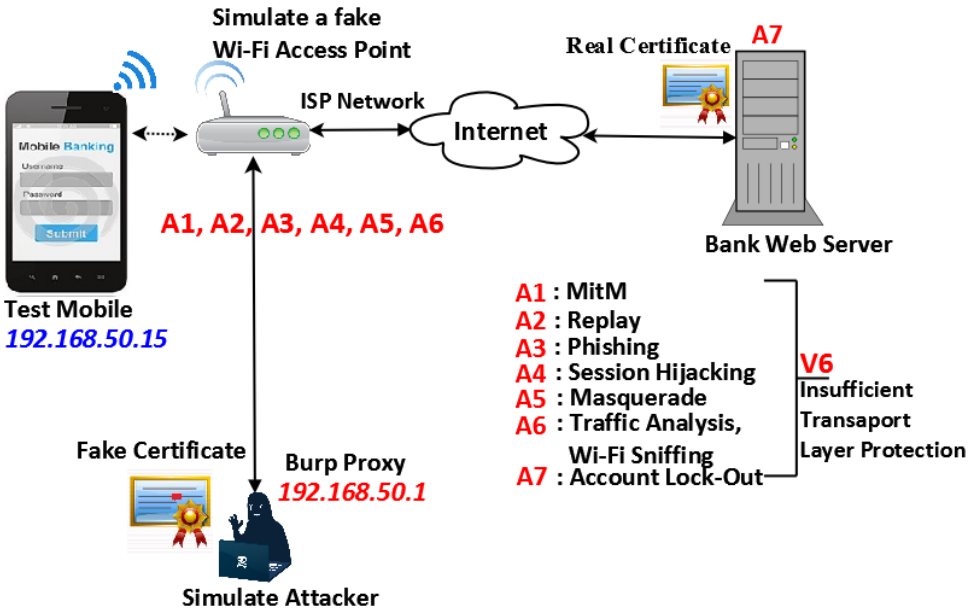


Figure 5.5: Testing process of dynamic analysis

The situation of the attacker is actively monitoring over the network and control of the fake WiFi access point (AP). The MBA is trying to establish a secure channel (HTTPs) connection with the corresponding bank server. This request received by the attacker and immediately send self-certificate to the MBA via fake WiFi access point. Also, the attacker requests the bank server for HTTPs traffic, and it receives the original certificate from the bank server. Finally, two secure (HTTPs) connections established. One is from MBA to an attacker, and the other is attacker to the bank server. Such an arrangement possible only when the receiving host does not check the authenticity of the certificate it receives.

5.4 Case study

We have taken a sample of five Android and three iOS mobile banking applications that are currently being deployed by the nationalized public sector banks in India and are used by their respective customers. However, for security and reputation reasons, we anonymize the exact names of the banks whose banking apps we have considered for testing. Most of these banks have used external developers (called vendors,

anonymized as D1, D2, etc. in the Table 5.3) to develop their banking apps.

Table 5.3: Data set, (D_i indicates developer D) for an app.

Bank_App	Platform	No. of customers downloaded
D1-Bank_1	Android	10,000,000
D2-Bank_2	Android	1,000,000
D3-Bank_3	Android	5,000,000
D3-Bank_4	Android	500,000
D4-Bank_5	Android	500,000
D1-iBank_1	iOS	5,000,000
D2-iBank_2	iOS	500,000
D3-iBank_3	iOS	100,000

5.5 Results and Observations

We have conducted static security analysis for five Android MBAs as a case study, which is described in Table 5.3 and performed static analysis using tools discussed in Section 5.3.1 and list specified in Table 5.2. We further did the testing of MBAs, and provide a security assessment for D4-Bank_5 as shown in Table 5.4.

Table 5.4: The list of exported activities, services and broadcast receivers for D4-Bank_5

Item	Explanation
Activities	com.lcode.uconpatracker.LoginActivity
	com.lcode.uconpatracker.MainActivity
	com.lcode.uconpatracker.RecoveryDetails
	com.lcode.uconpatracker.AssignAccounts
	com.lcode.uconpatracker.AssignedRecallHistory
	com.lcode.uconpatracker.Reminder
	com.lcode.uconpatracker.Message
	com.lcode.uconpatracker.PasswordResetActivity
	com.lcode.uconpatracker.PasswordActivity
	com.lcode.uconpatracker.UserRegister
Services	com.lcode.network.ReachService
Broadcast receivers	com.lcode.util.Autostart
	com.lcode.network.NotificationBarAlarm
	com.lcode.uconpatracker.IncomingSms

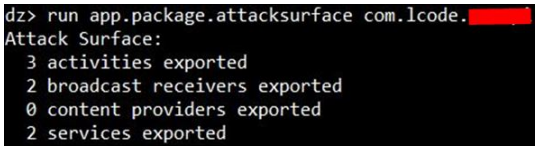


Figure 5.6: Vulnerabilities are identified on the attack surface

5.5.1 Static analysis for Android MBAs

The list of vulnerabilities is identified at the attack surface of Android MBAs are briefly explained as paragraphs.

(1) **External storage access:** The app accesses the external storage directory, also referred to as the SD card. External storage can be accessed by any app on a device with the READ/WRITE_EXTERNAL_STORAGE permission. It is therefore recommended not to store any sensitive information in external storage.

(2) **Exported activities to other apps:** These are the application components, which allow user interaction. It is required during the security assessment to find out which applications are launched without permissions in case any of them provide access to confidential information. In Figure 5.6 shows list of exported activities are three.

(3) **Permits data to be stored in back-end databases and restored:** The app allows backup of its data. A malicious actor with physical access to the device could get access to sensitive data by retrieving private files, databases, shared preferences files, caches or libraries within the app.

(4) **The app is debuggable:** A malicious actor with physical access to a device that has USB debugging enabled can attach a debugger to the app's process during execution. It is dangerous because it could expose sensitive data, enable reverse engineering, and allow the execution of arbitrary code.

(5) **Hard-coded cryptographic key:** Cryptographic keys should never be hard-coded, as an attacker can find the key using open source and freely available tools. It may compromise the security of whatever the key is used to protect.

(6) **The app contains native code:** Native code does not have the same security protection as Java and is vulnerable to buffer overflows, use after free errors, and off-by-one errors. Native code can also be loaded from untrusted sources, such as a shared directory or the network.

(7) **Allows cryptographic algorithms in ECB mode:** The app developers should not

Figure 5.7: List of permissions used by the app

use electronic code book (ECB) mode for encryption. Because ECB mode for symmetric encryption approach does not support a general notion of privacy and it cannot be secure against chosen plaintext attacks (IND-CPA) [417, 418].

(8) **Check for dangerous permissions used by the app:** The developer should strictly avoid dangerous permissions used in the app. Otherwise, these permissions [366] control access to potentially dangerous APIs calls, like those related to collecting secure data or spending money. The dangerous permissions are when we use an app without user permission, read the list of contact numbers, showing geographic location, send the text message to untrusted parties, etc. We have identified some of the permissions given by the app as shown in Figure 5.7.

(9) **Allows reflection code:** Reflection is used to instantiate new objects, invoke methods, and to get and set fields at runtime. While reflection can be used for a legitimate purpose, it is also commonly employed by malware to obfuscate malicious behavior.

(10) **Uses object deserialization:** The app calls the java.io.ObjectInputStream.readObject() method to deserialize objects into memory. Object deserialization is a common source of vulnerabilities, particularly when the object may be from an untrusted source. It is recommended to avoid object deserialization when possible, or otherwise to harden the ObjectInputStream against

attacks. One powerful hardening technique is to override the resolveClass() only to allow expected classes. It can be implemented by subclassing ObjectInputStream [419].

(11) **Insecure Pseudo-random number generation:** If the app uses a pseudo-random number generator which returns a predictable sequence of numbers that is unsuitable for security purposes.

(12) **Broadcast receivers accessible to other apps:** The app developer frequently keep useful data at an attack surface; it makes an opportunity for an adversary to do great things, from executing arbitrary code to proliferating data [72] for security reasons, we only present a snapshot of the attack surface of MBA as shown in Figure 5.6.

(13) **Content providers accessible to other apps and app allows SQL injection:** The app doesn't allow permissions [366] for the content providers or use of permissions restrict access to all content providers. Because of this reason, the adversary cannot perform any SQL injection attacks on the database [24].

(14) **The source code is not obfuscated:** If the app does not obfuscate the majority of its code by renaming classes, fields, methods, and variables, then the attacker easily to decompile the app into the near-original source code using reverse engineering process [420]. It is recommended to obfuscate the app's code using a tool such as "ProGuard" to make it more difficult to reverse engineer.

(15) **Logs information:** The app stores logging information to the system log. While apps often log information for debugged purposes, this should be removed before an app is launched into the market/production. No sensitive information, such as keys or authentication tokens, should ever be written to the system log. Otherwise leaking information to Logs is possible [421].

(16) **Does app executes environment commands:** The app runs Linux-style environment commands. It can be used to run standard terminal commands or arbitrary files. However, this requires appropriate input sanitization and data protection.

(17) **Uses cipher that does not provide integrity:** The app contains a cipher that does not include a keyed-hash message authentication code (HMAC) to verify the integrity of data before attempting to decrypt it. If the ciphertext can be controlled by an attacker, the cipher provides no way to detect that the data has been tampered. Manually using an HMAC with a cipher that does not include it can lead to errors.

Galois-counter mode (GCM) is one cipher mode that provides authenticated encryption [419].

(18) **Use of weak security algorithms or hash functions:** Mobile application developers should strictly avoid the cryptographic algorithms, which are proven to be broken or weak. The MD4, MD5, SHA1, and RC4 algorithms are dangerous if used for sensitive data because of it highly vulnerable to collision attacks [422].

(19) **Weakly configured XML parser:** The app may use poorly configured XML parsing libraries. If XML is parsed from an untrusted source, this can lead to XML external entity (XXE) and Denial of Service (DoS) attacks.

(20) **Weak construction of socket factory:** The app creates an SSL socket factory that may be vulnerable to MitM attacks. The SSLSocketFactory class creates sockets that do not automatically perform certificate hostname validation, leaving the burden on the developer to manually build and use a HostnameVerifier. Failure to do so could allow an attacker to MitM the SSL socket connection by presenting the legitimate signed certificate for a different hostname.

The following vulnerabilities are identified during static analysis of Android MBAs as described as above paragraphs, and bar graph as shown in Figure 5.8. In this figure, the horizontal axis specifies the name of vulnerability test and the vertical axis shows the number of vulnerabilities.

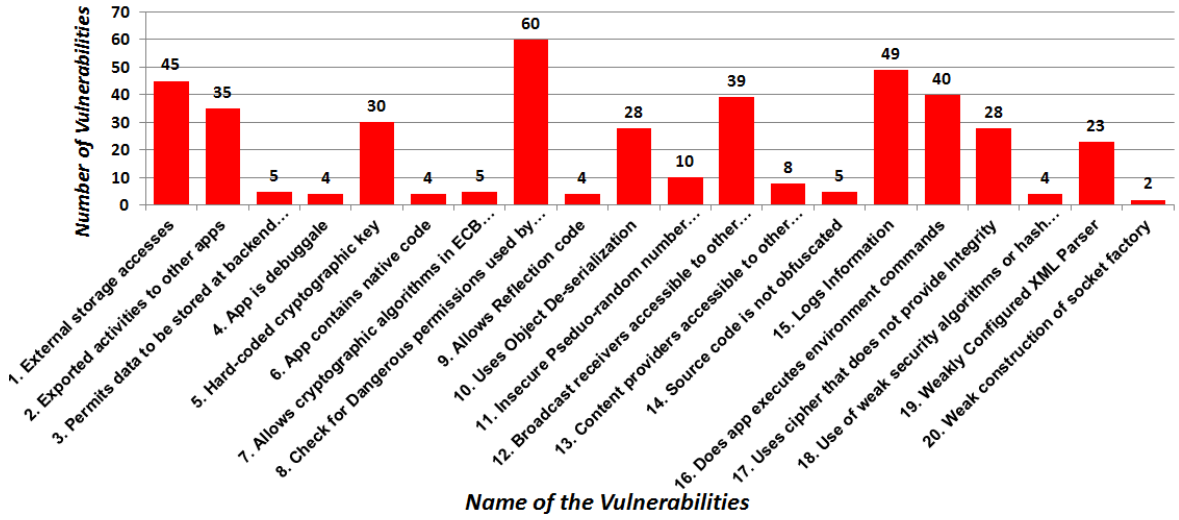


Figure 5.8: Static analysis of Android MBAs

5.5.2 Dynamic analysis of Android MBAs

Dynamic analysis is one of the most important aspects when it comes to Android applications. An attacker can change the application behavior as per requirement. To perform dynamic analysis of MBAs, for data transmission between a mobile app and the server vice-versa. The MBA requires the users (testers/analyzers) to be registered with the banks otherwise detailed analysis of MBA is not possible during communication level security. The lack of SSL pinning (certificate/public key pinning) leads to attacker can implement MitM attack.

The following observations made during dynamic analysis:

(1) D1-Bank_1: This MBA requires that the customer registration with a bank to gain user_ID and password. The customer's "user_id" & "Password" can be trivially captured and the customer's account compromised. The attacker efficiently implement a MitM attack because of poor SSL setup. Once the adversary gained login credentials from a legitimate user, they can be used to login to the account. We have observed that user_ID and password are not in an encrypted format, and obtain the JavaScript Object Notation (JSON) formatted request data as in Figure 5.9.

```
POST/api/login HTTP/1.1
Content-Type: application/json
Content-Type: application/json; charset=utf-8
User-Agent: Dalvik/2.1.0 (Linux; U;  Android 5.1.1; SM-J200G Build/LMY47X)
Host: npatracker.D1-Bank_1
Connection: close
Content-Length: 40
Customer Credentials Details:
{
"userId" : "admin",
"password": "admin123"
}
```

Figure 5.9: User_ID & password is shown as plain-text

(2) D2-Bank_2: This banking application requires the customer to register his mobile number for every new mobile device. The registered mobile number will get an OTP and validates the user for performing transactions. In this Android app, the customer has to request a mobile PIN (MPIN) to carry out the banking transactions. The application uses HTTP protocol for communication between the client app and a bank server. If the customer is not aware of the original server, the app may connect to another server exactly same as legitimate one. Based on this reason phishing attack is possible. Another possibility of attack is MitM, where an adversary can compromise the router and establish a secure connection with legitimate bank server and steals the information from the customer. The attacker gets a response from bank server against a query that can be modified or altered and forwarded to the app. The modified response is readily accepted by the MBA because one of the security features of message integrity is not verified in the app. If a customer initiates a transaction regarding account balance, then the client receives a response which is modified by the adversary, not by legitimate bank server. Moreover, two different mobiles can access same customer account, the response of balance inquiry will be different for each mobile device because one response is from the actual server and another one is from an adversary as shown in Figure 5.10 and Figure 5.11 respectively.

Original Balance Request	*Fake Balance Request*
POST http://mobile.D2-Bank_2/gprs/HTTP/1/1 Content-Type: text/plain User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.0.3; GT-I9100 Build/ML74K) Host mobile: D2-Bank_2 Connection: Close Accept-Encoding: gzip Content-Length: 63 MBSFTV0111 Bcv136,sLNDqzlvbE6WlaCDn3VDY0YZi3oDBaKfUUqUpjrvix0=	POST http://mobile.D2-Bank_2/gprs/HTTP/1/1 Content-Type: text/plain User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.0.3; GT-I9100 Build/ML74K) Host mobile: D2-Bank_2 Connection: Close Accept-Encoding: gzip Content-Length: 63 MBSFTV0111 Bcv136,sLNDqzlvbE6WlaCDn3VDY0YZi3oDBaKfUUqUpjrvix0=

Figure 5.10: Balance inquiry request comparison

Response to Original Balance Request	*Response to Fake Balance Request*
HTTP/1.1 200 OK DATE: Mon, 06 Mar 2017 05:32:54 GMT Server: Team(JaSuNa) X-Cache: MISS from 192.168.50.15 X-Cache-Lookup: MISS from 192.168.50.15:3128 Via:1.1 192.168.50.15 (squid/3.3.8) Connection: close Tid=080003165196, Msg=As of 06/03/17 total available balance of a/c XXXXXXXXXXX0860 is Rs.142	HTTP/1.1 200 OK DATE: Mon, 06 Mar 2017 06:18:07 GMT Server: Team(JaSuNa) X-Cache: MISS from 192.168.50.1 X-Cache-Lookup: MISS from 192.168.50.1:3128 Via:1.1 192.168.50.1 (squid/3.3.8) Connection: close Tid=080003171381, Msg=Request cannot be processed. Kindly Adjust Date/Time of your device or else goto settings and synchronize application

Figure 5.11: Balance inquiry response comparison

(3) D3-Bank_3: Initially, the customer is required to register a mobile phone number for mobile banking with strong cust_id (customer_id) and password. Consequently, the bank server sends an OTP to the registered mobile number of the client of a respective account holder, this is essential for authenticating the user at the server side. Here what we found that we capture the packet from the server, i.e., response, which contains the OTP as in plain-text format so it is possible that the adversary can perform MitM and session hijacking. The attacks as mentioned earlier are possible because bank server accepts fake certificates. The behavior of this app is shown in Figure 5.12. All the mobile banking apps, except D3-Bank_3, developed by the app developer of the D3 exhibit the same behavior as D3-Bank_4, hence no results produced.

```
POST/D3-BANK_3/servelt/NativeServlet01?ts=164672 HTTP/1.1
Content-Length: 164
Content-Type: application/x-www-form-urlencoded
Host: D3-BANK_3
Connection: close
Expect: 100-continue
Cookie: JSESSIONID=E6FB22137117580EED05A723C558ED45; LS
NONCEID=41e7a68b7e999af42f274724e9e2d343e28a0fb8dbdb1fe13fbf0a5868178872
Cookie2: $Version=1
BUILD_VERSION=14& USER_ID=215075231
&MOBILE_PIN=&FRM_OS=ANDROID&MOBILE_NUMBER=&METHOD_NAME=validateOTP&IMEI_NUM
BER=353327052273281&UNO=04785781714214074375&OTP=184176
```

Figure 5.12: OTP is in plain-text, not encrypted

(4) D4-Bank_5: For this app, we can perform account lock-out attack at the back-end server of a bank. But we couldn't execute session prediction attack because the corresponding server was responding unless if session id in the request made by the client is matched with session id's maintained by the server. To implement account lock-out attack by modifying the secure password and stimulate the lockout at the back-end server, even though the customer enters correct details of username and password. It is possible because the back-end server does not verify the data integrity in this case as opposed to login phase.

5.5.3 Static & Dynamic analysis for iOS MBAs

For iOS MBAs, to conduct a static analysis of the devices, testing tools, and testing framework are discussed in Section 5.3.1.2 and list specified in Table 5.2. If we take a look at OWASP Top 10 mobile risks [391], one of the top risks is insecure data storage. The iOS application interacts with the local system to store temporary as well as persistent data. The major risk arises when an application stores sensitive information insecurely in both persistent as well as temporary formats.
In iOS, there are different formats to store the data locally, as follows:

- SQLite files
- XML and plist
- Keychain data
- The NSUserDefaults class

- Core Data
- Log files

Each of this local storage has its own pros and cons. If the developers have not taken care of securing sensitive data, an attacker is more likely to access sensitive data on a device. An attacker having access to the rooted device can easily download application files and access user's confidential information.
The attacker uses the following capabilities using a threat model to determine and/or exploit existing vulnerabilities with the aim of avoiding security protections for iOS MBAs:

1. Jailbreak (iOS device): Jailbreaking is the privilege escalation from the mobile user to root user. Types of jailbreaks there are two types of jailbreak, described as follows:

   - Tethered: In tethered jailbreak, every time we reboot the device, we need to connect iDevice to the system to jailbreak it.
   - Untethered: In the untethered jailbreak, we need to jailbreak the device once and then we can use it all the time.

   In this security testing of MBAs, we jailbroke an iDevice running iOS 4s.

2. Reverse Engineering (MBA | Tool): Many times; an application developer makes the mistake of storing sensitive API keys and encryption keys at the client side. Now an attacker takes the opportunity, perform reverse engineering on the iOS application and convert to a source then find these encryption keys and other confidential data.

3. Install (Attacker configuration profile | Monitoring tools | SSL Certificate | Target device | MBA): It allows the attacker to install configuration file, self-signed certificates (SSL certificate), and monitoring tools on the target device. The target device is maybe (iOS device or proxy). The attacker can also load a configuration file on the target device because it indicates the location of the proxy in the network. It allows the attacker his presence is active in the network for implementation of MitM.

4. Method swizzling (Source code | MBA): It is changing the behaviour/functionality of a method at runtime. To perform method swizzling, we require information about the methods that are used in backend for the application. The idea as follows:

    - Reverse engineer the application, and attacker perform static analysis on the source code.
    - Call the appropriate method that is responsible for authentication.

5. Bypassing SSL pinning (Certificate | MBA): Many times, the attacker will not be able to intercept an application's HTTPs traffic, even after installing a self-signed (fake) certificate on the device. Nowadays, many applications implement SSL pinning, where an application checks the server's certificate against a known copy of that certificate. It prevents the use of any untrusted certificate that is being used for MitM attacks. The attacker bypass SSL pinning by using tools such as SSL Kill Switch. However, note that this is available only for jailbroken devices.

6. Encrypt (Information, Key): Permits the attacker to encrypt information using the key.

7. Decrypt (SSL certificate | HTTPS traffic | Files) used the attacker to decrypt the files certificates et., using the decryption key. The decrypted data will be stored in Attacker_State.

8. Re_Sign (K_x^+, K_x^-, SSL certificate) allows the attacker to modify the K_x^+ of the web server's certificate with a own key of attacker, which is then adversary used to Re_Sign the certificate with his K_x^-.

9. Capture (HTTPS traffic | SSL certificate | Session key): The attacker allows to intercept the network traffic between mobile device to bank web server and bank web server vice versa. The captured information is will be stored in the Adversary_State.

10. Access (Source files | Key): The attacker able to access target source files.

11. Extract (File, MBA | Key): The attacker captures the packet in an active network and extract the keys, or in the other case the attacker identifies the user's storage access and extracts the sensitive information.

12. Connect (User name, Password, MBA): It allows the attacker to establish a secure channel between MBA and backend server with a set of credentials (e.g., using the capability of extract and obtained the credentials).

13. SQL injection (MBA | bank server | iOS device) : It allows the attackers to tamper with existing data, spoof identity, cause repudiation issues such as changing balances, or voiding transactions and allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.

14. Intercept (SSL certificate | Network traffic | Key): One of the essential aspects of mobile communication level security is analyzing an application's network traffic between and installed MBA app in an iOS device and a back-end server. For example, the attacker intercepting traffic over HTTPs, need SSL certificate and handshake key. The intercepted information will be saved in the Adverary_State.

The attacker gained the knowledge on a set of goals, and successful execution of the above capabilities (e.g., captured packet information, keys, storage files, etc..,) will be saved in his/her Adversary_State.

5.5.3.1 Discussion

We now demonstrate how customer sensitive information leaked through various channels and attacker capabilities on various MBAs as shown in Table 5.5 and briefly discussed in following paragraphs.
(1) D1-iBank_1: In this MBA we successfully implement a MitM due to the lack of SSL pinning. The bank app does not verify the server's certificate which allows the attacker capabilities on D1-iBank_1 is Capture, Intercept, Bypassing SSL pinning, Access, etc. The attacker gains the legitimate access between the MBA and bank's server. During this analysis, the adversary intercepts the traffic and discover the following sensitive information of customer shown as plain-text format such as User-name, Account

number, MPIN (four digits), One Time Password (OTP). For security reasons, we are unavailable to provide results of D1-iBank_1.
(2) D2-iBank_2: We installed D2-iBank_2 on a jailbroken iDevice. To perform reverse engineer the application using "class-dump-z" and access the source code to get information. During static analysis, we used "Extract" attacker capability on various files whether customer's confidential data is saved in plist. Many iDevice app developers had mistakenly stored customer's credentials in the plist files. An attacker can quickly look into these plist files for user's credentials. For penetration of iOS apps the tester or analyzer will have to perform download/upload files with iDevice and Mac system using "Secure File Transfer Protocol (SFTP)". We download application files on iDevice to MAC using SFTP. The location of files specified in Table 5.5. Here we observe that all the customer's sensitive information that is filled in D2-iBank_2 app is stored insecurely as shown in Figure 5.13.
(3) D3-iBank_3: For this case study, the primary goal is to acquire the password and key which is used to Decrypt the sensitive documents stored in SQLite databases. In this app, the attacker capabilities are Decrypt, Capture, Intercept, Bypassing SSL pinning, Access, etc. First, we installed this app on a jailbroken device then perform both static (application), and dynamic (communication) analysis. NSUserDefaults is used for customization as per the user's preferences. Many times, the iOS app developer uses the NSUserDefaults format to store customer's confidential data. Core Data is an "object-relational mapping (ORM)" that creates a layer between user interface and database. The developers prefer Core Data as it is faster regarding record creation that the traditional SQLite format. From the security point of view, these files are similar to SQLite. For this D3-iBank_3 MBA, we test core data, i.e., stored in the SQLite database. The customer's details given in the app of D3-iBank_3, and use the submit button. Then downloading application files using SFTP. We observe that user's credentials stored in the plain-text format as shown in Figure 5.14.

Table 5.5: Static & dynamic analysis of iOS MBAs

Case study on iOS apps/Devices	D1-iBank_1	D2-iBank_2	D3-iBank_1	Jailbroken iDevice	SFTP	Mac Book Pro
Adversary capabilities used	• Jailbreak • Install • Re_Sign • Capture • Extract • Intercept • Bypassing SSL pinning • Access • Connect	• Jailbreak • Reverse Engineering • Method swizzling • Re_Sign • Extract • Intercept • Access	• Jailbreak • Reverse Engineering • Decrypt • Install • Method swizzling • Re_Sign • Capture • Extract • Intercept • Access	• Jailbreak • Reverse Engineering • Install • Method swizzling • Re_Sign • Capture • Extract • Intercept • Access	• Install • Extract • Intercept • Connect	• Adversary installs various tools to monitoring the network traffic such as BurpSuite Host, and Wireshark
Vulnerabilities found	• V6 : Insufficient transport layer protection	• V1 : Insecure Data Storage • V2 : Lack of Binary Protection	• V1 : Insecure Data Storage • V2 : Lack of Binary Protection • V3 : Unintended data leakage	Default root password unchanged	• V6 • V3	**NA**
Violation of security goals	• The app easily accepts self-certificates without SSL pinning	• An attacker can easily look into these plist files for sensitive information	• An attacker can quickly look into these NSUser Default class and Keychain data for sensitive information	• Compromise of sensitive information, device resulting unauthorized access	• Adversary successfully implementing MitM	**NA**
Type of user data/location	• /private/var/mobile/Applications/	• /private/var/mobile/Containers /Data/Application/	• /private/var/mobile/Library/Mail/ • /private/var/mobile/Library/SMS	**NA**	**NA**	• Adversary_State

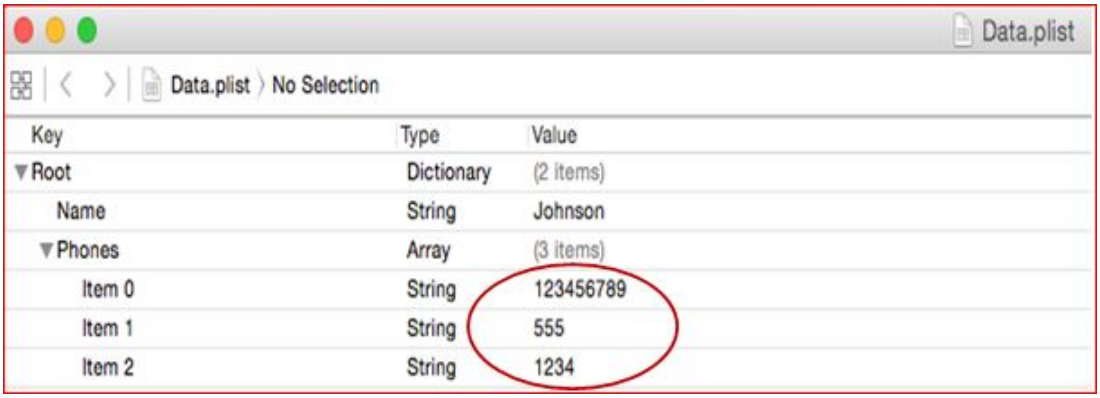


Figure 5.13: User's credentials are in plain-text in plist files

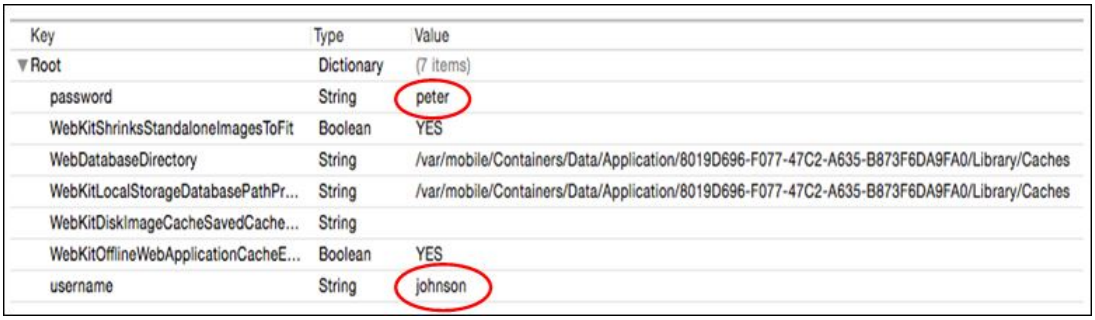


Figure 5.14: User's confidential data in plain-text

Based on the above observations, we make the following recommendations:

1. Customers must take care or avoid installing the MBAs from unofficial app stores as they might contain malware.

2. The app developers mistakenly stored customer's credentials and sensitive documents on iDevice local storage, such as plist, keychain, SQLite, NSUserDefault and so on.

3. All MBAs is having the facility of SSL pinning otherwise, it is chance for attacker implement MitM.

5.5.4 Compliance of MBAs to the OWASP

Table 5.2 summarizes our findings and maps the vulnerabilities identified with seven of the top 10 vulnerabilities detected by the OWASP mobile security project in 2014 [391].

- M1: Weak Server Side Controls
- M2: Insecure Data Storage
- M3: Insufficient Transport Layer Protection
- M4: Unintended Data Leakage
- M5: Poor Authorisation and Authentication

- M6: Broken Cryptography
- M7: Client Side Injection
- M8: Security Decisions via Untrusted Inputs
- M9: Improper Session Handling
- M10: Lack of Binary Protections

Most of the MBAs, we have identified M3 is a common design flaw. This flaw exposes an individual user's data and can lead to account theft. If the adversary intercepts an admin account, the entire site could be exposed. Poor SSL setup can also facilitate phishing and MitM attacks. Specially app developers should focus on security aspects on using HTTPs during connection establishment.

5.6 Summary

In this chapter, we have presented a threat model that helps systematically to test and analyze mobile banking applications. It helps to detect and mitigate the vulnerabilities at application and communication level. We have mainly focused on security testing of two dominant platforms, of Android and iOS Mobile Banking Applications. This chapter addresses various unknown vulnerabilities in MBAs and implementation of MitM attack during dynamic analysis of applications. With the results, we have showed that MitM attack is easily possible even though the application is run with the HTTPs protocol because of insufficient transport layer protection that leads to poor implementation of SSL framework at the bank server-side and mobile applications. We have performed security testing on various rooted, jailbroken and non-jailbroken devices, and identified common vulnerabilities. While in previous works authors determined some attacks but did not provide mitigation of vulnerability techniques. Every organization, financial institutions, banks, and third-party labs need to conduct the vulnerability assessment in depth as proposed, once in a year or twice for security inspection checks to the MBAs for detection of vulnerabilities and exploits. Hence this chapter addresses novel security aspects for MBAs development.

Chapter 6

Novel Authentication Protocol for Prevention of Phishing and Pharming Attacks in Mobile Banking

The current mobile authentication protocols place an extra burden on mobile device users to detect and avoid phishing and pharming attacks. In this chapter, we propose a novel mobile banking authentication protocol that employs an authentication server (AS), which sends a nonce to the mobile user device to be signed, so that he can avoid phishing and pharming attacks. Phishing attacks are fraudulent e-mail messages appearing to originate from legitimate enterprises to access private information and commit identity theft. Pharming is a form of attack on the network infrastructure where the user is redirected to the fraudulent website even when the user enters the right web "uniform resource locator (URL) or internet protocol (IP)" address. In this protocol, the message signed on the mobile device includes the information of the bank. Hence, upon receiving the signed information, the authentication server can quickly identify the attack and notify the bank. Our proposed protocol is experimentally validated by a formal method of model checking tool namely AVISPA.

6.1 Introduction

In the current digital communication scenario, both phishing and pharming attacks are increasing exponentially, with the purpose of stealing sensitive user information. Phishing attacks are possible on both desktop and mobile computing environments in many forms. Various phishing attacks in the desktop environment are fake advertisements, fake voice calls, bulk e-mails, and SMS with a fake web address. Mobile browser-based banking also suffers from phishing, pharming, and other associated attacks such as Man-in-the-Middle (MitM). The phishing attack is the most common attack, in which people receive spam or fake e-mails that attempt to access confidential data to commit fraud. In practical observation, the victims can receive phishing e-mails which redirect them to an adversary-created web page that looks similar to the legitimate website of the victim's service provider such as a university, financial institution, bank, etc. These similar looking web pages are designed to steal information or to perform a social engineering-based attack.

Although there are robust security measures implemented and public awareness is increasing rapidly, attackers are still able to perform phishing attacks successfully. The main reason for this is that the attackers use the latest professional and sophisticated equipment. When the phishing e-mail appears in the mailbox, the user may think that it is a legitimate e-mail from his service provider and inputs the personal information being requested, allowing, the attacker to steal the information and use it for fraudulent purposes. There is a possibility of introducing an illegitimate domain naming system (DNS) server into the network that can exploit a user's request for a web address and redirect it to a malicious web address when a device is connected to the broadband network through a wired or wireless medium. This scenario is possible when a DNS or IP routing mechanism is compromised. For example, when a mobile device is connected to an open wireless router (WiFi), there is a possibility of the occurrence of pharming through compromised DNS or IP routing that can redirect the user to a malicious website.

In this chapter, we propose a novel authentication protocol for the prevention of phishing attacks in the mobile environment. This protocol is analyzed using AVISPA. Formal verification and validation of cryptography protocols are needed because they identify vulnerabilities that may lead the proposed protocol to insufficiency, inconsistency,

or rupture. In AVISPA, the initial step is to write the proposed framework protocol in the "high-level protocol specification language (HLPSL)" to be executed by the AVISPA back-end servers. The next step is to test our protocol for all possible complete traces for the presence of deficiencies [350].

6.1.1 Organization of the chapter

The remaining sections of this chapter are organized as follows. Section 6.2 describes the proposed protocol. Section 6.3 presents the security analysis of the proposed framework. We conduct a test and analyze different banks as a case study in Section 6.4. Section 6.5 discusses implementation issues. Finally, section 6.6 concludes the chapter.

6.1.2 Motivation and Contributions

Phishing and pharming attacks pose the same threat to mobile phones as they do to personal computer (PC) platforms. Many mobile users trust their smartphones more than their PCs. But, these mobile phones are more vulnerable to phishing. Mobile applications lack security properties such as confidentiality, authentication, authorization, data integrity, and non-repudiation. So, mobile apps shold be less trusted than desktops [423] and the trust also depends on platform security of application permission in the environment of either Android or iOS. Websites in Mobile-based browsers are also unsafe. The mobile applications must follow the standard Same Origin Policy in [424], meaning that websites on any platform must be isolated from each other. In this study, we provide an authentication protocol for avoiding phishing and pharming attacks in any working domain.

For this chapter, we studied and examined a total of 21 mobile banking apps working under Android environment that is presently deployed by private and government sector banks in India, which are used by their corresponding users. The analysis conducted indicates that most of the mobile banking apps possess vulnerabilities and suffer from serious issues involved before login and after login to the mobile banking apps. Moreover, some banking apps use the simple HTTP protocol for performing transactions without considering security parameters such as confidentiality, authorization,

authentication, integrity, and non-repudiation. Further analysis regarding mobile banking apps is discussed in Section 6.4 as a case study, for security reasons we anonymize the names of the banks. Hence, our security analysis of mobile bank apps suggests that regular conduction of cybersecurity drills every twice/once in a year or a level of revising the bank apps is needed to direct the basic and advanced security flaws in MBAs.

6.1.2.1 Contributions

Most of the existing approaches for preventing phishing schemes fail to achieve their goals of network security and privacy preservation. These limitations have motivated us to analyze the current protocols for network attacks such as phishing and pharming. This chapter includes the following contributions:

- We implement an authentication protocol for avoiding phishing and pharming attacks.
- Through detailed formal and informal security analysis, we show that our proposed approach is secure against phishing and pharming attacks.
- We surveyed 21 mobile bank applications (MBAs) working under an environment of Android which is currently deployed by the government sector and private sector banks and used by their appropriate users.
- A systematic investigative approach in exploiting and detecting various unknown vulnerabilities is presented, particularly in mobile banking and other financial applications.
- We simulate our proposed approach with the verification model checking tool AVISPA, which rigorously analyzes our scheme and shows that our proposed approach is secure and safe.

6.2 Proposed Method

This chapter proposes a novel authentication protocol to prevent phishing and pharming attacks. This protocol uses the encryption mechanism of ECDSA [425] used for

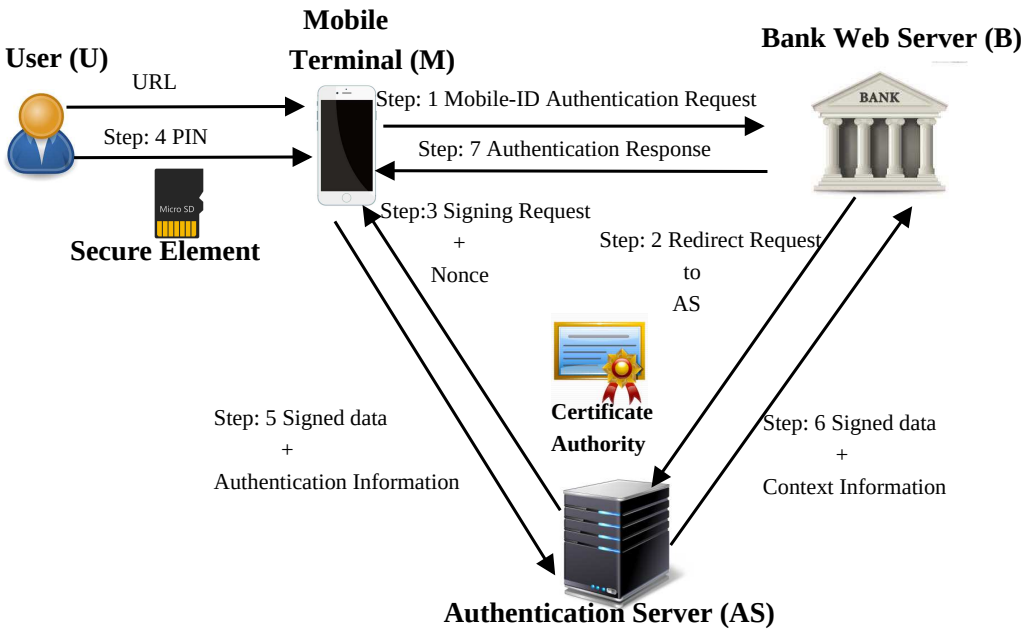


Figure 6.1: Proposed method in case of legitimate site

generating and verifying digital signatures and for encryption/decryption of messages using ECIES.

6.2.1 Proposed model for defending against phishing attacks

Figure 6.1 depicts the schematic diagram for the case of a legitimate site. The user interacts with the legal service provider (bank) through his mobile device, which contains the secure element (SE). The bank's web server interacts with a user through web pages. The user has a PIN associated with his mobile device, which is used to sign the nonce.

6.2.1.1 Entities and initial conditions

The notations used in this chapter is listed in Table 1.

- User U registers with the AS to obtain RC_U
- AS will have the RC_U of U
- Mobile browsing application of U is programmed for signing capability

- Public keys are accessible for all communicating entities
- The connection between bank and AS is over a secure channel (SSL/TLS)
- The user's mobile device M is accessed by a registered user
- Every communicating entity only has its own private key.

The following steps are necessary for a user to connect to a legitimate server which are presented in Table 6.1.

- **Step 1.** The user inputs the URL of the bank into the browser and clicks on an appropriate link within the bank website by giving his valid credentials such as the mobile ID..
  $B \leftarrow M : M_{id}, URL$
- **Step 2.** After providing valid credentials to the bank website, the bank site redirects the user to an authentication server (AS) over a secure connection.
  $AS \leftarrow B : M_{id}, B_{id}, Link$
- **Step 3.** AS sends a nonce N or a "one-time password (OTP)" to the mobile device of the user for the purpose of authentication.
  $M \leftarrow AS : AS_{id}, N$
- **Step 4.** After a nonce message is received from the authentication server, the user enters a PIN number into the mobile device.
  $M \leftarrow U : PIN$
- **Step 5.** At this time, the mobile user constructs a signature message consisting of authentication information (AI) and a nonce value which was already sent by the authentication server. The AI is appended to the random number or nonce. The AI specifies the bank site B, M is attempting to establish a connection. The authentic information could contain details of URL, IP address, a hash of digital certificates in SSL/TLS format, etc. Along with the signature information, a user also sends M_{id}. The format of authentication information is $\{h\,(N||AI)\}\,K_U^-$ which is hashed and signed with the private key of the user and verified by the AS with the public key of a user.
  $AS \leftarrow M : M_{id}, \left\{SIGN_M^{AS}(AI), AI\right\}$

- **Step 6.** AS performs signature verification [354] . Additionally, AS verifies whether the AI is correct or not because the bank website information and mobile user information could redirected in step 2. If the comparison is successful and the signature is verified, AS informs the bank via a signed message. This information is also sent by AS to certifying authority (CA) from bank B. The signed message consists of the context information (CI) of mobile M_{id}, identity information of bank B_{id} and time stamp value (TS) signed by AS. The signed hash message is encrypted with a private key of AS and is verified to originate from B.
  $\{h\left(M_{id}|| \, B_{id}|| \, TS\right)\} K_{AS}^{-}$
  $B \leftarrow AS : AS_{id}, \left\{SIGN_{AS}^{B}(CI), CI\right\}$

- **Step 7.** Bank B verifies the signed message sent by the AS for the confirmation of the successful authentication of the mobile user.
  $M \leftarrow B$: Successful Authentication.

Table 6.1: Proposed method against phishing attack

Mobile User	Bank	Authentication Server
Legitimate Site		
• [1] User needs to connect with the bank URL, for establish a connection between user's browser to B. For that user clicks on the appropriate link within the website of B which allows him/her to log in using mobile device authentication details. By this way, the user chooses an AS to log in. $\xrightarrow{M_{id}, URL}$		
	• [2] Bank server establishes a secure connection between B to the AS using SSL/TLS protocol, and bank server redirect user authentication request to AS over the secure connection. $\xrightarrow{M_{id}, B_{id}, Link}$	
		• [3] AS sends a nonce N or random number to mobile device of the user for authentication purpose. $\xleftarrow{AS_{id}, N}$
• [4] The mobile user enters his/her PIN number into the device		
• [5] Now mobile user constructs a signature message consists of AI and Nonce value which is already sent by the authentication server. The AI specifies the bank site B, a hash of digital certificates in SSL/TLS format, etc. The format of AI is $\{h(N\|\|AI)\}K_U^-$ which is hashed and signed with the private key of the user and verified to be AS. $\xrightarrow{M_{id}, \{SIGN_M^{AS}(AI), AI\}}$		
		• [6] AS compares the data from bank site which is given redirection in step 2. If the comparison is successful then AS informs to the bank by a signed message via CA to bank B. The Context Information (CI) consists of M_{id}, B_{id} and TS for avoiding replay attacks. The signed hash message is encrypted with private key of AS and verified to be B. $\{h(M_{id}\|\|B_{id}\|\|TS)\}K_{AS}^-$ $\xleftarrow{AS_{id}, \{SIGN_{AS}^{B}(CI), CI\}}$
	• [7] Now bank sends successful authentication to mobile user. $\xleftarrow{\text{Success}}$	

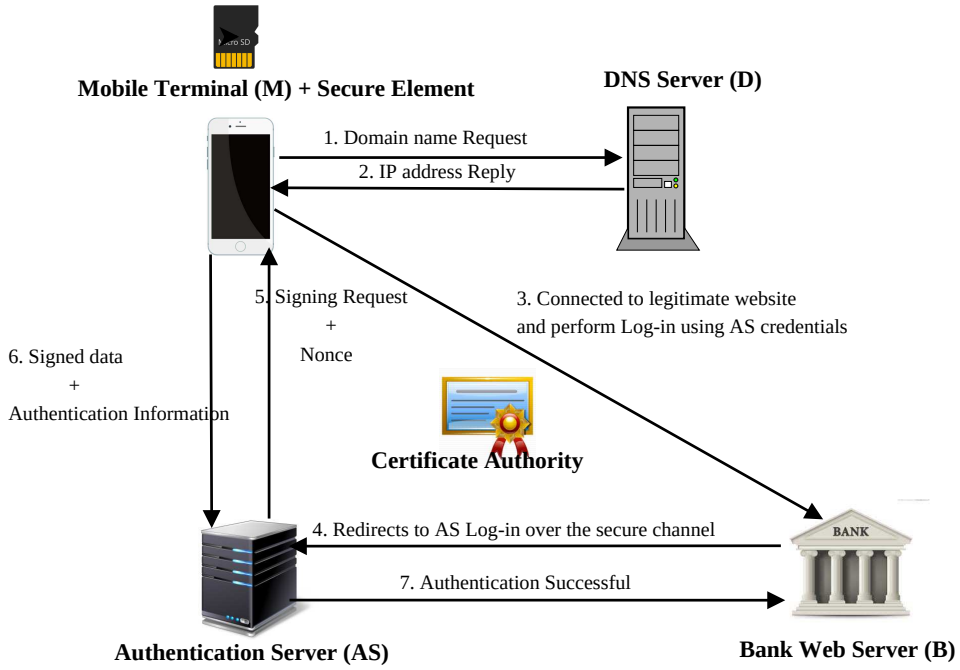


Figure 6.2: Proposed method for defending against pharming attack

6.2.2 Pharming attack

Figure 6.2, depicts how the user interacts with the legitimate service provider (bank) through his mobile device, which contains the SE. The bank has a web server which interacts with the user through web pages. When the user enters a legitimate URL, the DNS translates the corresponding URL to its legitimate IP address.

6.2.2.1 Initial conditions

- U registers with AS to obtain RC_U.
- AS will have the RC_U of U.
- U connects to a broadband wireless connection.
- Mobile browsing application of U is programmed for signing capability.
- Public keys are accessible to all communicating entities.
- Connection between bank and AS is over a secure channel (SSL/TLS).
- M is accessed by a registered user.
- Every communicating entity has its own private key.

6.2.2.2 Protocol steps

The sequence of steps of the scheme against a pharming attack is presented below and is also presented in Table 6.2.

- **Step 1.** User enters the URL of the bank into the browser to request the nearest DNS server for the corresponding IP address of the URL.
  $D \leftarrow M : D_{req}, URL$

- **Step 2.** The DNS server looks up the corresponding IP address of the URL and sends the IP address in reply to the browser.
  $M \leftarrow D : D_{res}, IP$

- **Step 3.** Browser connects the user to the related web page of IP address. User clicks on the appropriate link within the website of the bank which allows him to log in using AS credentials.
  $B \leftarrow M : M_{id}, Link$

- **Step 4.** Bank's website redirects the user request to AS over the SSL/TLS secure connection.
  $AS \leftarrow B : B_{id}, Link$

- **Step 5.** AS sends a nonce N or random number to the mobile device of the user for the purpose of authentication.
  $M \leftarrow AS : AS_{id}, N$

- **Step 6.** At this moment, the mobile user constructs a signature message composed of AI and the Nonce value already sent by the AS. The AI is appended to the random value or nonce. The authentication information specifies the bank site B with which M is attempting to establish a connection. The authentic information could contain the details of URL, IP address, a hash of digital certificates in SSL/TLS format, etc. Along with the signature information, the user also sends the M_{id}. The format of authentication information is
  $\{h\,(N||AI)\}\,K_U^-$ which is hashed and signed with the private key of the user and verified by AS with the public key of the user.
  $AS \leftarrow M : M_{id}, \left\{SIGN_M^{AS}(AI), AI\right\}$

- **Step 7.** AS performs signature verification using the certificate validation procedure. as described in cryptography theory and practice by [354]. Additionally, AS verifies whether the AI is correct or not because the bank website information and mobile user information are given as a redirection in step 2. If the comparison is successful and the signature is verified, then AS informs the bank, as shown in Figure 6.3.

- **Step 8.** $M \leftarrow B$: The bank sends the mobile user as as a successful authentication response otherwise and a authentication fail message is sent.

Table 6.2: Proposed method against pharming attack

Mobile User	DNS	Bank	Authentication Server
Legitimate Site			
• [1] The user determines the URL of the bank into the browser. It requests the nearest DNS server for the respective IP address of the URL. $\xrightarrow{D_{req}, URL}$			
	• [2] The DNS server lookup for the corresponding IP address of the URL and sends IP reply to the browser $\xleftarrow{D_{res}, IP}$		
• [3] Browser connects the user to the respective IP address webpage. User clicks on the appropriate link within the website of Bank which allows him/her to login using AS credentials. $\xrightarrow{M_{id}, Link}$			
		• [4] Bank's website redirects the User to AS over the secure connection. $\xrightarrow{M_{id}, B_{id}, Link}$	
			• [5] AS sends a nonce N or random number to a mobile device of the user for the purpose of authentication. $\xleftarrow{AS_{id}, N}$
• [6] The mobile user constructs a signature message consists of AI and Nonce value which is already sent by the AS. The AI specifies the bank site, mobile user-id, a hash of digital certificates in SSL/TLS format, etc. The format of AI is $\xrightarrow{M_{id}, \{SIGN_M^{AS}(AI), AI\}}$			
			• [7] AS compares the data from bank which is given redirection in step 2. If the signature verification is successful then AS informs to the bank as authentication Success $\xleftarrow{}$
	• [8] Now bank sends Authentication success $\xleftarrow{}$		

6.2.3 Adversary $\mathcal{A}$ model

Theorem 1 *Our proposed framework for prevention of phishing attack is secure against an illegitimate site created by an adversary to impersonate a legitimate site and commit fraud.*

Proof: To verify our scheme in the attack scenario, we aim to build an adversary $\mathcal{A}$, who establishes a session with the legitimate bank server B and the mobile user M. The attacker who created the fake web page tries to access user credentials through his web page which exactly (or very similarly) looks like the legitimate web page. As the valid page is redirected to the AS, the bogus site also mimics the same action and redirects to the AS. Then, based on the authentication information and signed information, the AS distinguishes between the malicious web page and the legitimate web page. Figure 6.3: depicts how the user interacts with the bogus service provider (bank) through his mobile device, which contains the SE. Consider the experiment Algorithm 4. It is clear that if the fake website of a $\mathcal{A}$ can acquire all the confidential data from the user and perform a phishing attack. AS a result, our framework not only detects the phishing attack through authentication of the various entities.

Theorem 2 *Our proposed framework for protecting from pharming attacks is secure against compromised DNS.*

Proof: Figure 6.4, depicts how the user interacts with the compromised DNS server (Attackers control) through his mobile device, which contains the SE. The DNS server redirects the user to the attacker's web page even if the legitimate URL is entered by the user. The fake web page through the DNS server tries to access the user credentials through its web page which looks like the legitimate web page. As the correct page redirects to the AS, the attacker also mimics the same action and redirects to the AS. Then, based on the IP address in the AI and the signed information, the AS detects the pharming attack. Algorithm 5 describes the complete sequence of steps.

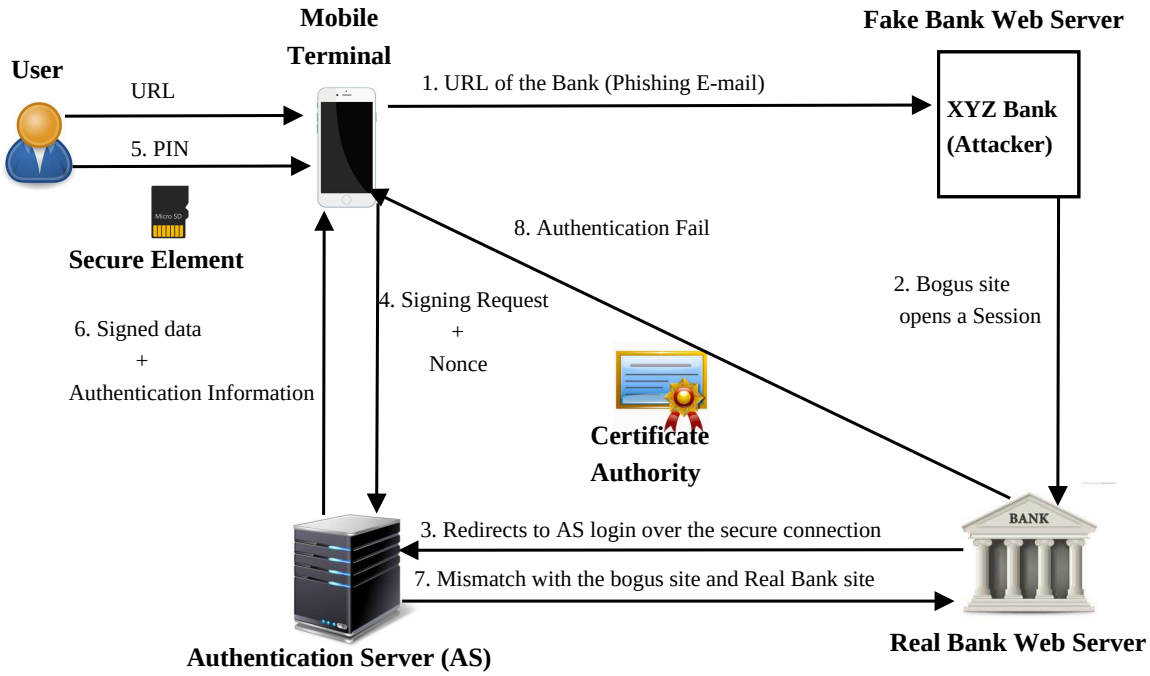


Figure 6.3: Proposed method in case of bogus site for prevention of phishing attack

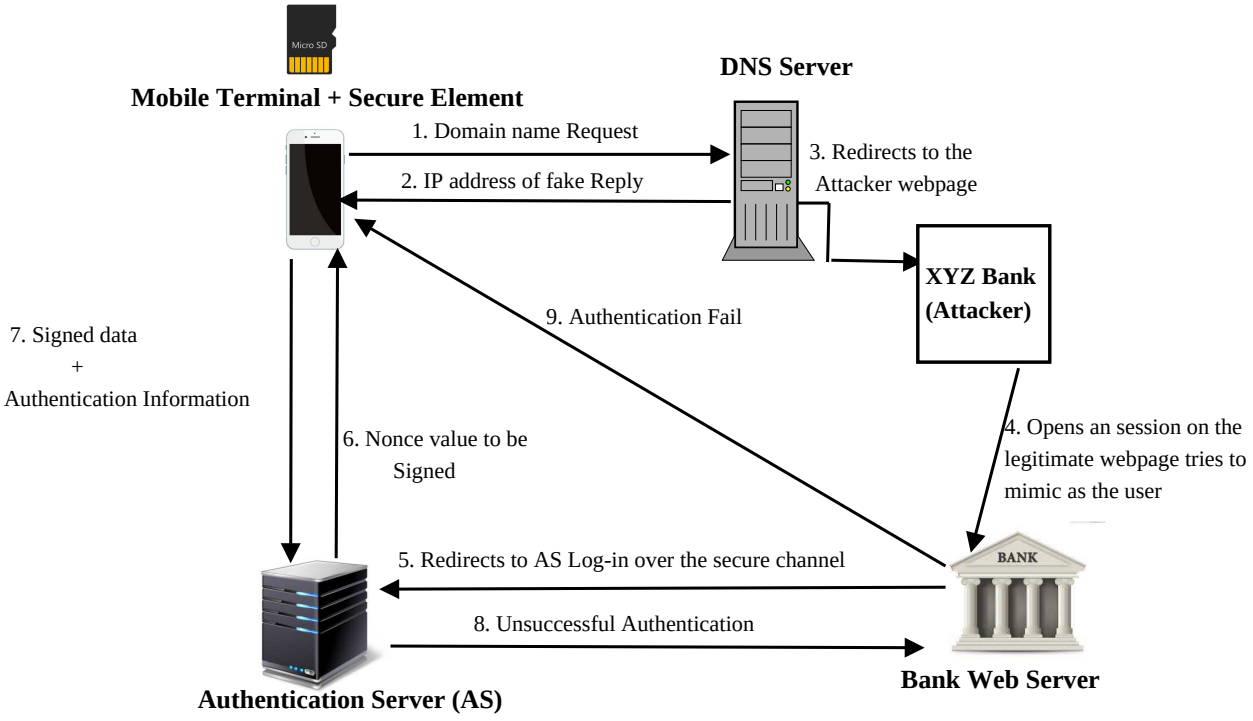


Figure 6.4: Proposed method in case of bogus site for prevention of pharming attack

Algorithm 4 $PHISH_{\mathcal{A},B,AS}^{SIGN,MU}$

1: User determines the adversary $\mathcal{A}$ of bogus URL of the bank through phishing e-mail into the browser. Provide his/her valid credentials on to the bogus site of the bank. For example, the user receives a phishing email that impersonates to be his/her bank as *ubank.com* but contains a link to the bogus site of *ubbank.com*. Now user provides valid credentials to the adversary $\mathcal{A}$ of bogus URL of the bank. $\mathcal{A} \leftarrow M : M_{id}, URL$

2: The bogus website of adversary $\mathcal{A}$ immediately establish a session to the legitimate bank website by impersonating the user and to commit fraud. Since user chooses to use AS credentials, the fake website has no other choice to mimic the genuine site.

3: Bank's website redirects the User to AS over the secure connection. $AS \leftarrow B : M_{id}, B_{id}, Link$

4: Upon the correct log-in credentials AS sends a nonce value to Mobile Device (M), which is to be signed. $M \leftarrow AS : AS_{id}, N$

5: The signed nonce number will be used for authentication purpose. In order to construct a digital signature, User is required to enter the correct PIN number to the mobile device.

6: Now user constructs a signed message of Authentication Information (AI) $AS \leftarrow M : M_{id}, \left\{SIGN_M^{AS}(AI), AI\right\}$. The format of authentication information is $\{h\,(N||AI)\}\,K_U^-$ which is hashed and signed with the private key of the user and verified by AS with the public key of a user.

7: After received a signed message from a mobile user, the AS decrypt with the public key of the user and performs signature verification. AS validates the user's certificate by using certificate validation procedure [354]. **if**$(Verification\left\{SIGN_M^{AS}(AI)\right\} = 1)$ Authentication & Integrity of AI is not compromised

8: **return** 1 (Success) else

9: **return** 0 (Failure).

10: Now AS compares information provided by a user and the bank. From this comparison user is connected to *xbbank.com* instead of *xybank.com*. This mismatch easily identified by AS, linking between the genuine and fake websites and informs *xybank.com* regarding this event occurring. As a result, our scheme prevents the real-time phishing attack in mobile banking applications.

Algorithm 5 $PHARM^{SIGN,MU}_{A,DNS,B,AS}$

1: User determines the URL of the Bank into the browser. It requests the nearest DNS Server for the respective IP address of the URL.
$D \leftarrow M : D_{req}, URL$

2: The DNS Server lookup for the corresponding IP address of the URL and redirects to attackers web server IP address as a reply to the browser request.
$M \leftarrow D : D_{res}, IP$

3: The DNS server redirects the user even after entering the legitimate URL.

4: The attacker opens a session on the legitimate web page to mimic the user action since User chooses link to use AS credentials, the fake site has no other choice to imitate the genuine site, so the real site redirects the attacker to AS.

5: Bank's website redirects the User to AS over the secure connection.
$AS \leftarrow B : B_{id}, Link$

6: AS sends a nonce N or random number to a mobile device of the user for authentication purpose.
$M \leftarrow AS : AS_{id}, N$

7: After receiving a nonce value from the AS, now user constructs an Authentication Information (AI) is attached to the nonce number. The AI specifies the bank site B, M is able to establish a connection. The AI could also contain other details of URL, IP address, a hash of digital certificates in SSL/TLS format, etc.
The format of authentication information is $\{h\,(N||AI)\}\,K_U^-$ which is hashed and signed with the private key of the user and verified by AS with the public key of a user
$AS \leftarrow M : M_{id}, \left\{SIGN^{AS}_M(AI), AI\right\}$

8: After received a signed message from a mobile user, the AS decrypt with the public key of the user and performs signature verification. AS validates the user's certificate by using certificate validation procedure[354].
if$(Verification\left\{SIGN^{AS}_M(AI)\right\} = 1)$
Authentication & Integrity of AI is not compromised

9: **return** 1 (Success) else

10: **return** 0 (Failure).
AS performs the signature verification and Authentication Information comparison it notices the mismatch between the real and bogus websites and informs bank for this incident.

6.3 Security Analysis of the Proposed Framework

In this section, we describe the formal verification approach using the AVISPA tool simulation. A detailed explanation of AVISPA-HLPSL is presented in the Appendix A.4.

6.3.0.1 Analysis of results

Once our protocol is written in the HLPSL specification, it is checked automatically to detect various attacks. AVISPA is integrated with four back-ends, which perform a rigorous security analysis of the protocol and provide the validation results. The output obtained is described in the following sections.

6.3.0.2 AVISPA results for securing against phishing attack

1. SUMMARY describes whether a tested protocol is safe, unsafe, or whether the security analysis is inconclusive.
2. DETAILS outlines the constraints under which the analyzed protocol is expressed as safe and what conditions have been used for identifying an attack.
3. PROTOCOL specifies the name of the protocol that is being tested.
4. The GOAL, BACKEND, and COMMENTS are the goal of the protocol, name of the AVISPA back-end used, and comments regarding the test, respectively.
5. Finally, STATISTICS presents the complete analysis as a statistical report of protocol evaluated.

Our proposed framework runs within all back-ends of AVISPA. For an execution test, the protocol analysis is successful. No susceptibilities to attacks were found, and the security goals were reached. So based on AVISPA results, our proposed framework is secure and safe from phishing attacks. Individual tool results are depicted in Figure 6.5 and Figure 6.6 respectively.

- "On-the-fly Model-Checker (OFMC)" reports our proposed framework is in a safe state.
- "CL-based Attack Searcher (CL-AtSe)" reports our proposed framework is in the safe state.

```
root@admin:~/Desktop# avispa phishing.hlpsl --ofmc
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /opt/avispa-1.1/testsuite/results/phishing.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.12s
  visitedNodes: 64 nodes
  depth: 6 plies
root@admin:~/Desktop#
```

Figure 6.5: AVISPA-OFMC result for phishing attack

```
root@admin:~/Desktop# avispa phishing.hlpsl --satmc
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH

PROTOCOL
  phishing.if

GOAL
  %% see the HLPSL specification..

BACKEND
  SATMC

COMMENTS

STATISTICS
  attackFound               false     boolean
  upperBoundReached         true      boolean
  graphLeveledOff           1         steps
  satSolver                 zchaff    solver
  maxStepsNumber            30        steps
  stepsNumber               1         steps
  atomsNumber               0         atoms
  clausesNumber             0         clauses
  encodingTime              0.0       seconds
  solvingTime               0         seconds
  if2sateCompilationTime    0.24      seconds

ATTACK TRACE
  %% no attacks have been found..
```

Figure 6.6: AVISPA-SATMC result for phishing attack

- "SAT-based Model-Checker (SATMC)" reports our proposed framework is in a safe state.
- "Tree Automata-based Protocol Analyser (TA4SP)" reports that some rules stated in the protocol may be not fired, so it does not perform the verification.

No attacks were found to be successful, and the security goals were reached.

6.3.0.3 AVISPA results for securing against pharming attacks

Once our protocol is written in HLPSL specification, it is checked for detecting various pharming attacks. AVISPA integrates with four back-ends, which perform the rigorous security analysis of the protocol and provide the validation results. The AVISPA code is given in Appendix (A.3). Our proposed framework is run within all back-ends of AVISPA. For an execution test, the protocol analysis was successful. No revealed attacks were found, and the security goals were reached. Thus our proposed framework is secure and safe from pharming attacks. Individual tool results are depicted in Figure 6.7 and Figure 6.8.

- AVISPA back-end of "On-the-fly Model-Checker (OFMC)" reports our proposed framework is in the safe state.
- AVISPA back-end of "CL-based Attack Searcher (CL-AtSe)" reports our proposed framework is in a safe state.
- AVISPA back-end of "SAT-based Model-Checker (SATMC)" reports our proposed framework is in the safe state.
- AVISPA back-end of "Tree Automata-based Protocol Analyser (TA4SP)" reports that some rules stated in the protocol may be not fired, so it does not perform the verification.

No successful attacks were found, and the security goals were reached.

```
root@admin:~/Desktop# avispa pharming.hlpsl --ofmc
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /opt/avispa-1.1/testsuite/results/pharming.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.81s
  visitedNodes: 256 nodes
  depth: 8 plies
```

Figure 6.7: AVISPA-OFMC result for pharming attack

```
root@admin:~/Desktop# avispa pharming.hlpsl --satmc
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  BOUNDED_SEARCH_DEPTH
  BOUNDED_MESSAGE_DEPTH
PROTOCOL
  pharming.if
GOAL
  %% see the HLPSL specification..
BACKEND
  SATMC
COMMENTS
STATISTICS
  attackFound               false     boolean
  upperBoundReached         true      boolean
  graphLeveledOff           1         steps
  satSolver                 zchaff    solver
  maxStepsNumber            30        steps
  stepsNumber               1         steps
  atomsNumber               0         atoms
  clausesNumber             0         clauses
  encodingTime              0.02      seconds
  solvingTime               0         seconds
  if2sateCompilationTime    0.56      seconds
ATTACK TRACE
  %% no attacks have been found..
```

Figure 6.8: AVISPA-SATMC result for pharming attack

6.3.1 Security characteristics compared with related schemes

In this section, we compare the security features of the proposed protocol with other related works. Table 6.3, presents a comparison of security and design features provided

by the proposed protocol and other relevant frameworks for securing against phishing and pharming attacks. Some of the related works only identify phishing and pharming attacks, but do not avoid these attacks. The proposed method has the potential to both identify and avoid phishing and pharming attacks.

Table 6.3: Summary of resilience to the attacks

Related works /Attack	**[426]**	**[427]**	**[428]**	**[423]**	**[429]**	**[430]**	**[431]**	**[432]**	**[433]**	**[434]**	**[435]**	**[436]**	Proposed protocol
Protect from phishing	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
Prevention of pharming	Yes	Yes	Yes	No	No	No	No	No	Yes	No	Yes	Yes	Yes
Resilient against Session Hijacking	Yes	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	Yes	Yes
Secure against MitM	Yes	No	No	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Prevention of Replay	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Protect from Masquerade	Yes	No	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Prevention of Wi-Fi sniffing	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Secure against Traffic analysis	No	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Prevents Eavesdropping	No	No	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Avoid DoS	No	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Formal Verification by AVISPA	No	No	No	No	No	No	No	No	No	No	No	No	Yes

6.4 Case Study

We have taken 21 mobile banking applications (banking apps) that are currently deployed by the nationalized public and private sector banks in India [437] and are used by their respective customers. However, for security and reputation reasons, we unknown the exact names of the banks whose banking apps we have considered for testing. Table 6.4, shows some of the mobile baking services of mobile banking applications(MBAs) developed by outside developers called vendors, the name of the developer also anonymized as D1, D2, etc. The objective of our investigation for mobile bank applications is exploiting the phishing, pharming, and other associated attacks. For this testing, the tester simulates the real attacker and targets the victims and bank servers. We have identified that an adversary can be easily implemented MitM setup even if the bank apps are running under HTTPs environment for establishing a communication channel and maintain sessions with their bank back-end servers. More-

over, some bank apps use the HTTP protocol for conduction of financial transaction and transfer customer information without considering security. The more case studies with the results of different MBAs also discussed in previous chapter 5.4.

Table 6.4: Data Set, (D_i indicates developer D) for an MBA

Bank_App	No. customers downloaded	MS	MP	BI	FT
D1-Bank_1	500,000	Yes	No	Yes	Yes
D1-Bank_2	500,000	Yes	Yes	Yes	Yes
D1-Bank_3	5,000,000	Yes	Yes	Yes	Yes
D1-Bank_4	500,000	Yes	Yes	Yes	Yes
D1-Bank_5	500,000	Yes	No	Yes	Yes
D1-Bank_6	50,000	Yes	Yes	Yes	Yes
D1-Bank_7	500,000	Yes	Yes	Yes	Yes
D1-Bank_8	500,000	Yes	Yes	Yes	Yes
D2-Bank_1	1,000,000	Yes	No	Yes	Yes
D2-Bank_2	5000000	Yes	No	Yes	Yes
D3-Bank_1	500,000	Yes	Yes	Yes	Yes
D3-Bank_2	10,000,000	Yes	Yes	Yes	Yes
D4-Bank_1	1,000,000	Yes	Yes	Yes	Yes
D5-Bank_1	500,000	Yes	No	Yes	Yes

MS: Mini Statement, MP: Merchant Payment, BI: Balance Inquiry, FT: Funds Transfer,

6.5 Implementation

Most mobile banking apps are developed by a third party or vendor or developer only, so the possibility of attacks is easier, as discussed in Section 6.4; hence our proposed solution is implemented in a secure environment as shown in Figure 6.9. The mobile user communicates with the bank using the Internet. The bank and authentication server communicate in an SSL/TLS environment and authentication server and mobile user use an SMS authorization model. In this model, the authentication server sends

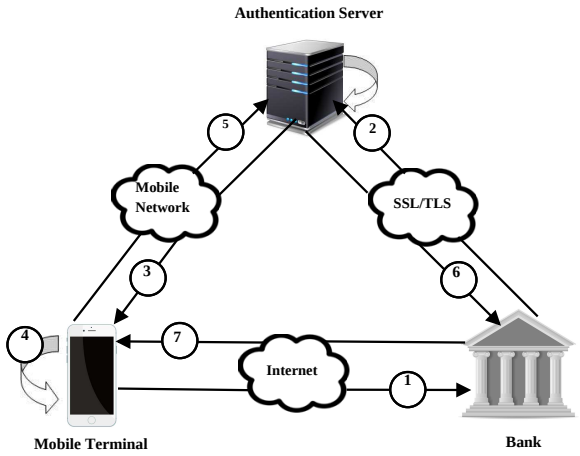


Figure 6.9: Protocol steps for secure against phishing

any SMS messages to the user's mobile terminal through the cellular network, which is different and separate from the Internet. By verifying the nonce or OTP received from the user's mobile terminal, the authentication server concludes that the mobile user received the SMS message through the cellular network, read it, and constructed a message and submitted it through the Internet. The security of this approach is based on the belief that it is difficult for an adversary to steal the user's mobile device and to attack the mobile network directly using the stolen device [438].

6.5.1 Experimental setup

The proposed solution is implemented as a pilot project to secure against phishing and pharming at the web servers of the bank and AS on the server side and assuming the mobile banking browsing application of a user is programmed for signing capability. Every mobile banking app is assumed to have these facilities before user registration with a proper mobile banking website. For signature generation and verification using ECDSA, HTML5 is used for structuring and presenting the contents on a web page. As HTML5 has come up with many new syntactic features, For example, the API's and "document object model (DOM)" are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for any invalid documents. "Cascading style sheets (CSS)" are used for describing the presentation of documents written in HTML. For a mobile website, various frameworks are available, and we have

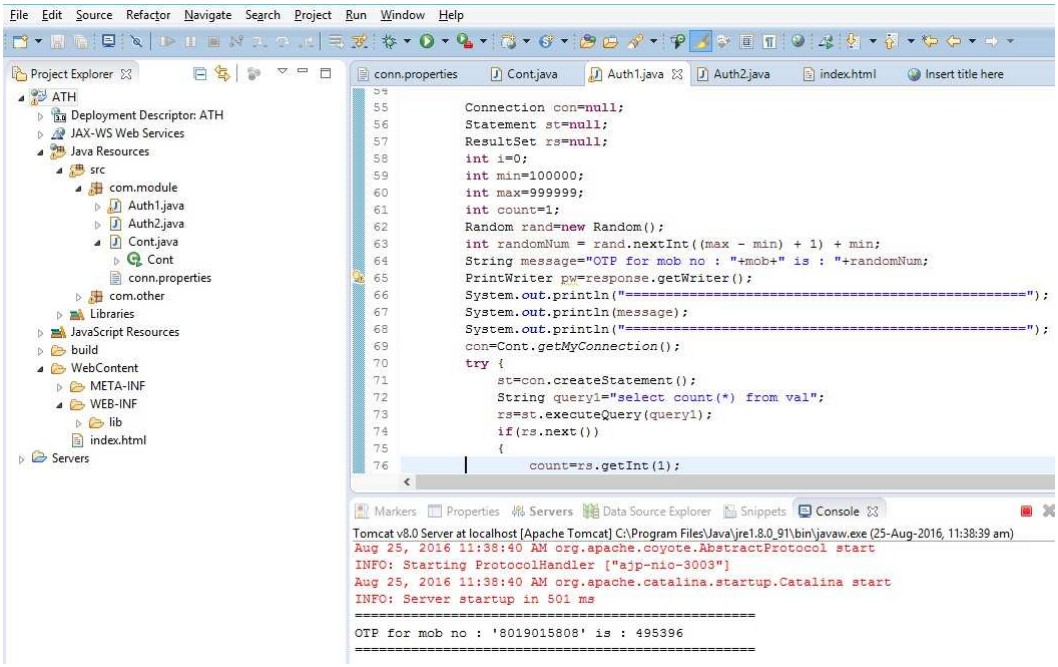


Figure 6.10: Authentication server generates OTP

used the Jquery mobile website development framework to develop our mobile website. It is a touch-optimized framework, more specifically a JavaScript library, which provides various classes which we can use with our HTML5 code. Using the classes of the Jquery mobile framework assists in loading the contents per the framework's mobile website design standards.

By using Javascript, CSS and Jquery mobile framework properties are enabled in our HTML page, and Java servlets, which are Java programs that extend the capabilities of a server, are adopted. Although servlets can respond to many types of requests, they most commonly implement applications hosted on web servers. Servlets3.0 is used for writing the back-end coding to perform user validations for the database and back-end operations. Servlet handled requests coming from a web page are processed and forwarded to another server, HTML page, JSP page, or another servlet. Java server pages (JSP) are used as a medium for writing Java code and HTML code together in one file. So for checking validations and then providing the Web page, we use JSP.

Screen shots of the proposed framework for securing against phishing attacks are shown in Figure 6.10, 6.11, 6.12, 6.13, 6.14, and 6.15.

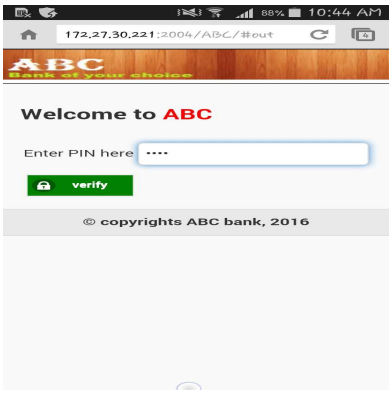


Figure 6.11: Mobile PIN input page

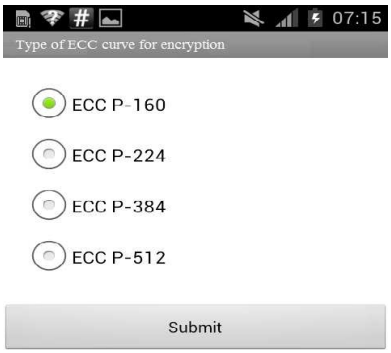


Figure 6.12: User select type of ECC curve for signature encryption

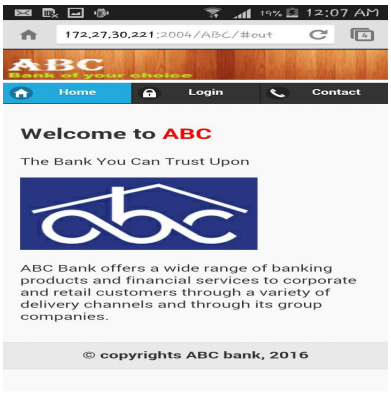


Figure 6.13: ABC bank home page

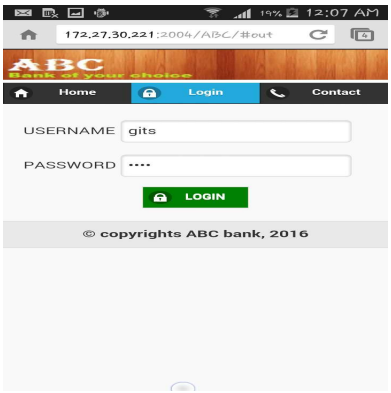


Figure 6.14: Login page for ABC mobile banking

```
==========================================================

==    REDIRECTED TO AUTHENTICATION SERVER :              ==
==    http://172.27.30.114:3002/ATH/Auth1?mo='8019015808'==

==========================================================
```

Figure 6.15: Request redirected to AS

6.6 Summary

In this chapter, we have presented a novel authentication mechanism that can assist mobile banking users in avoiding phishing and pharming attacks. It can also identify the occurrence of an attack. The proposed approach makes use of an authentication server that verifies the signed data to overcome the potential of both phishing and pharming attacks. The security analysis of the proposed algorithms was carried out using AVISPA, which demonstrated that the proposed solution is feasible in practice and the results obtained proved that the protocol is safe and secure. We believe that our study and analysis will be beneficial to end users, researchers, mobile app developers, bankers, and financial institutions. Our case study results showed the potential to improve the security practices in real-time mobile app development and security testing.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

Mobile phones have the power to enable payments anywhere and at any time, but the existing mobile payment solutions do not ensure end-to-end security. This thesis proposes secure mobile payment protocols suitable for wireless environment, which ensure end-to-end security at the application layer for financial transactions. The proposed mobile payment protocols use digital signature mechanisms based on ECC.
The proposed mobile payment protocols ensure robust security properties such as authentication, data integrity, confidentiality, and non-repudiation, and have been tested using various static and dynamic analysis tools so that they achieve payment secrecy, forward secrecy, order secrecy, and prevents overspending, money laundering, and double spending. In addition to this, our proposed mobile payment protocols withstand replay, MitM, and impersonation attacks. The security properties of the proposed protocols in this thesis are successfully verified using BAN logic, AVISPA, and Scyther tools. All the proposed mobile payment protocols are compared with known related protocols and found to perform better than the existing ones regarding communication cost, computational cost, and the number security properties ensured. Thus this thesis enhances security in the mobile payments field.
Following are brief discussions of the unique characteristics of our proposed protocols in this thesis.

7.1.1 End-to-End security

Security on the endpoints (client-client or client-server for peer-to-peer (P2P)) is an absolute requirement for secure communication and transactions. Such a solution contains the following components:

- Identity: This component encompasses known and verifiable entity identities on both ends; note that an identity can be temporary for a connection. For example, a user often is identified by user-name and password, whereas a server may be identified through a server certificate.
- Protocols: Protocols for example, SSL/TLS are used to dynamically negotiate session keys, and to provide the required security functions (for example, encryption and integrity verification) for a connection. Protocols use algorithms to implement these functions.
- Algorithms: Algorithms (for example, "Elliptic Curve Cryptography (ECC)", "Advanced Encryption Standard (AES)", and "Secure Hash Algorithm (SHA-1))" use the previously mentioned session keys to protect data in transit, for example through encryption or integrity checks.
- Secure implementation: The endpoint (client or server) that runs one of these protocols must be free of attacks that could compromise security. Also, malware can compromise security, for example by logging keystrokes on a PC.
- Secure operation: Users and operators have to understand the security mechanisms, and how to deal with exceptions. For example, web browsers warn about invalid server certificates, but users can override the warning and still make the connection. This concern is a nontechnical one but is of critical concern for security today.

Based on the above security components, In our thesis, we have proposed and developed protocols for mobile payments and banking to achieve end-to-end security.

7.1.2 Optimal consumption of resources

The proposed secure mobile payment protocols are designed to use a digital signature based on ECDSA. The advantage of ECC compared to other public-key cryptosystems are higher speed, lower power consumption, bandwidth savings, storage efficiency and adaptable certificates; hence, ECC is suitable for resource-constrained devices like mobile phones and are applicable for lightweight cryptography protocols.

7.1.3 Formal verification

We have carried out formal verification of all proposed mobile payment protocols in this thesis using various approaches: BAN logic, AVISPA, and Scyther tools. The security analysis using formal verification methods verifies that our proposed protocols are safe and withstand various attacks such as MitM, replay, and impersonation attacks.

7.1.4 Security testing

Our proposed protocols are verified using static and dynamic analyses tools such as Drozer, VirusTotal, apkanalyzer, Dex2Jar, etc. From this security testing, we resolve various attacks involving SQL-injection, improper session handling, unintended data-leakage, etc.

7.2 Future Directions

Research and development is a continuous process. Although the main objectives concerning in this thesis were accomplished, there remain possibilities for the enhancement of this presented work and future lines of investigation.

- To make the proposed protocols more secure, biometric-based authentication can be added.
- Our proposed protocols can be adopted in mobile ad-hoc networks (MANET) based payments using mobile agents.
- An automated testing framework dedicated to mobile banking applications may be developed.

List of Publications

[1] Sriramulu Bojjagani, V.N. Sastry, "A Secure End-to-End SMS-based Mobile Banking Protocol", *International Journal of Communication Systems (IJCS), Vol. 30, No. 15, pp. 1-19, Wiley, 2017.*

[2] Sriramulu Bojjagani, V.N. Sastry, "SSMBP: A Secure SMS-based Mobile Banking Protocol with Formal Verification", *In Proceedings of the* 11^{th} *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. (WiMob'2015)* Abu Dhabi, UAE, pages 252-259, Oct 19-21, 2015.

[3] Sriramulu Bojjagani, V.N. Sastry, "STAMBA: Security Testing for Android Mobile Banking Apps", *In Proceedings of the* 2^{nd} *International Symposium on Signal Processing and Intelligent Recognition Systems (SIRS'2015)* Trivandrum, India, pages 671-683, December 16-19, 2015.

[4] Sriramulu Bojjagani, V.N. Sastry, "VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps", *In Proceedings of the* 3^{rd} *IEEE International Conference on Collaboration and Internet Computing (CIC'2017)* San Jose, California, USA, pages 77-86, Oct 15-17, 2017.

[5] Sriramulu Bojjagani, V.N. Sastry, "A Secure End-to-End Proximity NFC-based Mobile Payment Protocol", *Computer Standards & Interfaces (CSI), Elsevier* (Under Review).

[6] Sriramulu Bojjagani, V.N. Sastry, " A novel authentication protocol for prevention of phishing and pharming attacks in mobile banking", *International Journal of*

Trust Management in Computing and Communications (IJTMCC), Inderscience (Under Review).

[7] Sriramulu Bojjagani, V.N. Sastry, "Systematic Survey on Mobile Payments, Protocols, and Security Infrastructure", *Computers & Security, (COSE), Elsevier* (Under Review).

References

[1] RAJNISH TIWARI, STEPHAN BUSE, AND CORNELIUS HERSTATT. **Mobile services in banking sector: the role of innovative business solutions in generating competitive advantage: Technology and Innovation Managment Working Paper No. 48**. 2007.

[2] KATERINA MARKOSKA, IRYNA IVANOCHKO, ET AL. **Mobile Banking Services—Business Information Management with Mobile Payments**. In *Agile Information Business*, pages 125–175. Springer, 2017.

[3] THOMAS R EISENMANN, GEOFFREY PARKER, AND MARSHALL W VAN ALSTYNE. **Opening platforms: how, when and why?** 2008.

[4] JESÚS TÉLLEZ AND SHERALI ZEADALLY. **Architectures and Models for Mobile Payment Systems**. In *Mobile Payment Systems*, pages 35–91. Springer, 2017.

[5] LINGLING GAO AND KEREM AKSEL WAECHTER. **Examining the role of initial trust in user adoption of mobile payment services: an empirical investigation**. *Information Systems Frontiers*, **19**(3):525–548, 2017.

[6] NIINA MALLAT. **Exploring consumer adoption of mobile payments–A qualitative study**. *The Journal of Strategic Information Systems*, **16**(4):413–432, 2007.

[7] YORIS A AU AND ROBERT J KAUFFMAN. **The economics of mobile payments: Understanding stakeholder issues for an emerging financial technology application**. *Electronic Commerce Research and Applications*, **7**(2):141–164, 2008.

[8] MAHIL CARR. **Mobile payment systems and services: an introduction**. In *Mobile Payment Forum*, pages 1–12, 2007.

[9] PAUL GERHARDT SCHIERZ, OLIVER SCHILKE, AND BERND W WIRTZ. **Understanding consumer acceptance of mobile payment services: An empirical analysis**. *Electronic commerce research and applications*, **9**(3):209–216, 2010.

[10] LEI-DA CHEN. **A model of consumer acceptance of mobile payment**. *International Journal of Mobile Communications*, **6**(1):32–52, 2008.

[11] JIE GUO AND HARRY BOUWMAN. **An analytical framework for an m-payment ecosystem: A merchants perspective**. *Telecommunications Policy*, **40**(2):147–167, 2016.

[12] AAKANKSHA GAUR AND JAN ONDRUS. **The role of banks in the mobile payment ecosystem: a strategic asset perspective**. In *ACM Proceedings of the 14th annual international conference on electronic commerce, Singapore, August 6-8, 2012*, pages 171–177.

[13] MARK DE REUVER, EDGAR VERSCHUUR, FATEMEH NIKAYIN, NARCISO CERPA, AND HARRY BOUWMAN. **Collective action for mobile payment platforms: A case study on collaboration issues between banks and telecom operators**. *Electronic Commerce Research and Applications*, **14**(5):331–344, 2015.

[14] JIE GUO, SHAHROKH NIKOU, AND WAGA BOUWMAN. **Analyzing the business model for mobile payment from banks' perspective: An empirical study**. In *24th European Regional Conference of the International Telecommunication Society, Florence, Italy, October 20 - 23, 2013*.

[15] SHAHRYAR SOROOSHIAN. **Business Ethics for Mobile Network Operators**. *Science and Engineering Ethics*, pages 1–2, 2017.

[16] TORSTEN J GERPOTT AND PHIL MEINERT. **Correlates of using the billing system of a mobile network operator to pay for digital goods and services**. *Information Systems Frontiers*, **18**(6):1265–1283, 2016.

[17] JAN DAMSGAARD AND PING GAO. **A framework for analyzing mobile telecommunications market development**. *Mobile Information Systems*, pages 169–182, 2005.

[18] TAN SOO FUN, LEAU YU BENG, JONATHAN LIKOH, AND ROZAINI ROSLAN. **A lightweight and private mobile payment protocol by using mobile network operator**. In *IEEE International Conference on. Computer and Communication Engineering, (ICCCE'2008), Kuala Lumpur, Malaysia, May 13 - 15, 2008*, pages 162–166.

[19] LAETITIA CHAIX AND DOMINIQUE TORRE. **Four models for mobile payments**. *University Nice Sophia-Antipolis, JEL Classification E*, **42**:O33, 2011.

[20] STAMATIS KARNOUSKOS. **Mobile payment: a journey through existing procedures and standardization initiatives**. *IEEE Communications Surveys & Tutorials*, **6**(4), 2004.

[21] SHAIK SHAKEEL AHAMAD, VN SASTRY, AND SIBA K UDGATA. **A secure mobile wallet framework with formal verification**. *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, **4**(2):1–15, 2012.

[22] SHAIK SHAKEEL AHAMAD, VN SASTRY, AND MADHUSOODHNAN NAIR. **A Biometric based Secure Mobile Payment Framework**. In *4th IEEE International Conference on. Computer and Communication Technology (ICCCT' 2013) , Allahabad, India, September 20 - 22, 2013*, pages 239–246.

[23] SHAIK SHAKEEL AHAMAD, IBRAHIM AL-SHOURBAJI, AND SAMAHER AL-JANABI. **A secure NFC mobile payment protocol based on biometrics with formal verification**. *International Journal of Internet Technology and Secured Transactions*, **6**(2):103–132, 2016.

[24] S BOJJAGANI AND VN SASTRY. **A secure end-to-end SMS-based mobile banking protocol**. *International Journal of Communication Systems*, **30**(15):1–19, 2017.

[25] SHAIK SHAKEEL AHAMAD, VN SASTRY, AND SIBA K UDGATA. **A Secure and Optimized Proximity Mobile Payment Framework with Formal Verification**. *Mobile Commerce: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, page 161, 2017.

[26] JUAN CARLOS LÓPEZ PIMENTEL AND RAÚL MONROY. **Formal support to security protocol development: A survey**. *Computación y Sistemas*, **12**(1):89–108, 2008.

[27] REEMA PATEL, BHAVESH BORISANIYA, AVI PATEL, DHIREN PATEL, MUTTUKRISHNAN RAJARAJAN, AND ANDREA ZISMAN. **Comparative analysis of formal model checking tools for security protocol verification**. In *International Conference on Network Security and Applications*, pages 152–163. Springer, 2010.

[28] CHIARA BRAGHIN, NATASHA SHARYGINA, AND KATERINA BARONE-ADESI. **A model checking-based approach for security policy verification of mobile systems**. *Formal Aspects of Computing*, **23**(5):627–648, 2011.

[29] HUILING SHI, WENKE MA, MEIHONG YANG, AND XINCHANG ZHANG. **A Case Study of Model Checking Retail Banking System with SPIN.** *JCP*, **7**(10):2503–2510, 2012.

[30] LLANOS TOBARRA, DIEGO CAZORLA, FERNANDO CUARTERO, GREGORIO DÍAZ, AND EMILIA CAMBRONERO. **Model checking wireless sensor network security protocols: TinySec+ LEAP+ TinyPK**. *Telecommunication Systems*, **40**(3-4):91–99, 2009.

[31] AVISPA. **Automated Validation of Internet Security Protocols**. `http://www.avispa-project.org/web-interface/basic.php`, Jan 2014.

[32] CASIMIER JOSEPH FRANCISCUS CREMERS. *Scyther: Semantics and verification of security protocols*. Eindhoven University of Technology Eindhoven, Netherlands, 2006.

[33] LUCA VIGANÒ. **Automated security protocol analysis with the AVISPA tool**. *Electronic Notes in Theoretical Computer Science*, **155**:61–86, 2006.

[34] ALESSANDRO ARMANDO, DAVID BASIN, YOHAN BOICHUT, YANNICK CHEVALIER, LUCA COMPAGNA, JORGE CUÉLLAR, P HANKES DRIELSMA, PIERRE-CYRILLE HÉAM, OLGA KOUCHNARENKO, JACOPO MANTOVANI,

ET AL. **The AVISPA tool for the automated validation of internet security protocols and applications**. In *International conference on computer aided verification*, pages 281–285. Springer, 2005.

[35] AVISPA TEAM. **AVISPA v1. 0 User Manual**. 2006.

[36] HLPSL TUTORIAL. **A Beginner's Guide to Modelling and Analysing Internet Security Protocols (2005)**. *Available at [AH-03]*.

[37] DAVID BASIN, SEBASTIAN MÖDERSHEIM, AND LUCA VIGANO. **OFMC: A symbolic model checker for security protocols**. *International Journal of Information Security*, **4**(3):181–208, 2005.

[38] CAS CREMERS. **The Scyther tool: Automatic verification of security protocols**, 2009.

[39] MICHAEL BURROWS AND MARTIN ABADI. **A logic of authentication**. In *Proc. R. Soc. Lond. A*, **426**, pages 233–271. The Royal Society, 1989.

[40] PAUL F SYVERSON AND PAUL C VAN OORSCHOT. **On unifying some cryptographic protocol logics**. In *IEEE Computer Society Symposium on. Research in Security and Privacy, 1994*, pages 14–28.

[41] BARBARA KITCHENHAM. **Procedures for performing systematic reviews**. *Keele, UK, Keele University*, **33**(2004):1–26, 2004.

[42] PEARL BRERETON, BARBARA A KITCHENHAM, DAVID BUDGEN, MARK TURNER, AND MOHAMED KHALIL. **Lessons from applying the systematic literature review process within the software engineering domain**. *Journal of systems and software*, **80**(4):571–583, 2007.

[43] ALIREZA SADEGHI, HAMID BAGHERI, JOSHUA GARCIA, AND SAM MALEK. **A taxonomy and qualitative comparison of program analysis techniques for security assessment of android software**. *IEEE Transactions on Software Engineering*, **43**(6):492–530, 2017.

[44] CHANDRASHEKAR JATOTH, GR GANGADHARAN, AND RAJKUMAR BUYYA. **Computational intelligence based QoS-aware web service composition: A systematic literature review**. *IEEE Transactions on Services Computing*, **10**(3):475–492, 2017.

[45] CAROLINA MARTINS, TIAGO OLIVEIRA, AND ALEŠ POPOVIČ. **Understanding the Internet banking adoption: A unified theory of acceptance and use of technology and perceived risk application**. *International Journal of Information Management*, **34**(1):1–13, 2014.

[46] TOMMI LAUKKANEN. **Internet vs mobile banking: comparing customer value perceptions**. *Business process management journal*, **13**(6):788–797, 2007.

[47] HSIU-FEN LIN. **An empirical investigation of mobile banking adoption: The effect of innovation attributes and knowledge-based trust**. *International journal of information management*, **31**(3):252–260, 2011.

[48] AININ SULAIMAN, NOOR ISMAWATI JAAFAR, AND SUHANA MOHEZAR. **An overview of mobile banking adoption among the urban community**. *International Journal of Mobile Communications*, **5**(2):157–168, 2006.

[49] JA-CHUL GU, SANG-CHUL LEE, AND YUNG-HO SUH. **Determinants of behavioral intention to mobile banking**. *Expert Systems with Applications*, **36**(9):11605–11616, 2009.

[50] THOMAS LERNER. *Mobile payment*. Springer, 2013.

[51] PIN LUARN AND HSIN-HUI LIN. **Toward an understanding of the behavioral intention to use mobile banking**. *Computers in human behavior*, **21**(6):873–891, 2005.

[52] AIJAZ A SHAIKH AND HEIKKI KARJALUOTO. **Mobile banking adoption: A literature review**. *Telematics and Informatics*, **32**(1):129–142, 2015.

[53] KWABENA FRIMPONG, OBAID AL-SHURIDAH, ALAN WILSON, AND FREDERICK ASAFO-ADJEI SARPONG. **Effect of inherent innovativeness and consumer readiness on attitudes to mobile banking**. *Journal of Financial Services Marketing*, pages 1–15, 2017.

[54] TAO ZHOU. **An empirical examination of continuance intention of mobile payment services**. *Decision Support Systems*, **54**(2):1085–1091, 2013.

[55] MARK SHERMAN. **An introduction to mobile payments: market drivers, applications, and inhibitors**. In *Proceedings of the: 1st ACM International Conference on Mobile Software Engineering and Systems, (MOBILESoft'14), Hyderabad, India, June 2-3, 2014*, pages 71–74.

[56] VEDASHREE MALI. **Demonetization: A step towards modified India**. *International Journal of Commerce and Management Research*, **2**(12):35–36, 2016.

[57] WEIDONG GUO. **Design of architecture for mobile payments system**. In *IEEE Conference on. Control and Decision , (CCDC'2008), Yantai, Shandong, China, July 2 - 4, 2008*, pages 1732–1735.

[58] MARKO HASSINEN. **SafeSMS-end-to-end encryption for SMS**. In *Telecommunications, 2005. ConTEL 2005. Proceedings of the 8th International Conference on*, **2**, pages 359–365. IEEE, 2005.

[59] MING KI CHONG. **Security of mobile banking: Secure SMS banking**. *Data Network Architectures Group. University of Cape Town, South Africa*, 2006.

[60] HANY HARB, HASSAN FARAHAT, AND MOHAMED EZZ. **SecureSMSPay: secure SMS mobile payment model**. In *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, pages 11–17. IEEE, 2008.

[61] DAVID LISONEK AND MARTIN DRAHANSKÝ. **SMS encryption for mobile communication**. In *Security Technology, 2008. SECTECH'08. International Conference on*, pages 198–201. IEEE, 2008.

[62] MOHSEN TOORANI AND A BEHESHTI. **SSMS-A secure SMS messaging protocol for the m-payment systems**. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 700–705. IEEE, 2008.

[63] JOHNNY LI-CHANG LO, JUDITH BISHOP, AND JAN HP ELOFF. **SMSSec: an end-to-end protocol for secure SMS**. *Computers & Security*, **27**(5):154–167, 2008.

[64] SITI DIANAH ABDUL BUJANG AND ALI SELAMAT. **Verification of mobile SMS application with model checking agent**. In *IEEE International Conference on. Information and Multimedia Technology, (ICIMT'09),Jeju Island, South Korea, December 16 - 18, 2009*, pages 361–365.

[65] ALFREDO DE SANTIS, ANIELLO CASTIGLIONE, GIUSEPPE CATTANEO, MAURIZIO CEMBALO, FABIO PETAGNA, AND UMBERTO FERRARO PETRILLO. **An extensible framework for efficient secure SMS**. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, pages 843–850. IEEE, 2010.

[66] SUPAKORN KUNGPISDAN AND MAYKIN WARASART. **Somp: An sms-based operator-assisted mobile payment protocol**. In *Proceedings of the 2010 National Computer Science and Engineering Conference. Chiangmai, Thailand: IEEE*, 2010.

[67] MARY AGOYI AND DEVRIM SERAL. **SMS security: an asymmetric encryption approach**. In *Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on*, pages 448–452. IEEE, 2010.

[68] SURAKARN DUANGPHASUK, MAYKIN WARASART, AND SUPAKORN KUNGPISDAN. **Design and accountability analysis of a secure SMS-based mobile payment protocol**. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*, pages 442–445. IEEE, 2011.

[69] NEETESH SAXENA AND NARENDRA S CHAUDHARI. **Secure encryption with digital signature approach for Short Message Service**. In *Information and Communication Technologies (WICT), 2012 World Congress on*, pages 803–806. IEEE, 2012.

[70] ANURAG ASHOK, SAUMEEL GAJERA, JANHV CHAKRABORTY, PRAJAKTA AROLKER, AND NITIKA RAI. **M-Wallet: An SMS based payment system**. 2012.

[71] NOR AZLINA BT ABD RAHMAN, KHALIDA SHAJARATUDDUR BT HARUN, AND YUSNITA BT YUSOF. **SMS banking transaction as an alternative for information, transfer and payment at merchant shops in Malaysia**. In *Information Technology and e-Services (ICITeS), 2013 3rd International Conference on*, pages 1–6. IEEE, 2013.

[72] SRIRAMULU BOJJAGANI AND VN SASTRY. **Stamba: Security testing for Android mobile banking apps**. In *Proceedings of: Second International Symposium on Signal Processing and Intelligent Recognition Systems (SIRS'15), Trivandrum, India, December 16-19, 2015*, pages 671–683.

[73] N RAGHU KISORE AND SUPRIYA SAGI. **A secure SMS protocol for implementing digital cash system**. In *4th International Conference On. Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, August 10-13, 2015*, pages 1883–1892.

[74] SAURABH SAMANTA, RADHESH MOHANDAS, AND ALWYN R PAIS. **Secure Short Message Peer-to-Peer Protocol**. *International Journal of Electronic Commerce Studies*, **3**(1):45, 2012.

[75] MHAIR KASHIF. **Secure SMS communication using encryption gateway and digital signature**. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 1430–1434. IEEE, 2014.

[76] LIEBANA-CABANILLAS FRANCISCO, MUÑOZ-LEIVA FRANCISCO, AND SANCHEZ-FERNANDEZ JUAN. **Payment systems in new electronic environments: consumer behavior in payment systems via SMS**. *International Journal of Information Technology & Decision Making*, **14**(02):421–449, 2015.

[77] ASAF SARAJLIC AND DAMIR OMERASEVIC. **Access channels in m-commerce services**. In *29th IEEE International Conference on. Information Technology Interfaces, (ITI 2007). Cavtat / Dubrovnik, Croatia,June 25-28, 2007*, pages 507–512.

[78] PUNEET GUPTA. **End To End USSD System**. *TATA Tele Service Limited, INDIA, July*, **7**:2010, 2010.

[79] MARYAM ASADI TEHRANI, ALI ASGHAR AMIDIAN, JAFAR MUHAMMADI, AND HAMIDREZA RABIEE. **A survey of system platforms for mobile payment**. In *Fourth International Conference on. Management of e-Commerce and e-Government (ICMeCG'2010), Chengdu, China, October 23-24, 2010*, pages 376–381.

[80] BARAKA W NYAMTIGA, ANAEL SAM, AND LOSERIAN S LAIZER. **Security Perspectives for USSD versus SMS in conducting mobile transactions: A case study of Tanzania**. *international journal of technology enhancements and emerging engineering research*, **1**(3):38–43, 2013.

[81] MICHAEL MASSOTH AND THOMAS BINGEL. **Performance of different mobile payment service concepts compared with a NFC-based solution**. In *Fourth International Conference on. Internet and Web Applications and Services, ICIW'09, Venice/Mestre, Italy, May 24-28, 2009*, pages 205–210.

[82] ESKO STROMMER, JOUNI KAARTINEN, JUHA PARKKA, ARTO YLISAUKKO-OJA, AND ILKKA KORHONEN. **Application of near field communication for health monitoring in daily life**. In *28th IEEE Annual International Conference on. Engineering in Medicine and Biology Society, (EMBS'06), New York, NY, USA, August 30 - Sept 3, 2006*, pages 3246–3249.

[83] JÜRGEN MORAK, ALEXANDER KOLLMANN, GÜNTER SCHREIER, ET AL. **Feasibility and usability of a home monitoring concept based on mobile phones and near field communication (NFC) technology**. In *Medinfo, IOS Press, 2007*, pages 112–116, 2007.

[84] IVÁN SÁNCHEZ, MARTA CORTÉS, AND JUKKA RIEKKI. **Controlling multimedia players using NFC enabled mobile phones**. In *ACM Proceedings of the 6th international conference on. Mobile and ubiquitous multimedia, (MUM'07), Oulu, Finland, December 12 - 14, 2007*, pages 118–124.

[85] JOSÉ BRAVO, RAMÓN HERVÁS, R GALLEGO, GREGORIO CASERO, MARCOS VERGARA, T CARMONA, CARMEN FUENTES, SALVADOR W NAVA, GABRIEL CHAVIRA, AND VLADIMIR VILLARREAL. **Enabling NFC technology to support activities in an Alzheimer's day center**. In *ACM Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments (PETRA'08), Athens, Greece , July 16 - 18, 2008*, page 81.

[86] GABRIEL CHAVIRA, SALVADOR W NAVA, RAMÓN HERVÁS, VLADIMIR VILLARREAL, JOSÉ BRAVO, SERGIO MARTÍN, AND MANUEL CASTRO. **Services through NFC technology in AmI Environment**. In *ACM Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, Linz, Austria, November 24 - 26, 2008*, pages 666–669.

[87] NIKOS KEFALAKIS, NEKTARIOS LEONTIADIS, JOHN SOLDATOS, KIEV GAMA, AND DIDIER DONSEZ. **Supply chain management and NFC picking demonstrations using the AspireRfid middleware platform**. In *ACM Proceedings of the. ACM/IFIP/USENIX Middleware'08 (Companion,08), Leuven, Belgium, December 01 - 05, 2008*, pages 66–69.

[88] IRENE CAPPIELLO, STEFANO PUGLIA, AND ANDREA VITALETTI. **Design and initial evaluation of a ubiquitous touch-based remote grocery shopping process**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009*, pages 9–14.

[89] JURGEN MORAK, VERENA SCHWETZ, DIETER HAYN, FRIEDRICH FRUHWALD, AND GUNTER SCHREIER. **Electronic data capture platform for clinical research based on mobile phones and near field communication technology**. In *30th Annual IEEE International Conference on. Engineering in Medicine and Biology Society, 2008. (EMBS'2008), Vancouver, BC, Canada, August 20 - 25, 2008*, pages 5334–5337.

[90] ANTOINE FRESSANCOURT, COLOMBE HÉRAULT, AND ERIC PTAK. **NFCSocial: Social networking in mobility through IMS and NFC**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24 Feb, 2009*, pages 24–29.

[91] ROBERT HARDY, ENRICO RUKZIO, MATTHIAS WAGNER, AND MASSIMO PAOLUCCI. **Exploring expressive nfc-based mobile phone interaction with large dynamic displays**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24-24, 2009*, pages 36–41.

[92] STEPHAN KARPISCHEK, FLORIAN MICHAHELLES, FLORIAN RESATSCH, AND ELGAR FLEISCH. **Mobile sales assistant-an nfc-based product information system for retailers**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24-24, 2009*, pages 20–23.

[93] STEPHAN KARPISCHEK, FLORIAN MICHAHELLES, ALBRECHT BEREUTER, AND ELGAR FLEISCH. **A maintenance system based on near field communication**. In *Third IEEE International Conference on. Next Generation Mobile Applications, Services and Technologies, (NGMAST'09), Cardiff, Wales, UK, September 15 - 18, 2009*, pages 234–238. IEEE, 2009.

[94] FABIAN KNEISSL, RICHARD ROTTGER, UWE SANDNER, JAN MARCO LEIMEISTER, AND HELMUT KRCMAR. **All-i-touch as combination of nfc and lifestyle**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009*, pages 51–55.

[95] ADAM MARCUS, GUIDO DAVIDZON, DENISE LAW, NAMRATA VERMA, RICH FLETCHER, AAMIR KHAN, AND LUIS SARMENTA. **Using NFC-enabled mobile phones for public health in developing countries**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009*, pages 30–35.

[96] J MORAK, D HAYN, P KASTNER, M DROBICS, AND G SCHREIER. **Near field communication technology as the key for data acquisition in clinical research**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009*, pages 15–19.

[97] ERKKI SIIRA, TUOMO TUIKKA, AND VILI TORMANEN. **Location-based mobile Wiki using NFC tag infrastructure**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09)*.

[98] PILAR CASTRO GARRIDO, GUILLERMO MATAS MIRAZ, IRENE LUQUE RUIZ, AND MIGUEL ÁNGEL GÓMEZ-NIETO. **A model for the development of NFC context-awareness applications on Internet of Things**. In *Second IEEE International Workshop on. Near Field Communication (NFC), Monaco, Monaco, April 20-20, 2010*, pages 9–14.

[99] FELIX KÖBLER, PHILIP KOENE, HELMUT KRCMAR, MATTHIAS ALTMANN, AND JAN MARCO LEIMEISTER. **LocaTag-An NFC-based system enhancing instant messaging tools with real-time user location**. In *Second IEEE International Workshop on. Near Field Communication (NFC), Monaco, Monaco, April 20 - 20, 2010*, pages 57–61.

[100] M VERGARA, P DÍAZ-HELLÍN, J FONTECHA, R HERVÁS, C SÁNCHEZ-BARBA, C FUENTES, AND JOSÉ BRAVO. **Mobile prescription: An NFC-based proposal for AAL**. In *Second IEEE International Workshop on. Near Field Communication (NFC), Monaco, Monaco, April 20 - 20, 2010*, pages 27–32.

[101] HSU-CHEN CHENG, WEN-WEI LIAO, TIAN-YOW CHI, AND SIAO-YUN WEI. **A secure and practical key management mechanism for NFC read-write mode**. In *13th IEEE International Conference on. Advanced Communication Technology (ICACT), Gangwon-Do, Korea (South), Feb 13 - 16, 2011*, pages 1095–1011.

[102] PARDIS POURGHOMI, GHEORGHITA GHINEA, ET AL. **A proposed NFC payment application**. *International Journal of Advanced Computer Science and Applications (IJACSA), 2013arXiv preprint arXiv:1312.2828*, **4**.

[103] SANDRA DOMINIKUS AND MANFRED AIGNER. **mCoupons: An application for near field communication (NFC)**. In *21st IEEE International Conference on. Advanced Information Networking and Applications Workshops, (AINAW'07), Niagara Falls, Ont., Canada, May 21 - 23, 2007*, **2**, pages 421–428.

[104] STEFANO LEVIALDI GHÌRON, SERENA SPOSATO, CARLO MARIA MEDAGLIA, AND ALICE MORONI. **NFC ticketing: A prototype and usability test of an NFC-based virtual ticketing application**. In *First IEEE*

International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009, pages 45–50.

[105] MARC PASQUET, JOAN REYNAUD, AND CHRISTOPHE ROSENBERGER. **Secure payment with NFC mobile phone in the SmartTouch project**. In *IEEE International Symposium on. Collaborative Technologies and Systems, (CTS'2008), Irvine, CA, USA, May 19 - 23, 2008*, pages 121–126.

[106] H AZIZA. **NFC technology in mobile phone next-generation services**. In *Second IEEE International Workshop on. Near Field Communication (NFC'2010), Monaco, Monaco, April 20 - 20, 2010*, pages 21–26.

[107] GUILLERMO MATAS MIRAZ, IRENE LUQUE RUIZ, AND MIGUEL ÁNGEL GÓMEZ-NIETO. **How NFC can be used for the compliance of European higher education area guidelines in European universities**. In *First IEEE International Workshop on. Near Field Communication, (NFC'09), Hagenberg, Austria, Feb 24 - 24, 2009*, pages 3–8.

[108] YUE-SHAN CHANG, CHING-LUNG CHANG, YUNG-SHUAN HUNG, AND CHING-TSORNG TSAI. **NCASH: NFC phone-enabled personalized context awareness smart-home environment**. *Cybernetics and Systems: An International Journal*, **41**(2):123–145, 2010.

[109] JEF NEEFS, FREDERIK SCHROOYEN, JEROEN DOGGEN, AND KAREL RENCKENS. **Paper ticketing vs. Electronic Ticketing based on off-line system'Tapango'**. In *Second IEEE International Workshop on. Near Field Communication (NFC'2010), Monaco, Monaco, April 20 - 20, 2010*, pages 3–8.

[110] RAINER STEFFEN, JÖRG PREISSINGER, TOBIAS SCHÖLLERMANN, ARMIN MÜLLER, AND INGO SCHNABEL. **Near field communication (NFC) in an automotive environment**. In *Second IEEE International Workshop on. Near Field Communication (NFC), Monaco, Monaco, April 20 - 20, 2010*, pages 15–20.

[111] RAJESH KRISHNA BALAN, NARAYAN RAMASUBBU, KOMSIT PRAKOBPHOL, NICOLAS CHRISTIN, AND JASON HONG. **mFerio: the**

design and evaluation of a peer-to-peer mobile payment system. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 291–304. ACM, 2009.

[112] ERKKI SIIRA AND VILI TÖRMÄNEN. **The impact of NFC on multimodal social media application**. In *Second IEEE International Workshop on. Near Field Communication (NFC), Monaco, Monaco, April 20 - 20, 2010*, pages 51–56.

[113] LISHOY FRANCIS, GERHARD HANCKE, KEITH MAYES, AND KONSTANTINOS MARKANTONAKIS. **Practical NFC peer-to-peer relay attack using mobile phones**. In *Springer Proceedings of: International Workshop on. Radio Frequency Identification: Security and Privacy Issues (RFIDSec' 2010), Istanbul, Turkey, June 8 - 9, 2010*, pages 35–49.

[114] LUCA MAINETTI, LUIGI PATRONO, AND ROBERTO VERGALLO. **IDA-Pay: A secure and efficient micro-payment system based on Peer-to-Peer NFC technology for Android mobile devices**. **8**(4):117–125, 2012.

[115] DAVID M MONTEIRO, JOEL JPC RODRIGUES, AND JAIME LLORET. **A secure NFC application for credit transfer among mobile phones**. In *IEEE International Conference on. Computer, Information and Telecommunication Systems (CITS'2012), Amman, Jordan, May 14 - 16, 2012*, pages 1–5.

[116] RAJALAKSHMI NANDAKUMAR, KRISHNA KANT CHINTALAPUDI, VENKAT PADMANABHAN, AND RAMARATHNAM VENKATESAN. **Dhwani: secure peer-to-peer acoustic NFC**. In *ACM SIGCOMM Computer Communication Review*, **43**, pages 63–74. ACM, 2013.

[117] JIAJUN JIM CHEN AND CARL ADAMS. **Short-range wireless technologies with mobile payments systems**. In *Proceedings of the 6th international conference on Electronic commerce*, pages 649–656. ACM, 2004.

[118] SOJEN PRADHAN, ELAINE LAWRENCE, AND AGNIESZKA ZMIJEWSKA. **Bluetooth as an enabling technology in mobile transactions**. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, **2**, pages 53–58. IEEE, 2005.

[119] GIANLUIGI ME AND ALEX SCHUSTER. **A mobile local payment system Bluetooth based**. In *International Symposium on Wireless Communications (ISWSN'05)*, 2005.

[120] TZU-CHANG YEH, JIAN-REN PENG, SHENG-SHIH WANG, AND JUN-PING HSU. **Securing Bluetooth Communications.** *IJ Network Security*, **14**(4):229–235, 2012.

[121] RUSHABH PATEL, AKHIL KUNCHE, NIHAR MISHRA, ZAKWAN BHAIYAT, AND RAHUL JOSHI. **Paytooth-A Cashless Mobile Payment System based on Bluetooth**. *International Journal of Computer Applications*, **120**(24), 2015.

[122] ALEXANDER YOHAN, NAI-WEI LO, VINCENTIUS RANDY, SHIH-JEN CHEN, AND MING-YUAN HSU. **A Novel Authentication Protocol for Micropayment with Wearable Devices**. In *ACM Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, Danang, Viet Nam, January 04 - 06, 2016*, page 18.

[123] TIANTIAN HAN AND LEI DING. **Design and implementation of Bluetooth beacon in mobile payment system**. In *AIP Conference Proceedings*, **1864**, page 020019. AIP Publishing, 2017.

[124] ALEXANDER YOHAN, NAI-WEI LO, AND DONI WINATA. **An Indoor Positioning-Based Mobile Payment System Using Bluetooth Low Energy Technology**. *Sensors*, **18**(4):974, 2018.

[125] DOMINIQUE PARET AND RODERICK RIESCO. *RFID and contactless smart card applications*. Wiley Online Library, 2005.

[126] WEIDONG GUO. **Research on mobile commerce based on RFID**. In *IEEE International Conference on. Service Operations and Logistics, and Informatics,(IEEE/SOLI'2008), Beijing, China, October 12 -15, 2008*, **1**, pages 853–857.

[127] YULI FU AND QI FU. **Scheme and secure protocol of mobile payment based on RFID**. In *IEEE 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, (ASID'2009). ,City University of Hong Kong,Hong Kong, China, August 20-22, 2009*, pages 631–634.

[128] MOHAMMED A QADEER, NADEEM AKHTAR, SHALINI GOVIL, AND ANUJA VARSHNEY. **A novel scheme for mobile payment using RFID-enabled smart SIMcard**. In *IEEE International Conference on. Future Computer and Communication, (ICFCC'2009), Kuala Lumpar, Malaysia, April 3-5, 2009*, pages 339–343.

[129] JUNWEI ZOU, CHU ZHANG, CHONGBO DONG, CHUNXIAO FAN, AND ZHIGANG WEN. **Mobile payment based on RFID-SIM card**. In *IEEE 10th International Conference onComputer and Information Technology (CIT), University of Bradford, Bradford, United Kingdom,June 29- July 01, 2010*, pages 2052–2054.

[130] IZABELA LACMANOVIĆ, BILJANA RADULOVIĆ, AND DEJAN LACMANOVIĆ. **Contactless payment systems based on RFID technology**. In *Proceedings of the: IEEE 33rd International Convention, (MIPRO'2010) ,Opatija, Croatia, May 24-28, 2010*, pages 1114–1119.

[131] WEI DAI, SHUO ZHOU, GUANGJUN LUO, ZONGXING CHEN, AND LING XIE. **Analyze on Mobile Payment Based on RFID**. *Procedia Environmental Sciences*, **10**:950–955, 2011.

[132] DONG YANG AND QINGXIAN WANG. **The study on the application of RFID-based mobile payment to the Internet of Things**. In *International Conference on. Multimedia Technology (ICMT'2011),Hangzhou, China, July 26 - 28, 2011*, pages 908–911.

[133] SWATI R ZOPE AND MARUTI LIMKAR. **RFID based Bill Generation and Payment through Mobile**. *International Journal of Computer Science and Network (IJCSN)*, **1**(3), 2012.

[134] SL TING AND ALBERT HC TSANG. **Design of an RFID-based inventory control and management system: a case study**. *The West Indian journal of engineering*, 2012.

[135] XUAN LIU, HAITAO ZHANG, JINGXIAN FANG, GUAN GUAN, AND YUNDI HUANG. **Intelligent shopping cart with quick payment based on dynamic target tracking**. In *IEEE 4th International Conference on. Cloud Computing and*

Intelligence Systems (CCIS'2016), Beijing, China, August 17 - 19, 2016, pages 88–93.

[136] GREG GOGOLIN AND ERIN GOGOLIN. **The Use of Embedded Mobile, RFID, Location Based Services, and Augmented Reality in Mobile Applications**. *International Journal of Handheld Computing Research (IJHCR)*, **8**(1):42–52, 2017.

[137] PI HUANG AND ANTHONY C BOUCOUVALAS. **Future personal" e-payment": IRFM**. *IEEE Wireless Communications*, **13**(1):60–66, 2006.

[138] K OK, V COSKUN, AND B OZDENIZCI. **Near Field Communication: From Theory to Practice**, 2012.

[139] JAESIK LEE, CHANG-HYUN CHO, AND MOON-SEOG JUN. **Secure quick response-payment (QR-Pay) system using mobile device**. In *13th IEEE International Conference on. Advanced Communication Technology (ICACT), Gangwon-Do, Korea (South), Gangwon-Do, Korea (South), Feb 13 - 16, 2011*, pages 1424–1427.

[140] DAVID CONDE-LAGOA, ENRIQUE COSTA-MONTENEGRO, FRANCISCO J GONZÁLEZ-CASTAÑO, AND FELIPE GIL-CASTIÑEIRA. **Secure eTickets based on QR-Codes with user-encrypted content**. In *IEEE International Conference on. Consumer Electronics (ICCE), 2010 Digest of Technical Papers, January 11-13, 2010*, pages 257–258.

[141] HATMA SURYOTRISONGKO, BAMBANG SETIAWAN, ET AL. **A Novel Mobile Payment Scheme based on Secure Quick Response Payment with Minimal Infrastructure for Cooperative Enterprise in Developing Countries**. *Procedia-Social and Behavioral Sciences*, **65**:906–912, 2012.

[142] JAE-SIK LEE, HAN-NA YOU, CHANG-HYUN CHO, AND MOON-SEOG JUN. **A design secure QR-Login user authentication protocol and assurance methods for the safety of critical data using smart device**. *The Journal of Korean Institute of Communications and Information Sciences*, **37**(10):949–964, 2012.

[143] SANA NSEIR, NAEL HIRZALLAH, AND MUSBAH AQEL. **A secure mobile payment system using QR code**. In *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*, pages 111–114. IEEE, 2013.

[144] FRANCISCO LIÉBANA-CABANILLAS, IVIANE RAMOS DE LUNA, AND FRANCISCO J MONTORO-RÍOS. **User behaviour in QR mobile payment system: the QR Payment Acceptance Model**. *Technology Analysis & Strategic Management*, **27**(9):1031–1049, 2015.

[145] PUCHONG SUBPRATATSAVEE AND PRAMOTE KUACHAROEN. **Transaction authentication using HMAC-based one-time password and QR code**. In *Computer Science and its Applications*, pages 93–98. Springer, 2015.

[146] PUCHONG SUBPRATATSAVEE AND PRAMOTE KUACHAROEN. **Internet banking transaction authentication using mobile one-time password and QR code**. *Advanced Science Letters*, **21**(10):3189–3193, 2015.

[147] ZLATKO ČOVIĆ, ÜRMÖS VIKTOR, JÁNOS SIMON, DALIBOR DOBRILOVIĆ, AND ŽELJKO STOJANOV. **Usage of QR codes in web based system for the electronic market research**. In *Intelligent Systems and Informatics (SISY), 2016 IEEE 14th International Symposium on*, pages 187–192. IEEE, 2016.

[148] YANG-WAI CHOW, WILLY SUSILO, GUOMIN YANG, MAN HO AU, AND CONG WANG. **Authentication and transaction verification using QR codes with a mobile device**. In *Proceedings in: 9th International Conference on. Security, Privacy, and Anonymity in Computation, Communication, and Storage, SpaCCS-2016, Zhangjiajie, China, November 16-18, 2016*, pages 437–451. Springer, 2016.

[149] XIAOLING ZHU, ZHENGFENG HOU, DONGHUI HU, AND JING ZHANG. **Secure and Efficient Mobile Payment Using QR Code in an Environment with Dishonest Authority**. In *9th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS '2016), China, November 16-18, 2016*, pages 452–465.

[150] JIANFENG LU, ZAORANG YANG, LINA LI, WENQIANG YUAN, LI LI, AND CHIN-CHEN CHANG. **Multiple schemes for mobile payment authentication using QR code and visual cryptography**. *Mobile Information Systems*, **2017**, 2017.

[151] STIG F MJOLSNES AND CHUNMING RONG. **Localized credentials for server assisted mobile wallet**. In *IEEE International Conference on. Computer Networks and Mobile Computing, Beijing, China, Oct. 16-19 ,2001*, pages 203–208.

[152] HAO ZHAO AND SEAD MUFTIC. **The concept of secure mobile wallet**. In *Internet Security (WorldCIS), 2011 World Congress on*, pages 54–58. IEEE, 2011.

[153] JAEMIN PARK, KYOUNGTAE KIM, AND MINJEONG KIM. **The Aegis: UICC-Based Security Framework**. In *Second IEEE International Conference on. Future Generation Communication and Networking, 2008. (FGCN'08), Sanya, Hainan Island, China, December 13 - 15, 2008*, **1**, pages 264–269.

[154] VINCENT ALIMI AND MARC PASQUET. **Post-distribution provisioning and personalization of a payment application on a UICC-based Secure Element**. In *IEEE International Conference on. Availability, Reliability and Security, 2009. (ARES'09), Fukuoka, Japan, March 16 - 19, 2009*, pages 701–705.

[155] HAO ZHAO AND SEAD MUFTIC. **Design and implementation of a mobile transactions client system: Secure UICC mobile wallet**. *International Journal for information security research*, **1**(3):113–120, 2011.

[156] IOANNIS KOUNELIS, HAO ZHAO, AND SEAD MUFTIC. **Secure Middleware for Mobile Phones and UICC Applications**. In *International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, London, UK, June 22 - 24, 2011*, pages 143–152.

[157] JAEMIN PARK, KIYOUNG BAEK, AND CHEOLOH KANG. **Secure Profile Provisioning Architecture for Embedded UICC**. In *IEEE Eighth International Conference on. Availability, Reliability and Security (ARES'13), Universität Regensburg, Germany, September 2-6, 2013*, pages 297–303.

[158] SHAIK SHAKEEL AHAMAD, VN SASTRY, AND SIBA K UDGATA. **Secure mobile payment framework based on UICC with formal verification**. *International Journal of Computational Science and Engineering*, **9**(4):355–370, 2014.

[159] SUPAKORN KUNGPISDAN. *Modelling, design, and analysis of secure mobile payment systems*. PhD thesis, Monash University, 2005.

[160] XIAOLIN ZHENG AND DEREN CHEN. **Study of mobile payments system**. In *E-Commerce, 2003. CEC 2003. IEEE International Conference On*, pages 24–27. IEEE, 2003.

[161] YEN CHOON CHING AND HEINZ KREFT. **FairCASH: Concepts and Framework**. In *IEEE Second International Conference on. Next Generation Mobile Applications, Services and Technologies, (NGMAST'08), Washington, DC, USA, September 16-19, 2008*, pages 269–274.

[162] OMAIMA BAMASAK. **Exploring consumers acceptance of mobile payments–an empirical study**. *International Journal of Information Technology, Communications and Convergence*, **1**(2):173–185, 2011.

[163] EMMA L SLADE, MICHAEL D WILLIAMS, AND YOGESH K DWIVEDI. **Mobile payment adoption: Classification and review of the extant literature**. *The Marketing Review*, **13**(2):167–190, 2013.

[164] JESUS TELLEZ ISAAC AND ZEADALLY SHERALI. **Secure mobile payment systems**. *IT Professional*, **16**(3):36–43, 2014.

[165] ABDULKAREEM ALJOHANI. **Mobile Payments with Instant Settlements**. In *IEEE Eighth International Conference on. Next Generation Mobile Applications, Services and Technologies (NGMAST'2014), Oxford, United Kingdom, September 10-12, 2014*, pages 181–185.

[166] JESÚS TÉLLEZ AND SHERALI ZEADALLY. **Mobile Payment Systems**. 2017.

[167] SANGJIN KIM AND HEEKUCK OH. **An atomic micropayment system for a mobile computing environment**. *IEICE Transactions on information and Systems*, **84**(6):709–716, 2001.

[168] RÓBERT PÁRHONYI, LAMBERT JM NIEUWENHUIS, AND AIKO PRAS. **Second generation micropayment systems: lessons learned**. In *Challenges of Expanding Internet: E-Commerce, E-Business, and E-Government*, pages 345–359. Springer, 2005.

[169] XIAOLING DAI, OLUWATOMI AYOADE, AND JOHN GRUNDY. **Off-line micropayment protocol for multiple vendors in mobile commerce**. In *Seventh IEEE International Conference on. Parallel and Distributed Computing, Applications and Technologies, (PDCAT'06), Taipei, Taiwan, December 4 - 7, 2006*, pages 197–202.

[170] NAN JIANG, XIANG-DONG LIU, JING-YING ZHAO, AND DE-LI YANG. **A mobile micropayment protocol based on chaos**. In *Eighth IEEE International Conference onMobile Business, (ICMB'2009), Dalian, China, June 27 - 28, 2009*, pages 284–289.

[171] NAN JIANG, YANG RI-JING, LIN ZHENG-KUI, ET AL. **Mobile micropayment protocol based on chaotic dual Hash chains**. *Journal of Jilin University (Engineering and Technology Eition)*, **2**:055, 2010.

[172] ANDREU PERE ISERN-DEYÀ, M MAGDALENA PAYERAS-CAPELLÀ, MACIÀ MUT-PUIGSERVER, JOSEP-LLUÍS FERRER-GOMILA, AND LLORENÇ HUGUET-ROTGER. **Micropayment scheme implementation on the Android platform with performance evaluation**. In *Proceedings of the 10th ACM International Conference on. Advances in Mobile Computing & Multimedia, Bali, Indonesia, December 03 - 05, 2012*, pages 64–73.

[173] ANDREU PERE ISERN-DEYÀ, M MAGDALENA PAYERAS-CAPELLÀ, MACIÀ MUT-PUIGSERVER, AND JOSEP L FERRER-GOMILA. **Anonymous and fair micropayment scheme with protection against coupon theft**. *International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS)*, **4**(2):54–71, 2013.

[174] ANDREU PERE ISERN-DEYÀ, M MAGDALENA PAYERAS-CAPELLÀ, MACIÀ MUT-PUIGSERVER, JOSEP-LLUIS FERRER-GOMILA, AND LLORENÇ HUGUET-ROTGER. **Micropayment proposal with formal verification using coloured**

Petri nets and performance analysis on the Android platform. *International Journal of Business Intelligence and Data Mining*, **8**(1):74–104, 2013.

[175] VANESA DAZA, ROBERTO DI PIETRO, FLAVIO LOMBARDI, AND MATTEO SIGNORINI. **Force: fully off-line secure credits for mobile micro payments**. In *Security and Cryptography (SECRYPT), 2014 11th International Conference on*, pages 1–12. IEEE, 2014.

[176] XIAODI HUANG, JINSONG BAO, XIAOLING DAI, EDWIN SINGH, WEIDONG HUANG, AND CHANGQIN HUANG. **M&E-NetPay: A Micropayment System for Mobile and Electronic Commerce**. *Symmetry*, **8**(8):74, 2016.

[177] VANESA DAZA, ROBERTO DI PIETRO, FLAVIO LOMBARDI, AND MATTEO SIGNORINI. **FRoDO: Fraud Resilient Device for Off-Line Micro-Payments**. *IEEE Transactions on Dependable and Secure Computing*, **13**(2):296–311, 2016.

[178] YINING LIU, QUANYU ZHAO, GAO LIU, LIANG CHANG, AND JIAN SHEN. **A Fairness-Enhanced Micropayment Scheme**. *Wireless Personal Communications*, **96**(4):5267–5278, 2017.

[179] SIMON FONG AND EDISON LAI. **Mobile mini-payment scheme using SMS-credit**. *Computational Science and Its Applications–ICCSA 2005*, pages 39–78, 2005.

[180] PHILIP GARNER, REUBEN EDWARDS, AND PAUL COULTON. **Card-based macropayment for mobile phones**. In *IEEE International Conference on. Mobile Business, (ICMB'06), Copenhagen, Denmark, June 26-27, 2006.*, pages 4–4.

[181] S KUMAR AND S RABARA. **An architectural design for secure mobile remote macro-payments**. *Journal of Next Generation Information Technology*, **1**(2), 2010.

[182] TAN SOO FUN, LEAU YU BENG, AND MOHD NORHISHAM RAZALI. **Review of mobile macro-payment schemes**. *Journal of Advances in Computer Networks*, **1**(4), 2013.

[183] R SAI ANAND AND CE VENI MADHAVAN. **An online, transferable e-cash payment system**. In *INDOCRYPT*, pages 93–103. Springer, 2000.

[184] JIAN-SEN WANG, FUW-YI YANG, AND INCHEON PAIK. **A novel e-cash payment protocol using trapdoor hash function on smart mobile devices**. *International Journal of Computer Science and Network Security*, **11**(6):12–19, 2011.

[185] FUW-YI YANG, ZHEN-WEI LIU, AND SU-HUI CHIU. **Mobile Banking Payment System.** *JoWUA*, **2**(3):85–95, 2011.

[186] ARPITA MAZUMDAR AND DEBASIS GIRI. **On-line electronic payment system using signcryption**. *Procedia Technology*, **6**:930–938, 2012.

[187] FUW-YI YANG, SU-HUI CHIU, AND HUNG-YUMR CHEN. **A New E-cash Scheme Based on a Trapdoor Hash Function**. *Advances in Information Sciences and Service Sciences*, pages 229–237, 2012.

[188] XIAOJUN WEN, YONGZHI CHEN, AND JUNBIN FANG. **An inter-bank E-payment protocol based on quantum proxy blind signature**. *Quantum information processing*, **12**(1):549–558, 2013.

[189] SARVESH TANWAR ET AL. **Implementation of Fair-exchange and Anonymous Online E-cash Protocol for E-commerce**. *International Journal of Information Technology and Computer Science (IJITCS)*, **8**(8):66, 2016.

[190] SK HAFIZUL ISLAM, RUHUL AMIN, GP BISWAS, MOHAMMAD S OBAIDAT, AND MUHAMMAD KHURRAM KHAN. **Provably secure pairing-free identity-based partially blind signature scheme and its application in online e-cash system**. *Arabian Journal for Science and Engineering*, **41**(8):3163–3176, 2016.

[191] WOEI-JIUNN TSAUR, LO-YAO YEH, AND TONG-YING JUANG. **Efficient and Secure ECC-based Electronic Cash Payment Applications for Internet Consumers**. In *The 18th Int'l Conf on Internet Computing and Internet of Things, (ICOMP'17), Las Vegas, Nevada, USA,July 17-20, 2017*.

[192] WEN-SHENQ JUANG. **RO-cash: An efficient and practical recoverable prepaid offline e-cash scheme using bilinear pairings**. *Journal of Systems and Software*, **83**(4):638–645, 2010.

[193] ZIBA ESLAMI AND MEHDI TALEBI. **A new untraceable off-line electronic cash system**. *Electronic Commerce Research and Applications*, **10**(1):59–66, 2011.

[194] YALIN CHEN, JUE-SAM CHOU, HUNG-MIN SUN, AND MING-HSUN CHO. **A novel electronic cash system with trustee-based anonymity revocation from pairing**. *Electronic Commerce Research and Applications*, **10**(6):673–682, 2011.

[195] Y BASERI, B TAKHTAEI, AND J MOHAJERI. **Secure untraceable off-line electronic cash system**. *Scientia Iranica*, **20**(3):637–646, 2013.

[196] CONG WANG, HONGXIANG SUN, HUA ZHANG, AND ZHENGPING JIN. **An improved off-line electronic cash scheme**. In *Fifth IEEE International Conference on. Computational and Information Sciences (ICCIS'2013), Shiyang, China, June 21 - 23, 2013*, pages 438–441.

[197] SATTAR J ABOUD. **Analysis of Offline E-cash Schemes**. *International Journal*, **2**(8):406–410, 2014.

[198] CHUN-I FAN, WEI-ZHE SUN, AND HOI-TUNG HAU. **Date attachable offline electronic cash scheme**. *The Scientific World Journal*, **2014**, 2014.

[199] YALIN CHEN AND JUE-SAM CHOU. **Cryptanalysis on" Secure untraceable off-line electronic cash system".** *IACR Cryptology ePrint Archive*, **2014**:63, 2014.

[200] YALIN CHEN AND JUE-SAM CHOU. **On the Privacy of" User Efficient Recoverable Off-Line E-Cash Scheme with Fast Anonymity Revoking".** *IJ Network Security*, **17**(6):708–711, 2015.

[201] FENG WANG, CHIN-CHEN CHANG, AND CHANGLU LIN. **Security Analysis on Secure Untraceable Off-line Electronic Cash System.** *IJ Network Security*, **18**(3):454–458, 2016.

[202] FENG WANG, CHIN-CHEN CHANG, AND SHIH-CHANG CHANG. **Robust offline e-cash scheme with recoverability and anonymity revocation**. *Security and Communication Networks*, **9**(14):2412–2421, 2016.

[203] RON BORZEKOWSKI, K KISER ELIZABETH, AND AHMED SHAISTA. **Consumers' use of debit cards: patterns, preferences, and price response**. *Journal of Money, Credit and Banking*, **40**(1):149–172, 2008.

[204] ANDRÉ LEVISSE, NIMAL MANUEL, AND MARTIN SJOLUND. **Getting more from prepaid mobile services**. *McKinsey Quarterly*, 2008.

[205] AMIR TABAKOVIC. **The prepaid mobile wallet: A powerful product for an impatient ecosystem**. *Journal of Payments Strategy & Systems*, **8**(3):254–263, 2014.

[206] JING-JANG HWANG, TZU-CHANG YEH, AND JUNG-BIN LI. **Securing online credit card payments without disclosing privacy information**. *Computer Standards & Interfaces*, **25**(2):119–129, 2003.

[207] SUPAKORN KUNGPISDAN, BALA SRINIVASAN, AND PHU DUNG LE. **Lightweight mobile credit-card payment protocol**. In *International Conference on Cryptology in India*, pages 295–308. Springer, 2003.

[208] YINGJIU LI AND XINWEN ZHANG. **A security-enhanced one-time payment scheme for credit card**. In *Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications, 2004. Proceedings. 14th International Workshop on*, pages 40–47. IEEE, 2004.

[209] CHI PO CHEONG. **A Simple and Secure Credit Card-Based Payment System**. In *Electronic Services: Concepts, Methodologies, Tools and Applications*, pages 834–842. IGI Global, 2010.

[210] SHALINI GUPTA AND RAHUL JOHARI. **A new framework for credit card transactions involving mutual authentication between cardholder and merchant**. In *IEEE International Conference on. Communication Systems and Network Technologies (CSNT'2011), Katra, Jammu, India, June 3-5, 2011*, pages 22–26.

[211] AHMED WATHIK NAJI, ANAS S HOUSAIN, BB ZAIDAN, AA ZAIDAN, AND SHIHAB A HAMEED. **Security improvement of credit card online purchasing system**. *Scientific Research and Essays*, **6**(16):3357–3370, 2011.

[212] SOMAYEH NADERI VESAL AND MOHAMMAD FATHIAN. **Efficient and secure credit card payment protocol for mobile devices**. *International Journal of Information and Computer Security*, **5**(2):105–114, 2012.

[213] KUO-JUI WEI, JUNG-SAN LEE, AND SHIN-JEN CHEN. **Enhancing the Security of Credit Card Transaction based on Visual DSC**. *KSII Transactions on Internet and Information Systems (TIIS)*, **9**(3):1231–1245, 2015.

[214] I-MODE NTT DOCOMO. **Mobile payments via phone bill**. `https://www.nttdocomo.co.jp/`, Nov 2017.

[215] SUPAKORN KUNGPISDAN. **Accountability of centralized payment systems: formal reasoning, protocol design and analysis**. *IETE Technical Review*, **27**(5):351–364, 2010.

[216] MAHMOUD ELKHODR, SEYED SHAHRESTANI, AND KALED KOUROUCHE. **A proposal to improve the security of mobile banking applications**. In *10th IEEE International Conference oN. ICT and Knowledge Engineering, Siam University, Bangkok, Thailand, November 21 - 23, 2012*, pages 260–265.

[217] LEILA ESMAEILI, ZEINAB BORHANI-FARD, AND MOHAMMAD ALI ARASTEH. **A Survey on Mobile Payment Systems Security**. *Research Journal of Applied Sciences, Engineering and Technology*, **4**(20):4043–4050, 2012.

[218] ANURAG KUMAR JAIN AND DEVENDRA SHANBHAG. **Addressing security and privacy risks in mobile applications**. *IT Professional*, **14**(5):28–33, 2012.

[219] TAO ZHOU. **An empirical examination of initial trust in mobile payment**. *Wireless personal communications*, **77**(2):1519–1531, 2014.

[220] LEILI NOSRATI AND AMIR MASSOUD BIDGOLI. **Security assessment of mobile-banking**. In *International Conference and Workshop on. Computing and Communication (IEMCON), University of British Columbia, Vancouver, Canada, October 15-17, 2015*, pages 1–5.

[221] SRIRAMULU BOJJAGANI AND VN SASTRY. **VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps**. In *IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, California, USA, October 15 - 17, 2017*, pages 77–86.

[222] SIAMAK SOLAT. **Security of Electronic Payment Systems: A Comprehensive Survey**. *arXiv preprint arXiv:1701.04556*, 2017.

[223] MARKO HASSINEN, KONSTANTIN HYPPÖNEN, AND KEIJO HAATAJA. **An open, PKI-based mobile payment system**. *Emerging trends in information and communication security*, pages 86–100, 2006.

[224] ELENA TRICHINA, KONSTANTIN HYPPÖNEN, AND MARKO HASSINEN. **SIM-enabled Open Mobile Payment System Based on Nation-wide PKI.** In *ISSE*, pages 355–366. Springer, 2007.

[225] MARKO HASSINEN, KONSTANTIN HYPPÖNEN, AND ELENA TRICHINA. **Utilizing national public-key infrastructure in mobile payment systems**. *Electronic Commerce Research and Applications*, **7**(2):214–231, 2008.

[226] NOR BADRUL ANUAR, LAI NGAN KUEN, OMAR ZAKARIA, ABDULLAH GANI, AND AINUDDIN WAHID ABDUL WAHAB. **GSM mobile SMS/MMS using public key infrastructure: m-PKI**. *WSEAS Transactions on computers*, **7**(8):1219–1229, 2008.

[227] MOHSEN TOORANI AND A BEHESHTI. **LPKI-a lightweight public key infrastructure for the mobile environments**. In *11th IEEE Singapore International Conference on. Communication Systems, (ICCS'2008), Guangzhou, China, November 19 - 21, 2008*, pages 162–166.

[228] C NARENDIRAN, S ALBERT RABARA, AND N RAJENDRAN. **Public key infrastructure for mobile banking security**. In *IEEE Global Mobile Congress, Shanghai, China, October 12 -14, 2009*, pages 1–6.

[229] LONGYI LI AND LIHUA TAO. **Security study of mobile business based on WPKI**. In *IEEE Eighth International Conference on. Mobile Business, (ICMB'2009), Dalian, China, June 27 - 28, 2009,*, pages 301–304.

[230] GAUTHIER VAN DAMME, NICOLAS LUYCKX, AND KAREL WOUTERS. **A PKI-Based Mobile Banking Demonstrator.** In *EuroPKI*, pages 147–158. Springer, 2011.

[231] MAJDA OMER ALBASHEER AND EIHAB BM BASHIER. **Enhanced model for PKI certificate validation in the mobile banking**. In *IEEE International Conference on. Computing, Electrical and Electronics Engineering (ICCEEE), Khartoum, Sudan, August 26 - 28, 2013*, pages 470–476.

[232] WANG YUE. **SEMOPIAS: A Novel Secure and Efficient Mutual Open PKI Identity Authentication Scheme for Mobile Commerce**. In *The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems, Hohhot, China, July 14-15, 2015*, pages 373–380.

[233] ARTUR ROMÃO AND MIGUEL MIRA DA SILVA. **Secure mobile agent digital signatures with proxy certificates**. In *E-Commerce Agents*, pages 206–220. Springer, 2001.

[234] CHUNG-MING OU AND CHUNG-REN OU. **Adaptation of proxy certificates to non-repudiation protocol of agent-based mobile payment systems**. *Applied Intelligence*, **30**(3):233–243, 2009.

[235] MOHAMMAD HASAN SAMADANI, MEHDI SHAJARI, AND MEHDI AHANIHA. **A survey on mobile digital signature models**. In *Proceedings of the 12th ACM International Conference on. Electronic Commerce: Roadmap for the Future of Electronic Business, (ICEC'10), Honolulu, HI, USA, August 02 - 04, 2010*, pages 141–145.

[236] SADIQ ALMUAIRFI, PRAKASH VEERARAGHAVAN, NAVEEN CHILAMKURTI, AND DOO-SOON PARK. **Anonymous proximity mobile payment (APMP)**. *Peer-to-Peer Networking and Applications*, **7**(4):620–627, 2014.

[237] KWOK-YAN LAM, SIU-LEUNG CHUNG, MING GU, AND JIA-GUANG SUN. **Lightweight security for mobile commerce transactions**. *Computer Communications*, **26**(18):2052–2060, 2003.

[238] LEVENT ERTAUL AND WEIMIN LU. **ECC based threshold cryptography for secure data forwarding and secure key exchange in MANET (I)**. In *Networking*, **5**, pages 102–13. Springer, 2005.

[239] PENG BINGA, FU CAIA, AND FU XIONGB. **Efficient mobile payment system based on ECC [J]**. *Application Research of Computers*, **9**:071, 2008.

[240] YUNPU ZHU AND JACQUELINE E RICE. **A lightweight architecture for secure two-party mobile payment**. In *International Conference on. Computational Science and Engineering, (CSE'09), Vancouver, BC, Canada, August 29 - 31, 2009*, **2**, pages 326–333.

[241] SHAGHAYEGH BAKHTIARI, AHMAD BARAANI, AND MOHAMMAD-REZA KHAYYAMBASHI. **Mobicash: A new anonymous mobile payment system implemented by elliptic curve cryptography**. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, **3**, pages 286–290. IEEE, 2009.

[242] JESÚS TÉLLEZ ISAAC, SHERALI ZEADALLY, AND JOSÉ SIERRA CÁMARA. **A lightweight secure mobile payment protocol for vehicular ad-hoc networks (VANETs)**. *Electronic Commerce Research*, **12**(1):97–123, 2012.

[243] LAWRENCE C PAULSON. **Verifying the SET protocol: Overview**. In *Formal Aspects of Security*, pages 4–14. Springer, 2003.

[244] HUY HOANG NGO, OSAMA DANDASH, PHU DUNG LE, BALA SRINIVASAN, AND CAMPBELL WILSON. **Formal verification of a secure mobile banking protocol**. In *International Conference on Computer Science and Information Technology*, pages 410–421. Springer, 2011.

[245] SHAIK SHAKEEL AHAMAD, NN SASTRY, AND SIBA K UDGATA. **Enhanced Mobile SET Protocol with Formal Verification**. In *Third IEEE International Conference on. Computer and Communication Technology (ICCCT'2012) , Uttar Pradesh, India, November 23 - 25, 2012*, pages 288–293.

[246] SHAIK SHAKEEL AHAMAD, SIBA K UDGATA, AND VN SASTRY. **A new mobile payment system with formal verification**. *International Journal of Internet Technology and Secured Transactions*, **4**(1):71–103, 2012.

[247] ALI ALSHEHRI, JOHANN A BRIFFA, STEVE SCHNEIDER, AND STEPHAN WESEMEYER. **Formal security analysis of NFC m-coupon protocols using casper/fdr**. In *5th International Workshop on. Near Field Communication (NFC), Zurich, Switzerland, Feb 05 - 05, 2013*, pages 1–6.

[248] SARAH ABUGHAZALAH, KOSTANTINOS MARKANTONAKIS, AND KEITH MAYES. **Secure mobile payment on NFC-enabled mobile phones formally analysed using CasperFDR**. In *13th IEEE International Conference on. Trust, Security and Privacy in Computing and Communications (TrustCom), Beijing, China, September 24 - 26, 2014*, pages 422–431.

[249] SRIRAMULU BOJJAGANI AND VN SASTRY. **SSMBP: A secure SMS-based mobile banking protocol with formal verification**. In *11th IEEE International Conference on. Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, UAE, Oct 19-21, 2015*, pages 252–259.

[250] VANGA ODELU, ASHOK KUMAR DAS, AND ADRIJIT GOSWAMI. **SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms**. *IEEE Transactions on Consumer Electronics*, **62**(1):30–38, 2016.

[251] NIKOLAOS ALEXIOU, STYLIANOS BASAGIANNIS, AND SOPHIA PETRIDOU. **Formal security analysis of near field communication using model checking**. *Computers & Security*, **60**:1–14, 2016.

[252] NIKOLAOS ALEXIOU, STYLIANOS BASAGIANNIS, AND SOPHIA PETRIDOU. **Security analysis of NFC relay attacks using probabilistic model checking**. In *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, August 04 - 08, 2014*, pages 524–529.

[253] ALIA FOURATI, HK BEN AYED, FAROUK KAMOUN, AND ABDELMALEK BENZEKRI. **A SET based approach to secure the payment in mobile commerce**. In *2002. Proceedings. LCN : 27th Annual IEEE Conference on. Local*

Computer Networks, Tampa, Florida, USA, November 6 - 8, 2002., pages 136–137.

[254] SUPAKORN KUNGPISDAN, BALA SRINIVASAN, AND PHU DUNG LE. **A secure account-based mobile payment protocol**. In *IEEE International Conference on. Information Technology: Coding and Computing, (ITCC'2004),Las Vegas, NV, USA, April 5 - 7, 2004*, **1**, pages 35–39.

[255] MING QI AND CHAOBO YANG. **Design and applications of wireless SET protocol**. In *Proceedings of the 3rd ACM international conference on Mobile technology, applications & systems, Bangkok, Thailand, October 25 - 27, 2006*, page 18.

[256] SRECKO BRLEK, SARDAOUNA HAMADOU, AND JOHN MULLINS. **A flaw in the electronic commerce protocol SET**. *Information Processing Letters*, **97**(3):104–108, 2006.

[257] ELIAS HADDAD AND BRIAN KING. **A simple secure M-commerce protocol SSMCP**. *IJCSNS International Journal of Computer Science and Network Security*, **7**(3):220–229, 2007.

[258] JING LI AND JINHUA LI. **Model checking the SET purchasing process protocol with SPIN**. In *5th IEEE International Conference on. Wireless Communications, Networking and Mobile Computing, (WiCom'09), Beijing, China, September 24-26, 2009*, pages 1–4.

[259] ZHONGBAO WAN AND WANG QING. **Secure mobile payment based on Super SET protocol**. In *2nd IEEE International Conference on. Advanced Computer Control (ICACC), Shenyang, China, March 27-29, 2010,*, **2**, pages 640–643.

[260] SABRINA M SHEDID AND MOHAMED KOUTA. **Modified SET protocol for mobile payment: an empirical analysis**. In *2nd IEEE International Conference on. Software Technology and Engineering (ICSTE' 2010) ,San Juan, PR, USA, October 03 - 05, 2010*, **1**, pages V1–350.

[261] VORUGUNTI CHANDRA SEKHAR AND MRUDULA SARVABHATLA. **A secure account-based mobile payment protocol with public key cryptography**. *ACEEE International Journal on Network Security*, **3**(1):5–9, 2012.

[262] SALEEM KADHIWAL AND ANWAR USMAN SHAHEED ZULFIQUAR. **Analysis of mobile payment security measures and different standards**. *Computer Fraud & Security*, **2007**(6):12–16, 2007.

[263] UPKAR VARSHNEY. **Mobile payments**. *Computer*, **35**(12):120–121, 2002.

[264] TNT NGUYEN, P SHUM, AND EH CHUA. **Secure end-to-end mobile payment system**.

[265] MONI NAOR AND ADI SHAMIR. **Visual cryptography**. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 1–12. Springer, 1994.

[266] SHIVANI AGARWAL, MITESH KHAPRA, BERNARD MENEZES, AND NIRAV UCHAT. **Security issues in mobile payment systems**. In *Proceedings of t he 5th International Conference on. E-Governance (ICEG'2007)*, pages 142–152, 2007.

[267] JAN ONDRUS, GIOVANNI CAMPONOVO, AND YVES PIGNEUR. **A proposal for a multi-perspective analysis of the mobile payment environment**. In *IEEE International Conference on. Mobile Business, (ICMB'2005), Sydney, Australia, July 11 - 13, 2005*, pages 659–662.

[268] A ZMIJEWSKA AND EM LAWRENCE. **Reshaping the framework for analysing success of mobile payment solutions**. *Proceedings of the IADIS International Conference on E-commerce, Porto, Portugal*, 2005.

[269] TOMI DAHLBERG, MILLA HUURROS, AND ANTTI AINAMO. **Lost opportunity why has dominant design failed to emerge for the mobile payment services market in Finland?** In *Proceedings of the 41st Annual Hawaii International Conference on. System Sciences, (HICSS'2008), Waikoloa, Big Island, Hawaii, January 07 - 10, 2008*, pages 83–83.

[270] JAN ONDRUS, KALLE LYYTINEN, AND YVES PIGNEUR. **Why mobile payments fail? Towards a dynamic and multi-perspective explanation**. In *Proceedings of the 42nd Annual Hawaii International Conference on. System Sciences, (HICSS'2008), Waikoloa, Big Island, Hawaii, January 05 - 08, 2009*, pages 1–10.

[271] JONAS HEDMAN AND STEFAN HENNINGSSON. **Competition and collaboration shaping the digital payment infrastructure**. In *Proceedings of the 14th ACM Annual International Conference on. Electronic Commerce, (ICEC'12), Singapore, August 07 - 08, 2012*, pages 178–185.

[272] EROL KAZAN AND JAN DAMSGAARD. **A framework for analyzing digital payment as a multi-sided platform: A study of three european NFC solutions**. pages 1–13.

[273] AVINASH GANNAMANENI, JAN ONDRUS, AND KALLE LYYTINEN. **A post-failure analysis of mobile payment platforms**. In *48th IEEE Hawaii International Conference on. System Sciences (HICSS), HI, USA, Jan 5 - 8, 2015*, pages 1159–1168.

[274] LJUPCO ANTOVSKI AND MARJAN GUSEV. **M-payments**. In *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, pages 95–100. IEEE, 2003.

[275] JERRY GAO, KRISHNAVENI EDUNURU, JACKY CAI, AND S PH D SHIM. **P2p-paid: a peer-to-peer wireless payment system**. In *The Second IEEE International Workshop on Mobile Commerce and Services, WMCS'05, Munich, Germany, July 19, 2005*, pages 102–111.

[276] RAHUL M GODBOLE AND ALWYN R PAIS. **Secure and efficient protocol for mobile payments**. In *Proceedings of the 10th international conference on Electronic commerce*, page 25. ACM, 2008.

[277] WENCHENG YANG, JIANKUN HU, JUCHENG YANG, SONG WANG, AND LEI SHU. **Biometrics for securing mobile payments: Benefits, challenges and solutions**. In *6th IEEE International Congress on. Image and Signal Processing (CISP'2013) , Hangzhou, China, December 16 - 18, 2013*, **3**, pages 1699–1704.

[278] CHRISTEL AUGSBURG AND JONAS HEDMAN. **Value added services and adoption of mobile payments**. In *ACM Proceedings of the Sixteenth International Conference on Electronic Commerce (ICEC-14), Philadelphia, PA, USA , August 05 - 06, 2014*, page 27.

[279] YUNG-HSIANG CHENG AND TING-YU HUANG. **High speed rail passengers' mobile ticketing adoption**. *Transportation Research Part C: Emerging Technologies*, **30**:143–160, 2013.

[280] TAO ZHOU. **The effect of initial trust on user adoption of mobile payment**. *Information Development*, **27**(4):290–300, 2011.

[281] TAO ZHOU, YAOBIN LU, AND BIN WANG. **Integrating TTF and UTAUT to explain mobile banking user adoption**. *Computers in human behavior*, **26**(4):760–767, 2010.

[282] KENNETH CC YANG. **Exploring factors affecting the adoption of mobile commerce in Singapore**. *Telematics and informatics*, **22**(3):257–277, 2005.

[283] SHUIQING YANG, YAOBIN LU, SUMEET GUPTA, YUZHI CAO, AND RUI ZHANG. **Mobile payment services adoption across time: An empirical study of the effects of behavioral beliefs, social influences, and personal traits**. *Computers in Human Behavior*, **28**(1):129–142, 2012.

[284] FRANCISCO LIÉBANA-CABANILLAS, JUAN SÁNCHEZ-FERNÁNDEZ, AND FRANCISCO MUÑOZ-LEIVA. **Antecedents of the adoption of the new mobile payment systems: The moderating effect of age**. *Computers in Human Behavior*, **35**:464–478, 2014.

[285] AIHUA ZHANGA, XING YUE, AND YING KONG. **Exploring culture factors affecting the adoption of mobile payment**. In *Tenth IEEE International Conference onMobile Business (ICMB), Como, Italy, June 20 - 21, 2011*, pages 263–267.

[286] RAKHI THAKUR AND MALA SRIVASTAVA. **Adoption readiness, personal innovativeness, perceived risk and usage intention across customer groups for mobile payment services in India**. *Internet Research*, **24**(3):369–392, 2014.

[287] EMMA SLADE, MICHAEL WILLIAMS, YOGESH DWIVEDI, AND NIALL PIERCY. **Exploring consumer adoption of proximity mobile payments**. *Journal of Strategic Marketing*, **23**(3):209–223, 2015.

[288] DONG-HEE SHIN. **Towards an understanding of the consumer acceptance of mobile wallet**. *Computers in Human Behavior*, **25**(6):1343–1354, 2009.

[289] DENNIS VIEHLAND AND ROSLYN SIU YOONG LEONG. **Acceptance and use of mobile payments**. *ACIS 2007 Proceedings*, page 16, 2007.

[290] TIAGO OLIVEIRA, MANOJ THOMAS, GONCALO BAPTISTA, AND FILIPE CAMPOS. **Mobile payment: Understanding the determinants of customer adoption and intention to recommend the technology**. *Computers in Human Behavior*, **61**:404–414, 2016.

[291] IKRAM DASTAN ET AL. **Factors affecting the adoption of mobile payment systems: an empirical analysis**. *Emerging Markets Journal*, **6**(1):17, 2016.

[292] JAN ONDRUS AND YVES PIGNEUR. **A multi-stakeholder multi-criteria assessment framework of mobile payments: an illustration with the Swiss public transportation industry**. In *Proceedings of the 39th IEEE Annual Hawaii International Conference on. System Sciences, 2006. (HICSS'06), Kauia, HI, USA, Janualry 4 - 7, 2006*, **2**, pages 42a–42a.

[293] SANGJO OH. **A stakeholder perspective on successful electronic payment systems diffusion**. In *Proceedings of the 39th IEEE Annual Hawaii International Conference on. System Sciences, 2006. (HICSS'06), Kauia, HI, USA, Janualry 4 - 7, 2006*, **8**, pages 186b–186b.

[294] DONG-HEE SHIN. **Modeling the interaction of users and mobile payment system: Conceptual framework**. *International Journal of Human-Computer Interaction*, **26**(10):917–940, 2010.

[295] ERIC WK SEE-TO, SAVVAS PAPAGIANNIDIS, AND J CHRISTOPHER WESTLAND. **The moderating role of income on consumers' preferences and usage for online and offline payment methods**. *Electronic Commerce Research*, **14**(2):189–213, 2014.

[296] REN-JUNN HWANG, SHENG-HUA SHIAU, AND DING-FAR JAN. **A new mobile payment scheme for roaming services**. *Electronic Commerce Research and Applications*, **6**(2):184–191, 2007.

[297] JONAS HEDMAN AND STEFAN HENNINGSSON. **The new normal: Market cooperation in the mobile payments ecosystem**. *Electronic Commerce Research and Applications*, **14**(5):305–318, 2015.

[298] LAURA GOEKE AND KEY POUSTTCHI. **A scenario-based analysis of mobile payment acceptance**. In *Ninth IEEE International Conference on. Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), Athens, Greece, June 13 - 15, 2010*, pages 371–378.

[299] ANTONIO GHEZZI, FILIPPO RENGA, RAFFAELLO BALOCCO, AND PAOLO PESCETTO. **Mobile Payment Applications: offer state of the art in the Italian market**. *info*, **12**(5):3–22, 2010.

[300] DENIS DENNEHY AND DAVID SAMMON. **Trends in mobile payments research: A literature review**. *Journal of Innovation Management*, **3**(1):49, 2015.

[301] TOMI DAHLBERG, NIINA MALLAT, AND ANSSI ÖÖRNI. **Trust enhanced technology acceptance modelconsumer acceptance of mobile payment solutions: Tentative evidence**. *Stockholm Mobility Roundtable*, **22**:23, 2003.

[302] TOMI DAHLBERG, NIINA MALLAT, JAN ONDRUS, AND AGNIESZKA ZMIJEWSKA. **Past, present and future of mobile payments research: A literature review**. *Electronic Commerce Research and Applications*, **7**(2):165–181, 2008.

[303] KEY POUSTTCHI. **A modeling approach and reference models for the analysis of mobile payment use cases**. *Electronic Commerce Research and Applications*, **7**(2):182–201, 2008.

[304] JAN ONDRUS AND YVES PIGNEUR. **Near field communication: an assessment for future payment systems**. *Information Systems and E-Business Management*, **7**(3):347–361, 2009.

[305] KEY POUSTTCHI, MAX SCHIESSLER, AND DIETMAR G WIEDEMANN. **Proposing a comprehensive framework for analysis and engineering of mobile payment business models**. *Information Systems and E-Business Management*, **7**(3):363–393, 2009.

[306] FERGAL CARTON, JONAS HEDMAN, JAN DAMSGAARD, KAY-TI TAN, AND J MCCARTHY. **Framework for mobile payments integration**. *The Electronic Journal Information Systems Evaluation*, **15**(1):13–24, 2012.

[307] JIE GUO, SHAHROKH NIKOU, AND HARRY BOUWMAN. **Business model for mobile payment in China**. *International Journal of Systems and Service-Oriented Engineering (IJSSOE)*, **5**(2):20–43, 2015.

[308] MANIK LAL DAS, ASHUTOSH SAXENA, AND VED P GULATI. **A security framework for mobile-to-mobile payment network**. In *IEEE International Conference on Personal Wireless Communications, ICPWC 2005, New Delhi, India, January 23-25, 2005*, pages 420–423.

[309] ROBERTO DI PIETRO, GIANLUIGI ME, AND MAURIZIO ADRIANO STRANGIO. **A two-factor mobile authentication scheme for secure financial transactions**. In *IEEE International Conference on. Mobile Business, (ICMB), Sydney, Australia, July 11-13, 2005*, pages 28–34.

[310] YANNIS LABROU, JONATHAN AGRE, LUSHENG JI, JESUS MOLINA, AND W-I CHEN. **Wireless wallet**. In *The First Annual International Conference on. Mobile and Ubiquitous Systems: Networking and Services, (MOBIQUITOUS-2004), Boston, MA, USA, August 22-26, 2004*, pages 32–41.

[311] OOK LEE. **Sound-based mobile payment system**. In *Proceedings of the IEEE International Conference on. Web Services, (ICWS-2004), San Diego, California, USA, July 6-9, 2004*, pages 820–821.

[312] JANNE LUKKARI, JANI KORHONEN, AND TIMO OJALA. **SmartRestaurant: mobile payments in context-aware environment**. In *ACM Proceedings of the 6th international conference on Electronic commerce, (ICEC-2004), Delft, Netherlands, October 25-27, 2004.*, pages 575–582.

[313] GIANLUIGI ME AND MAURIZIO ADRIANO STRANGIO. **EC-PAY: an efficient and secure ECC-based wireless local payment scheme**. In *Third IEEE International Conference on Information Technology and Applications, (ICITA- 2005), Sydney, Australia, July 4-7, 2005*, **2**, pages 442–447.

[314] ASHUTOSH SAXENA, MANIK LAL DAS, AND ANURAG GUPTA. **MMPS: a versatile mobile-to-mobile payment system**. In *IEEE International Conference on Mobile Business, (ICMB-2005), Sydney, Australia, July 11-13, 2005.*, pages 400–405.

[315] LEILA ABEDI, MOHAMMADALI NEMATBAKHSH, AND ABBAS ABDOLMALEKI. **A model for context aware mobile payment**. *Journal of theoretical and applied electronic commerce research*, **7**(3):1–10, 2012.

[316] ROGER CLARKE. **A risk assessment framework for mobile payments**. *BLED 2008 Proceedings*, page 40, 2008.

[317] IOANNIS G ASKOXYLAKIS, MICHAEL PRAMATEFTAKIS, DIOMEDES D KASTANIS, AND APOSTOLOS P TRAGANITIS. **Integration of a secure mobile payment system in a GSM/UMTS SIM smart card**. *System*, **12**:13, 2007.

[318] CHEN BANGDAO AND AW ROSCOE. **Mobile electronic identity: securing payment on mobile phones**. In *IFIP International Workshop on. Information Security Theory and Practices, Heraklion, Crete, Greece, June 30 - July 2, 2014,*, pages 22–37.

[319] TAN SOO FUN, LEAU YU BENG, ROZAINI ROSLAN, AND HABEEB SALEH HABEEB. **Privacy in new mobile payment protocol**. In *Proceedings of World Academy of Science, Engineering and Technology*, **30**, pages 443–447, 2008.

[320] DEVENDRA MANI TRIPATHI. **A note on modified SET protocol for mobile payment**. In *IEEE Sixth International Conference on. Internet Technology and Secured Transactions (ICITST'2011), December 11-14, Abu Dhabi, United Arab Emirates,2011*, pages 639–641.

[321] MOSTAFA HASHEM SHERIF. *Protocols for secure electronic commerce*. CRC press, 2016.

[322] MD HASIB FIROZ AND ZULFIQUAR AHMED. **Defensive protocol to ensure safe Mobile Financial Transaction in current context**. In *Third Asian Conference on. Defence Technology (ACDT), 2017, Phuket, Thailand,January 18 - 20, 2017*, pages 54–58.

[323] WANG XIN AND CHAO-WEI YUAN. **An semi-anonymity offline mobile payment protocol based on smart card**. *The Journal of China Universities of Posts and Telecommunications*, **17**:63–66, 2010.

[324] YUNPU ZHU ET AL. *A new architecture for secure two-party mobile payment transactions*. PhD thesis, Lethbridge, Alta.: University of Lethbridge, Dept. of Mathematics and Computer Science, c2010, 2010.

[325] PHONE LIN, HUNG-YUEH CHEN, YUGUANG FANG, JEU-YIH JENG, AND FANG-SUN LU. **A secure mobile electronic payment architecture platform for wireless mobile networks**. *IEEE Transactions on Wireless Communications*, **7**(7), 2008.

[326] AMIR HERZBERG. **Payments and banking with mobile personal devices**. *Communications of the ACM*, **46**(5):53–58, 2003.

[327] VAGAN TERZIYAN. **Ontological modelling of e-services to ensure appropriate mobile transactions**. *Intelligent Systems in Accounting, Finance and Management*, **11**(3):159–172, 2002.

[328] JARI VEIJALAINEN, VAGAN TERZIYAN, AND HENRY TIRRI. **Transaction management for m-commerce at a mobile terminal**. *Electronic Commerce Research and Applications*, **5**(3):229–245, 2006.

[329] SUSHILA MADAN. *Securing Transactions and Payment Systems for M-commerce*. IGI Global, 2016.

[330] DANNY TENGTI KAO. **The impact of transaction trust on consumers' intentions to adopt M-commerce: A cross-cultural investigation**. *CyberPsychology & Behavior*, **12**(2):225–229, 2009.

[331] JUN LIU, JIANXIN LIAO, AND XIAOMIN ZHU. **A system model and protocol for mobile payment**. In *IEEE International Conference on e-Business Engineering, (ICEBE-2005), Beijing, China, Oct 18-20,2005*, pages 638–641.

[332] MOHAMMED ALQAHTANI. *A framework of mobile transaction use: the user's perspective*. PhD thesis, University of East Anglia, 2015.

[333] MICHAEL GORDON AND SURESH SANKARANARAYANAN. **Biometric security mechanism in Mobile paymentts**. In *Seventh IEEE International Conference On. Wireless And Optical Communications Networks (WOCN), Colombo, Sri Lanka, September 06 - 08, 2010*, pages 1–6.

[334] N YILDIRIM AND A VAROL. **Mobile biometric security systems for today and future**. *Proceding Book*, page 86, 2014.

[335] DOYEL PAL, PRAVEENKUMAR KHETHAVATH, TINGTING CHEN, AND YUAN ZHANG. **Mobile payments in global markets using biometrics and cloud**. *International Journal of Communication Systems*, 2017.

[336] DILEEP KUMAR AND YEONSEUNG RYU. **A brief introduction of biometrics and fingerprint payment technology**. *International Journal of advanced science and Technology*, **4**:25–38, 2009.

[337] ZDENKO MEZGEC, ANDREJ MEDVED, AMOR CHOWDHURY, AND RAJKO SVEČKO. **MOBILE PAYMENTS–DESIGN OF NEW TERMINAL**. *Informacije MIDEM*, **38**(1):53–60, 2008.

[338] BAKHTIARI SHAGHAYEGH. **Using service oriented architecture in a new anonymous mobile payment system**. In *IEEE 2nd International Conference on. Software Engineering and Service Science (ICSESS), Beijing, China, July 15 - 17, 2011*, pages 393–396.

[339] WENMIN LI, QIAOYAN WEN, QI SU, AND ZHENGPING JIN. **An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network**. *Computer Communications*, **35**(2):188–195, 2012.

[340] CONSTANTIN POPESCU. **An anonymous mobile payment system based on bilinear pairings**. *Informatica*, **20**(4):579–590, 2009.

[341] LIISA KANNIAINEN. **Alternatives for banks to offer secure mobile payments**. *International Journal of Bank Marketing*, **28**(5):433–444, 2010.

[342] BENNETT GACKLE AND YANJUN ZUO. **A new approach for credit extended mobile phone payment**. *International Journal of Business Information Systems*, **15**(4):434–449, 2014.

[343] WEIHUI DAI, XIANG CAI, HAIFENG WU, WEIDONG ZHAO, AND XUAN LI. **An Integrated Mobile Phone Payment System Based on 3G Network.** *JNW*, **6**(9):1329–1336, 2011.

[344] LIANG HU, LING CHI, HONG-TU LI, WEI YUAN, YUYU SUN, AND JIANFENG CHU. **The Classic Security Application in M2M: the Authentication Scheme of Mobile Payment.** *KSII Transactions on Internet & Information Systems*, **6**(1), 2012.

[345] FALAYE ADEYINKA ADESUYI, OSHO OLUWAFEMI, ALABI ISIAQ OLUDARE, ADAMA NDAKO VICTOR, AND AMANAMBU VICTOR RICK. **Secure authentication for mobile banking using facial recognition**. *(IOSR-JCE) J. Comput. Eng.*, **10**(3):51–59, 2013.

[346] MARTA CAMPOS FERREIRA, HENRIQUETA NÓVOA, TERESA GALVÃO DIAS, AND JOÃO FALCÃO E CUNHA. **A proposal for a public transport ticketing solution based on customers' mobile devices**. *Procedia-Social and Behavioral Sciences*, **111**:232–241, 2014.

[347] TOMI DAHLBERG, JIE GUO, AND JAN ONDRUS. **A critical review of mobile payment research**. *Electronic Commerce Research and Applications*, **14**(5):265–284, 2015.

[348] KEMAL BICAKCI, DEVRIM UNAL, NADIR ASCIOGLU, AND OKTAY ADALIER. **Mobile authentication secure against man-in-the-middle attacks**. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on*, pages 273–276. IEEE, 2014.

[349] EDMUND M CLARKE, ORNA GRUMBERG, AND DORON PELED. *Model checking*. MIT Press, Cambridge, Massachusetts, 1999;, 1999.

[350] DANNY DOLEV AND ANDREW YAO. **On the security of public key protocols**. *IEEE Transactions on information theory*, **29**(2):198–208, 1983.

[351] ANTONIO MUÑOZ, ANTONIO MAÑA, AND DANIEL SERRANO. **AVISPA in the validation of ambient intelligence scenarios**. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 420–426. IEEE, 2009.

[352] E KAASENBROOD. **WiMAX security-A formal and informal analysis**. *Master's project, Eindhoven University of Technology, Department of Mathematics and Computer Science*, 2006.

[353] AHMED MOHAMED EL-AMIN, SALAH EL-AGOOZ, ALAA EL-DIN ROHIEM SHEHATA, ESSAM ABD-ELWANEES AMER, ET AL. **Design, verification and implementation of enhanced PKM WiMAX authentication protocol**. *International Journal of Computer Science and Telecommunications*, **4**:41–46, 2013.

[354] DOUGLAS R STINSON. *Cryptography: theory and practice*. Chapman & Hall/ CRC press, USA, 2005.

[355] MAHMOUD REZA HASHEMI AND ELAHE SOROUSH. **A secure m-payment protocol for mobile devices**. In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on*, pages 294–297. IEEE, 2006.

[356] C NARENDIRAN, S ALBERT RABARA, AND N RAJENDRAN. **Performance evaluation on end-to-end security architecture for mobile banking system**. In *Wireless Days, 2008. WD'08. 1st IFIP*, pages 1–5. IEEE, 2008.

[357] KRISTIN LAUTER. **The advantages of elliptic curve cryptography for wireless security**. *IEEE Wireless communications*, **11**(1):62–67, 2004.

[358] VIPUL GUPTA, SUMIT GUPTA, SHEUELING CHANG, AND DOUGLAS STEBILA. **Performance analysis of elliptic curve cryptography for SSL**. In *Proceedings of the 1st ACM workshop on Wireless security*, pages 87–94. ACM, 2002.

[359] GERALD MADLMAYR. **A mobile trusted computing architecture for a near field communication ecosystem**. In *Proceedings of the 10th international conference on information integration and web-based applications & services*, pages 563–566. ACM, 2008.

[360] BUSRA OZDENIZCI, MN AYDIN, VEDAT COSKUN, AND KEREM OK. **NFC research framework: a literature review and future research directions**. In *The 14th International Business Information Management Association (IBIMA) Conference. Istanbul, Turkey*, 2010.

[361] CHRISTIAN SAMINGER, S GRÜNBERGER, AND JOSEF LANGER. **An NFC ticketing system with a new approach of an inverse reader mode**. In *Near Field Communication (NFC), 2013 5th International Workshop on*, pages 1–5. IEEE, 2013.

[362] SHIRSHA GHOSH, ALAK MAJUMDER, JOYEETA GOSWAMI, ABHISHEK KUMAR, SARAJU P MOHANTY, AND BIDYUT K BHATTACHARYYA. **Swing-Pay: One Card Meets All User Payment and Identity Needs: A Digital Card Module using NFC and Biometric Authentication for Peer-to-Peer Payment**. *IEEE Consumer Electronics Magazine*, **6**(1):82–93, 2017.

[363] DAVID M MONTEIRO, JOEL JPC RODRIGUES, JAIME LLORET, AND SANDRA SENDRA. **A hybrid NFC–Bluetooth secure protocol for Credit Transfer among mobile phones**. *Security and Communication Networks*, **7**(2):325–337, 2014.

[364] UGO BIADER CEIPIDOR, CARLO MARIA MEDAGLIA, ANTONELLA MARINO, SERENA SPOSATO, AND ALICE MORONI. **Kernees: A protocol for mutual authentication between nfc phones and pos terminals for secure payment transactions**. In *Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on*, pages 115–120. IEEE, 2012.

[365] KAI FAN, PANFEI SONG, ZHAO DU, HAOJIN ZHU, HUI LI, YINTANG YANG, XINGHUA LI, AND CHAO YANG. **NFC Secure Payment and Verification Scheme with CS E-Ticket**. *Security and Communication Networks*, **2017**, 2017.

[366] ADRIENNE PORTER FELT, ERIKA CHIN, STEVE HANNA, DAWN SONG, AND DAVID WAGNER. **Android permissions demystified**. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 627–638. ACM, 2011.

[367] YUN-SEOK LEE, EUN KIM, AND MIN-SOO JUNG. **A NFC based authentication method for defense of the man in the middle attack**. In *Proceedings of the 3rd International Conference on Computer Science and Information Technology*, pages 10–14, 2013.

[368] MOHAMAD BADRA AND ROUBA BORGHOL BADRA. **A lightweight security protocol for NFC-based mobile payments**. *Procedia Computer Science*, **83**:705–711, 2016.

[369] MAYADA AL-TAMIMI AND ALI AL-HAJ. **Online security protocol for NFC mobile payment applications**. In *8th IEEE International Conference on Information Technology (ICIT), Amman, Jordan, May 17 - 18, 2017*, pages 827–832.

[370] PRERNA MOHIT, RUHUL AMIN, ARIJIT KARATI, GP BISWAS, AND MUHAMMAD KHURRAM KHAN. **A standard mutual authentication protocol for cloud computing based health care system**. *Journal of medical systems*, **41**(4):50, 2017.

[371] SHIN-YAN CHIOU, ZHAOQIN YING, AND JUNQIANG LIU. **Improvement of a privacy authentication scheme based on cloud for medical environment**. *Journal of medical systems*, **40**(4):101, 2016.

[372] STEVE QUIROLGICO, JEFFREY VOAS, TOM KARYGIANNIS, CHRISTOPH MICHAEL, AND KAREN SCARFONE. *Vetting the security of mobile applications*. US Department of Commerce, National Institute of Standards and Technology, US Dept. of Commerce, 2015.

[373] YONG LI, YUANYUAN ZHANG, JUANRU LI, AND DAWU GU. **icryptotracer: Dynamic analysis on misuse of cryptography functions in ios applications**. In *Proceedings. LNCS- 8792, International Conference on Network and System Security (NSS'2014), Xi'an, China, October 15-17, 2014*, pages 349–362.

[374] FANG YU, SHIN-YIN HUANG, LI-CHING CHIOU, AND RUA-HUAN TSAIH. **Clustering iOS executable using self-organizing maps**. In *IEEE International Joint Conference on. Neural Networks (IJCNN'13), Dallas, Texas, USA, Aug 4 - 9, 2013*, pages 1–8.

[375] MANUEL EGELE, CHRISTOPHER KRUEGEL, ENGIN KIRDA, AND GIOVANNI VIGNA. **PiOS: Detecting Privacy Leaks in iOS Applications.** In *Proceedings of the Network and Distributed System Security Symposium, (NDSS'11), San Diego, California, USA, 6-9 February, 2011*, pages 177–183.

[376] TIM WERTHMANN, RALF HUND, LUCAS DAVI, AHMAD-REZA SADEGHI, AND THORSTEN HOLZ. **PSiOS: bring your own privacy & security to iOS devices**. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, (ASIA CCS '13), Hangzhou, China, May 08 - 10, 2013*, pages 13–24.

[377] FENGGUO WEI, SANKARDAS ROY, XINMING OU, ET AL. **Amandroid: A precise and general inter-component data flow analysis framework for security vetting of android apps**. In *Proceedings of the SIGSAC ACM Conference on Computer and Communications Security, Scottsdale (CCS'14), AZ, USA , November 03 - 07, 2014*, pages 1329–1341.

[378] XIN CHEN AND SENCUN ZHU. **DroidJust: Automated functionality-aware privacy leakage analysis for Android applications**. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, (WiSec'15), New York, NY, USA, June 22 - 26, 2015*, page 5.

[379] YIBING ZHONGYANG, ZHI XIN, BING MAO, AND LI XIE. **DroidAlarm: an all-sided static analysis tool for Android privilege-escalation malware**. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, (ASIA CCS '13), Hangzhou, China, May 08 - 10, 2013*, pages 353–358.

[380] FANGDA CAI, HAO CHEN, YUANYI WU, AND YUAN ZHANG. **Appcracker: Widespread vulnerabilities in user and session authentication in mobile apps**. *IEEE Mobile Security Technologies (MOST), 2014*.

[381] JOHN SPAULDING, ALYSSA KRAUSS, AND AVINASH SRINIVASAN. **Exploring an open WiFi detection vulnerability as a malware attack vector on iOS devices**. In *7th IEEE International Conference on. Malicious and Unwanted Software (MALWARE'12), Fajardo, PR, USA, October 16 - 18, 2012*, pages 87–93.

[382] JOHN HUBBARD, KEN WEIMER, AND YU CHEN. **A study of SSL Proxy attacks on Android and iOS mobile applications**. In *11th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 10 Jan - 13 Jan 2014*, pages 86–91.

[383] MIN-WOO PARK, YOUNG-HYUN CHOI, JUNG-HO EOM, AND TAI-MYOUNG CHUNG. **Dangerous Wi-Fi access point: attacks to benign smartphone applications**. *Personal and ubiquitous computing*, **18**(6):1373–1386, 2014.

[384] DAVID SOUNTHIRARAJ, JUSTIN SAHS, GARRET GREENWOOD, ZHIQIANG LIN, AND LATIFUR KHAN. **Smv-hunter: Large scale, automated detection of ssl/tls man-in-the-middle vulnerabilities in android apps**. In *In Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS'14), San Diego, California, USA, February 23-26, 2014*.

[385] FRANÇOIS GAGNON, MARC-ANTOINE FERLAND, MARC-ANTOINE FORTIER, SIMON DESLOGES, JONATHAN OUELLET, AND CATHERINE BOILEAU. **AndroSSL: A platform to test Android applications connection security**. In *In: Proceedings of the International Symposium on Foundations and Practice of Security*, pages 294–302, 2015.

[386] YURY ZHAUNIAROVICH, MAQSOOD AHMAD, OLGA GADYATSKAYA, BRUNO CRISPO, AND FABIO MASSACCI. **Stadyna: Addressing the problem of dynamic code updates in the security analysis of android applications**. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY'15), San Antonio, TX, USA, March 02 - 04, 2015*, pages 37–48. ACM, 2015.

[387] YOU JOUNG HAM, HYUNG-WOO LEE, JAE DEOK LIM, AND JEONG NYEO KIM. **DroidVulMon–Android Based Mobile Device Vulnerability Analysis and**

Monitoring System. In *Seventh IEEE International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST'13), Prague, Czech Republic, September 25-27, 2013*, pages 26–31.

[388] LUYI XING, XIAORUI PAN, RUI WANG, KAN YUAN, AND XIAOFENG WANG. **Upgrading your android, elevating my malware: Privilege escalation through mobile os updating**. In *IEEE Symposium on. Security and Privacy (SP), SAN JOSE, CA, USA, MAY 18-21, 2014*, pages 393–408.

[389] APPTHORITY. https://www.appthority.com/company/press/press-releases/appthority-exposes-security-and-privacy-risks/, Accessed: 2017-04-02.

[390] STEPHEN M DYE AND KAREN SCARFONE. **A standard for developing secure mobile applications**. *Computer Standards & Interfaces*, **36**(3):524–530, 2014.

[391] OWASP-2014. https://www.owasp.org/index.php/OWASP_Mobile_Security_Projecttab=M-Security_Testing//, Accessed: 2015-06-04.

[392] SABA ARSHAD, MUNAM ALI SHAH, ABID KHAN, AND MANSOOR AHMED. **Android malware detection & protection: a survey**. *Int. J. Adv. Comput. Sci. Appl*, **7**(2):463–475, 2016.

[393] LUCAS DAVI, ALEXANDRA DMITRIENKO, AHMAD-REZA SADEGHI, AND MARCEL WINANDY. **Privilege escalation attacks on android**. In *13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010*, pages 346–360.

[394] DROZER. https://www.mwrinfosecurity.com/products/drozer/, Accessed: 2015-04-20.

[395] VIRUSTOTAL. https://www.virustotal.com/, Accessed: 2015-05-10.

[396] APKANALYSER. https://github.com/sonyxperiadev/ApkAnalyser, Accessed: 2015-02-24.

[397] DEX2JAR. https://sourceforge.net/projects/dex2jar/, Accessed: 2015-01-12.

[398] CLUTCH. http://www.cydiacrawler.com/index.php?cat=package&id=com.iphonecake.clutch2.WR6yMut97IU, Accessed: 2016-11-15.

[399] BAKSMALI. https://github.com/JesusFreke/smali, Accessed: 2016-11-18.

[400] KEYCHAIN-DUMPER. https://github.com/ptoomey3/Keychain-Dumper, Accessed: 2016-11-21.

[401] CLASS-DUMP Z. https://code.google.com/archive/p/networkpx/downloads/, Accessed: 2017-02-12.

[402] INTROSPY-ANDROID. https://github.com/iSECPartners/Introspy-Android/blob/master/Introspy-Android20Core.apk, Accessed: 2017-02-15.

[403] INTROSPY-IOS. http://isecpartners.github.io/Introspy-iOS, Accessed: 2017-02-17.

[404] CYDIA-SUBSTRATE. http://www.cydiasubstrate.com, Accessed: 2017-02-19.

[405] CYCRIPT. http://www.cycript.org, Accessed: 2017-02-19.

[406] IDB TOOL. http://www.idbtool.com/, Accessed: 2017-02-22.

[407] SQLITE BROWSER. http://sqlitebrowser.org/, Accessed: 2017-02-22.

[408] XCODER. https://developer.apple.com/xcode/download//, Accessed: 2017-03-02.

[409] FINDBUGS. https://findbugs.sourceforge.net, Accessed: 2017-02-12.

[410] BURP-SUITE. https://portswigger.net/burp/, Accessed: 2015-02-20.

[411] WIRESHARK. https://www.wireshark.org, Accessed: 2015-02-20.

[412] ANDRILLER. https://www.andriller.com/, Accessed: 2016-12-15.

[413] BLACKBAG. https://www.blackbagtech.com/software-products/blacklight-7/blacklight.html/, Accessed: 2016-12-15.

[414] IPHONE BACKUP. http://www.windowspasswordsrecovery.com/product/iphone-backup-unlocker.html/, Accessed: 2017-02-19.

[415] ENCASE. https://www.guidancesoftware.com/, Accessed: 2017-02-17.

[416] SLEUTHKIT. http://www.sleuthkit.org/index.php, Accessed: 2016-09-02.

[417] MANUEL EGELE, DAVID BRUMLEY, YANICK FRATANTONIO, AND CHRISTOPHER KRUEGEL. **An empirical study of cryptographic misuse in android applications**. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, (ASIA CCS '13), Hangzhou, China, May 08 - 10, 2013*, pages 73–84.

[418] ALEXIA CHATZIKONSTANTINOU, CHRISTOFOROS NTANTOGIAN, GEORGIOS KAROPOULOS, AND CHRISTOS XENAKIS. **Evaluation of cryptography usage in android applications**. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pages 83–90, 2016.

[419] APPCRITIQUE. https://appcritique.boozallen.com/static\/AppCritiqueSample.pdf, year= Accessed: 2017-02-01.

[420] ILSUN YOU AND KANGBIN YIM. **Malware obfuscation techniques: A brief survey**. In *IEEE International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA'10), FUKUOKA, JAPAN, November 4-6, 2010*, pages 297–300.

[421] WILLIAM ENCK, DAMIEN OCTEAU, PATRICK D MCDANIEL, AND SWARAT CHAUDHURI. **A study of android application security.** In *USENIX security symposium*, **2**, page 2, 2011.

[422] DAVID LAZAR, HAOGANG CHEN, XI WANG, AND NICKOLAI ZELDOVICH. **Why does cryptographic software fail?: a case study and open problems**. In *Proceedings of 5th ACM Asia-Pacific Workshop on Systems (APSys'14), Beijing, China — June 25 - 26, 2014*, page 7.

[423] ADRIENNE PORTER FELT AND DAVID WAGNER. *Phishing on mobile devices*. na, 2011.

[424] MICHAL ZALEWSKI. **Browser security handbook**. *Google Code*, 2010.

[425] DON JOHNSON, ALFRED MENEZES, AND SCOTT VANSTONE. **The elliptic curve digital signature algorithm (ECDSA)**. *International journal of information security*, **1**(1):36–63, 2001.

[426] BEN ADIDA. **Beamauth: two-factor web authentication with a bookmark**. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 48–57. ACM, 2007.

[427] SUN BIN, WEN QIAOYAN, AND LIANG XIAOYING. **A DNS based anti-phishing approach**. In *Second International Conference on.Networks Security Wireless Communications and Trusted Computing (NSWCTC), Wuhan, Hubei, China, 24-25 April 2010*, **2**, pages 262–265.

[428] JUNAID AHSENALI CHAUDHRY, SHAFIQUE AHMAD CHAUDHRY, AND ROBERT G RITTENHOUSE. **Phishing attacks and defenses**. *International Journal of Security and Its Applications*, **10**(1):247–256, 2016.

[429] JIE HOU AND QI YANG. **Defense against mobile phishing attack**. *Computer Security Course Project, http://wwwpersonal. umich. edu/yangqi/pivot/mobile phishing defense. pdf*, 2012.

[430] CHUN-YING HUANG, SHANG-PIN MA, WEI-LIN YEH, CHIA-YI LIN, AND CHIEN-TSUNG LIU. **Mitigate web phishing using site signatures**. In *TENCON 2010-2010 IEEE Region 10 Conference, Fukuoka, Japan, 21 - 24,- Nov 2010*, pages 803–808.

[431] CHUN-YING HUANG, SHANG-PIN MA, AND KUAN-TA CHEN. **Using one-time passwords to prevent password phishing attacks**. *Journal of Network and Computer Applications*, **34**(4):1292–1301, 2011.

[432] IL-KWON LIM, YOUNG-GIL PARK, AND JAE-KWANG LEE. **Design of Security Training System for Individual Users**. *Wireless Personal Communications*, **90**(3):1105–1120, 2016.

[433] MAHMOOD MOGHIMI AND ALI YAZDIAN VARJANI. **New rule-based phishing detection method**. *Expert systems with applications*, **53**:231–242, 2016.

[434] BRYAN PARNO, CYNTHIA KUO, AND ADRIAN PERRIG. **Phoolproof phishing prevention**. In *International Conference on Financial Cryptography and Data Security*, pages 1–19. Springer, 2006.

[435] BLAKE ROSS, COLLIN JACKSON, NICK MIYAKE, DAN BONEH, AND JOHN C MITCHELL. **Stronger Password Authentication Using Browser Extensions.** In *USENIX Security Symposium*, pages 17–32. Baltimore, MD, USA, 2005.

[436] MIN WU, ROBERT C MILLER, AND SIMSON L GARFINKEL. **Do security toolbars actually prevent phishing attacks?** In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610. ACM, 2006.

[437] HOSSAIN SHAHRIAR AND MOHAMMAD ZULKERNINE. **PhishTester: automatic testing of phishing attacks**. In *Fourth International Conference on. Secure Software Integration and Reliability Improvement (SSIRI), 2010*, pages 198–207.

[438] AUDUN JØSANG, MUHAMMED AL ZOMAI, AND SURIADI SURIADI. **Usability and privacy in identity management architectures**. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68, Ballarat, Australia, Jany 30 - Feb 02, 2007*, pages 143–152.

Appendix A

Formal Verification of Secure Mobile Payment Protocols

A.1 SMS: HLPSL specification

We specify our framework characteristics in HLPSL. We have five basic roles: payer, issuer bank, payment gateway, acquirer bank, and payee. Apart from these, we have another 2 roles, such as session, and an environment. In Figure A.1, initially receives the start signal because payer only initiating the transaction and change its state from 0 to 2 and then establish its secure connection with the issuer's bank (Ib) by sending the authentication to request $\langle Alpha', Pr_id', NPr1', Ts1' \rangle$. Before sending the request to the payer, construct a message as Alpha. The Alpha message consists of hashing of payer id, nonce, and timestamp of the payer. This information is encrypted with the issuer's bank public key and send with the help of Snd() operation. Note that secret $(NPr1', npr1, \{Pr, Ib\})$ and $secret\,(Ts1', \{Pr, Ib\})$ declaration means that payer keeps the secret information of timestamp and nonce with the issuer's bank. Now witness $(Pr, Ib, Payer_ibank_npr1, NPr1)$ this statement for authentication property Pr by Ib on NPr1, declares that agent Pr is witness for the information NPr1; this goal will be identified by the constant $payer_ibank_npr1$ in goal section.

****

role payer (Pr,Ib :agent,

pre-shared key between Payer and Issuer's bank

```
Skb1 :symmetric_ key,
Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,Place_branch,
B_code,Bank_info,PI,Amount :text,
Hash :hash_func,
SND_Ib,RCV_Pe,RCV_Ib :channel(dy))
played_by Pr def= local State :nat,
NPr1,NIb1,NPr2,NIb2,Alpha,Beta,Gamma,Ts1,Ts2,Tsib1:text,
KB,KPr :public_key
const payer,ibank,npr1,npr2,skb,payer_ibank_npr1,payer_ibank_skb1 :protocol_id
init State :=0
transition
1. State =0     ∧ RCV_Pe(start) =|>
State' :=2      ∧ NPr1' :=new()
                  ∧ Ts1' :=new()
                  ∧ Ts2' :=new()
                  ∧ NPr2' :=new()
                  ∧ Pr_id' :=(Pr_ph,Pr_mmid,Pr_mpin)
                  ∧ Bank_info' :=(B_name,Place_branch,B_code)
                  ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',Ts2',NPr2')
                  ∧ Alpha' :=Hash(Pr_id',NPr1',Ts1')
                  ∧ SND_Ib(Alpha'.Pr_id'.NPr1'.Ts1'_KB)
                  ∧ secret(NPr1',npr1,Pr,Ib)
                  ∧ secret(Ts1',Pr,Ib)
                  ∧ witness(Pr,Ib,payer_ibank_npr1,NPr1)
3. State =2     ∧ RCV_Pe(Beta'.Skb1.NIb1.Tsib1.Pr_id'_KPr) =|>
State' :=4      ∧ NPr2' :=new()
                  ∧ Ts2' :=new()
                  ∧ Bank_info' :=(B_name,Place_branch,B_code)
                  ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',Ts2',NPr2')
                  ∧ Gamma':= Hash(PI',Bank_info',Ts2',NPr2')
                  ∧ SND_Ib(Gamma'.Bank_info'.Ts2'.NPr2'_Skb1)
                  ∧ secret(NPr2',npr2,Pr,Ib)
                  ∧ witness(Pr,Ib,payer_ibank_skb1,Skb1)
```

end role
****

Figure A.1: HLPSL for Payer.

In response, the issuer's bank receives the request from the payer and decrypt the received message with bank's private key, and verify the signature message of a payer, if all the verification is successful then it generates a shared symmetric key Skb1 between Pr and Ib, and constructs a message beta which is shown in Figure A.2. Ib sends encrypted with payer's public key KPr. The message as $\langle Beta', Skbp1, NB1', Tsb1', Pr_id' \rangle$ in state 3.
After receiving this message from the Ib. The payer decrypts the message using his private key, and verify the signature message of the bank, if all the verification is successful then Pr construct a Gamma message and send this message in state 4 which is shown in Figure A.1. The message as $\langle Gamma', Bank_info', NPr2', Ts2' \rangle$ encrypted with shared symmetric key Skb1.
role ibank (Pr,Ib,Pg :agent,
Skbp1,Skb1 :symmetric_key,
Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,Place_branch,
B_code,Bank_info,PI,Amount :text,
Hash :hash_func,
Snd_Pr,Rcv_Ib,Rcv_Pr,Snd_PG :channel(dy))
played_by Ib def= local State :nat,
NPr1,NB1,NPr2,NB2,Alpha,Beta,
Gamma,Ts1,Ts2,Tsb1,Tsb2 :text,
KB,KPr :public_key
const payer,ibank,npr1,nb1,nap1,nb2,
npr2,nbpg1,skb1,ibank_payer_nb1,
ibank_payer_skb1,payer_ibank_npr1,ibank_pg_skbp1 :protocol_id
init State := 1
transition
2. State =1 ∧ Rcv_Ib(start) =|>
State':=3 ∧ NB1':= new()

```
                ∧ Tsb1' := new()
                ∧ Pr_id' :=(Pr_ph,Pr_mmid,Pr_mpin)
                ∧ Beta' :=Hash(Skb1,NB1',Tsb1,Pr_id')
                ∧ Snd_Pr(Beta'.Skbp1.NB1'.Tsb1.Pr_id'_KPr)
                ∧ secret(NB1',nb1,Pr,Ib)
                ∧ witness(Ib,Pr,ibank_payer_nb1,NB1)
4. State =3     ∧ Rcv_Ib(Alpha'.Pr_id'.NPr1'.Ts1'_KB) =|>
State' :=5      ∧ NB2' :=new()
                ∧ Pr_id' :=(Pr_ph,Pr_mmid,Pr_mpin)
                ∧ Bank_info' :=(B_name,Place_branch,B_code)
                ∧ secret(NB2',nb2,NB2')
                ∧ witness(Pr,Ib,payer_ibank_npr1,NPr1')
                ∧ witness(Ib,Pr,ibank_payer_skb1,Skb1)
5. State =5     ∧ Rcv_Pr(Gamma'.Bank_info'.Ts2'.NPr2'_Skb1) =|>
State' :=7      ∧ Tsb2' :=new()
                ∧ NB2' :=new()
                ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',Ts2',NPr2')
                ∧ Gamma':= Hash(PI',Bank_info',Ts2',NPr2')
                ∧ Snd_PG(Gamma'.Bank_info'.Tsb2'.NB2'_Skbp1)
                ∧secret(NB2',nb2,Ib,Pg)
                ∧ witness(Ib,Pg,ibank_pg_skbp1,Skbp1)
end role
**********************************************************************
```

Figure A.2: HLPSL for Issuing Bank

Upon receiving the message from payer Pr, issuer's bank Ib decrypt using a session key and checks the authenticity of the message, if verifications are successful then Ib sends a Gamma message to the payment gateway Pg in state 7. During the transaction phase, the message is encrypted with shared symmetric key of Skbp1, which is shown in Figure A.2.

Now Pg receives Gamma message from the Ib in state 3, with the secret $(NB2', nb2, \{Pg, Ib\})$ which is described in Figure A.3.

```
%%%
```

Figure A.3: HLPSL for Payment Gateway

```
    role aquirebank(Ab,Pg,Ib,Pr,Pe :agent,
Skbp1,Skpa1,Skab1 :symmetric_key,
Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,
Place_branch,B_code,Bank_info,PI,Amount :text,
Hash :hash_func,
Rcv_Ab,Snd_Pe :channel(dy))
played_by Ab def=
local State :nat,
NPr2,NB2,NAP1,NBPG1,Gamma,Ts2,
Tsb1,Tsb2,TsPG1,TsAP1:text,
Pay_status :message
const payer,ibank,npr1,npr2,skab1,nap1,nbpg1,pg_ibank_nb2,
pg_ab_skpa1,ab_pe_skab1:protocol_id
init State := 1
transition
8. State = 1     ∧ Rcv_Ab(start) =|>
State' := 3      ∧ TsPG1' := new()
                 ∧ NBPG1' := new()
                 ∧ Bank_info' :=(B_name,Place_branch,B_code)
                 ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',TsPG1',NBPG1')
                 ∧ Gamma' := Hash(PI',Bank_info',TsPG1',NBPG1')
                 ∧ secret(NBPG1',nbpg1,Pg,Ab)
                 ∧ witness(Pg,Ab,pg_ab_skpa1,Skpa1)
9. State =3      ∧ Rcv_Ab(Gamma'.Bank_info'.TsPG1'.NBPG1'_Skpa1) =|>
State' :=5       ∧ TsAP1':=new()
                 ∧ NAP1' :=new()
                 ∧ Bank_info' :=(B_name,Place_branch,B_code)
                 ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',TsAP1',NAP1')
                 ∧ Pay_status':= Hash(PI',TsAP1',NAP1')
```

∧ Snd_Pe(Pay_status'.TsAP1'.NAP1'_Skab1)
∧ secret(NAP1',nap1,Ab,Pe)
∧ witness(Ab,Pe,ab_pe_skab1,Skab1)
end role
****

Figure A.4: HLPSL for Acquiring Bank

Now, payment gateway decrypts the message and checks authentication of the message. If all verifications are successful then Pg verify the issuer's bank details of the remitter and construct a message with own time stamp TsPG1, and the random number or nonce of NBPG1. Send this message to acquirer's bank Ab, encrypted with shared symmetric key of Skpa1 in the state of 5. In Figure A.4, we have shown the code for acquirer's bank. In this role Ab receives $Rcv_Ab\langle\{Gamma', Bank_info', TsPG1', NBPG1'\}_Skpa1\rangle$ from the Pg during transaction phase. After that, Ab sends $Snd_Pe\langle\{Pay_status', TsAP1', NAP1'\}_Skab1\rangle$ to payee Pe. In this role, Ab verifies the Pg based on TsPG1 and NBPG1 included in the message. Figure A.5, contain two parts, one part is a role for payee Pe and another part we have implemented the role specification for the session. In the payee role segment, the payee has to receive payment confirmation message with the witness (Ab, Pe, ab_pe_skab1). In the another part of session segment, all the basic roles including the roles for Pr, Ib, Pg, Ab, Pe are served as instances with concrete arguments.
****
role payee(Ab,Pe,Pg :agent,
Skab1 :symmetric_key,
Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,
Place_branch,B_code,Bank_info,PI,Amount :text,
Hash :hash_func,
Rcv_Pe :channel(dy))
played_by Pe def=
local State :nat,
NPr2,NB2,NBPG1,NAP1,Gamma,Ts2,

```
Tsb1,Tsb2,TsPG1,TsAP1 :text,
Pay_status :message
const payer,ibank,npr1,npr2,nbpg1,pg_ibank_nb2,
pg_ab_skpa1,ab_pe_skab1 :protocol_id
init State := 1
transition
10. State = 1     ∧ Rcv_Pe(start) =|>
State':= 3      ∧ TsAP1':=new()
                  ∧ NAP1':=new()
                  ∧ Bank_info' :=(B_name,Place_branch,B_code)
                  ∧ PI' :=(Amount,Pe_ph,Pr_ph,Bank_info',TsAP1',NAP1')
                  ∧ Pay_status':= Hash(PI',TsAP1',NAP1')
                  ∧ Rcv_Pe(Pay_status'.TsAP1'.NAP1'_Skab1)
                  ∧ secret(NAP1',nap1,Ab,Pe)
                  ∧ witness(Ab,Pe,ab_pe_skab1,Skab1)
end role
**********************************************************************
```

Figure A.5: HLPSL for Payee

Figure A.6. Describes the role of the environment that includes intruder knowledge with all participating entities for the standard Dolev-Yao model checking [350], the overall constants and a composition of one or more sessions. The intruder can also participate as roles of legitimate users. Finally, we have specified the goals of the protocol in HLPSL. It is clear that the stated security goals of $authentcication_on\ pg_ibank_nb2$, pg_ab_skpa1 are not violated.

```
role session(Pr,Ib,Pg,Ab,Pe :agent,
Skb1,Skab1,Skbp1,Skpa1 :symmetric_key,
Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,Place_branch,
B_code,Bank_info,PI,Amount :text,
Hash :hash_func)
def= local SND_Ib,RCV_Pe,RCV_Ib,Snd_Pr,Rcv_Ib,Rcv_Pr,
Snd_PG,SPG,RPG,Rcv_Ab,Snd_Pe,RPE: channel(dy)
```

composition
payer(Pr,Ib,Skb1,Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,Place_branch,
B_code,Bank_info, PI,Amount,Hash,SND_Ib,RCV_Pe,RCV_Ib)∧
ibank(Pr,Ib,Pg,Skbp1,Skb1,Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,
Place_branch,B_code,Bank_info,PI, Amount,Hash,Snd_Pr,Rcv_Ib,Rcv_Pr,Snd_PG)∧
paygate(Ib,Pg,Ab,Skbp1,Skpa1,Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,
Place_branch,B_code,Bank_info, PI,Amount,Hash,SPG,RPG)∧
aquirebank(Ab,Pg,Ib,Pr,Pe,Skbp1,Skpa1,Skab1,Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,
B_name,Place_branch, B_code,Bank_info,PI,Amount,Hash,Rcv_Ab,Snd_Pe)
∧ payee(Ab,Pe,Pg,Skab1,Pr_id,Pr_ph,Pr_mmid,Pr_mpin,Pe_ph,B_name,Place_branch,
B_code,Bank_info,PI,Amount,Hash,RPE)
end role
****
role environment()
def=
local
Hash : hash_func
const
prr,ibb,pgg,abb,pee :agent,
skbb1,skabb1,skbpp1,skpaa1 :symmetric_key,
pr_id,pr_ph,pr_mmid,pr_mpin,pe_ph,b_name,place_branch,b_code,bank_info,pI,amount
:text
intruder_knowledge=i,prr,ibb,pgg,abb,pee
composition
session(prr,ibb,pgg,abb,pee,skbb1,skabb1,skbpp1,skpaa1,pr_id,pr_ph,pr_mmid,pr_mpin,
pe_ph,b_name, place_branch,b_code,bank_info,pI,amount,Hash)
∧ session(i,ibb,pgg,abb,pee,skbb1,skabb1,skbpp1,skpaa1,pr_id,pr_ph,pr_mmid,pr_mpin,
pe_ph,b_name, place_branch,b_code,bank_info,pI,amount,Hash)
∧ session(prr,i,pgg,abb,pee,skbb1,skabb1,skbpp1,skpaa1,pr_id,pr_ph,pr_mmid,pr_mpin,
pe_ph,b_name, place_branch,b_code,bank_info,pI,amount,Hash)
end role
****
goal

```
secrecy_of npr1,npr2,nbpg1,skpa1,skab1
authentication_on pg_ibank_nb2,pg_ab_skpa1,ab_pe_skab1,
ibank_pg_skbp1,pg_abank_skpa1
end goal
**********************************************************************
```

Figure A.6: HLPSL for session, environment and goal

A.2 SMS: SPDL specification

```
hashfunction H1,H2,H3,H4,H5,H6;
usertype PaySucc;
usertype Timestamp;
usertype PrId,PrPh,PePh,Pemmid,Prmmid,Prpin;
usertype Amt,Bname,PlaceBranch,Bcode,PI;
usertype SessionKey;
const pk1:Function;
const sk1:Function;
inversekeys(pk1,sk1);
usertype PayStatus,TransactionID;
protocol SMS(PR,IB,PG,AB,PE)
{
role PR
{
const npr1,npr2,nb2: Nonce;
var nb1: Nonce;
const skb1: SessionKey;
send_1(PR,IB,{{PrId,PrPh,Prmmid,Prpin,Bname,PlaceBranch,
Bcode,Prmmid,npr1}sk1(PR)}pk1(IB));
read_2(IB,PR,{{skb1,nb1,PrId}sk1(IB)}pk1(PR));
send_3(PR,IB,{H1(PI,Amt,PePh,Pemmid),PI,Pemmid,Amt,PePh}skb1);
read_9(IB,PR,{{PaySucc,Amt,TransactionID}sk1(IB)}pk1(PR));
```

```
claim_PR1(PR,Secret,skb1);
claim_PR2(PR,Secret,npr1);
claim_PR3(PR,Secret,nb1);
claim_PR4(PR,Secret,Prpin);
claim_PR5(PR,Niagree);
claim_PR6(PR,Nisynch);
}
role IB
{
const nb1,nb2: Nonce;
var npr1,np2: Nonce;
const skb1,skbp,skib,skpe: SessionKey;
read_1(PR,IB,{{PrId,PrPh,Prmmid,Prpin,Bname,PlaceBranch,
Bcode,Prmmid,npr1}sk1(PR)}pk1(IB));
send_2(IB,PR,{{skb1,nb1,PrId}sk1(IB)}pk1(PR));
read_3(PR,IB,{1́(PI,Amt,PePh,Pemmid),PI,Pemmid,Amt,PePh}skb1);
send_4(IB,PG,{H2(PI,Amt,nb2,Prmmid,Bname,Bcode),PI,Amt,nb2,
Prmmid,Bname,Bcode}skbp);
read_8(PG,IB,{H6(PayStatus,Amt,np2),PayStatus,Amt,np2}skib);
send_9(IB,PR,{{PaySucc,Amt,TransactionID}sk1(IB)}pk1(PR));
send_10(IB,PE,{(PaySucc,Amt,TransactionID)}skpe);
claim_IB1(IB,Secret,nb1);
claim_IB2(IB,Secret,npr1);
claim_IB3(IB,Niagree);
claim_IB4(IB,Nisynch);
}
role PG
{
const np1,np2:Nonce;
var nb2,na2:Nonce;
const skbp,skpa,skib,skpg:SessionKey;
read_4(IB,PG,H2(PI,Amt,nb2,Prmmid,Bname,Bcode),PI,Amt,nb2,
Prmmid,Bname,Bcodeskbp);
```

```
send_5(PG,AB,{H3(PI,Amt,np1,Prmmid,Bname,Bcode),PI,Amt,np1,
Prmmid,Bname,Bcode}skpa);
read_7(AB,PG,{H5(PayStatus,Amt,na2),PayStatus,Amt,na2}skpg);
send_8(PG,IB,{H6(PayStatus,Amt,np2),PayStatus,Amt,np2}skib);
claim_PG1(PG,Secret,nb2);
claim_PG2(PG,Secret,np1);
claim_PG3(PG,Niagree);
claim_PG4(PG,Nisynch);
}
role AB
{
const na1,na2:Nonce;
var np1:Nonce;
const skab,skpa,skpg:SessionKey;
read_5(PG,AB,{H3(PI,Amt,np1,Prmmid,Bname,Bcode),PI,Amt,
np1,Prmmid,Bname,Bcode}skpa);
send_6(AB,PE,{H4(PayStatus,Amt,na1),PayStatus,Amt,na1}skab);
send_7(AB,PG,{H5(PayStatus,Amt,na2),PayStatus,Amt,na2}skpg);
claim_AB1(AB,Secret,np1);
claim_AB2(AB,Secret,na1);
claim_AB3(AB,Niagree);
claim_AB4(AB,Nisynch);

    }
role PE
{
const na1:Nonce;
const skab,skpe:SessionKey;
read_6(AB,PE,{H4(PayStatus,Amt,na1),PayStatus,Amt,na1}skab);
read_10(IB,PE,{(PaySucc,Amt,TransactionID)}skpe);
claim_PE1(PE,Secret,na1);
claim_PE2(PE,Secret,skab);
claim_PE3(PE,Niagree);
```

claim_PE4(PE,Nisynch);
}
}
****

Figure A.7: SPDL for SMS Protocol.

A.3 NFC: HLPSL specification

In our proposed approach, there are five basic roles: (1) the customer; (2) the merchant; (3) the acquiring bank; (4) the issuing bank; (5) payment gateway. Apart from these, we have another two roles, such as session and an environment. Role specification for the customer in HLPSL is shown in Figure. A.8. Initially, receives the start signal because the customer (C) only initiating the transaction and change its state from 0 to 2 and then establish its secure connection with the merchant (M). The customer sending as $\langle Message1'.Cust_id'.IssuerBank_info'.NCu1'.TSc1'\rangle$ to the merchant with the help of SND() operation through a secure channel. In HLPSL specification the type declaration (dy) specifies that channel uses the Dolev-Yao attacker model [350]. The message1 consists of hashing of cust_id, IssuerBank_info, nonce and timestamp of the customer. This information is encrypted with the merchant public key. Note that secret $(NCu1', ncu1, \{C, M\})$ and $secret\,(TIDm', tidm, \{ibank, M\})$ declaration means that customer keeps the secret information of timestamp and nonce with the merchant. Now witness $(C, M, cust_merchant_ncu1, \{NCu1\})$ this statement for authentication purpose C by M on NCu1, declares that agent customer is witness for the information NCu1; this goal will be identified by the constant $cust_merchant_ncu1$ in goal section.
****

role cust(C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func,

SND_M,RCV_C,RCV_Ib :channel(dy))
played_by C def=
local State : nat,
NCu1,NM1,TSc1,Message1,Message2,Message3,Message8,PI_cib:text,
KC,KM : public_key
const cust,merchant,ncu1,nm1,ts1,km,cust_merchant_ncu1,
cust_merchant_idm,merchant_cust_km,cust_ibank_picib :protocol_id
init State := 0
transition
1. State = 0 $\wedge$ RCV_C(start) = $|>$
State'= 2 $\wedge$ NCu1':=new()
$\wedge$ TSc1':=new()
$\wedge$ NM1' :=new()
$\wedge$ TIDm' :=new()
$\wedge$ Cust_id':=(Cust_id,Custph_no,Cust_pin)
$\wedge$ IssuerBank_info':=(Issuerbank_name,Issuerbank_code)
$\wedge Message1' := Hash(Cust_id', NCu1, IssuerBank_info', TSc1)$
$\wedge SND_M(\{Message1'.Cust_id'.IssuerBank_info'.NCu1'.TSc1'\}_KM)$
$\wedge$ witness(C,M,cust_merchant_ncu1,NCu1)
3.State = 2 $\wedge$ RCV_C(Message2'.POIm.Cust_id'.IssuerBank_info'.Amt_m.NCu1'
.NM1'.Mer_id.TIDm'.TSc1'_KC) = $|>$
State' := 4 $\wedge$ POIm':=new()
$\wedge$ PI_cib':=(POIm'.Amt_m.NCu1'.TSc1'_Kcib)
$\wedge$ Message3' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',
Amt_c,Mer_id,PI_cib')
$\wedge$ SND_M(Message3'.PI_cib'.POIm.Cust_id'.IssuerBank_info'.NCu1'.
NM1'.Mer_id.TIDm'.TSc1'_KC)
$\wedge$ secret(NCu1',ncu1,C,M)
$\wedge$ witness(C,M,cust_merchant_ncu1,NCu1)
$\wedge$ witness(C,M,cust_merchant_idm,Mer_id)
$\wedge$ witness(C,Ib,cust_ibank_picib,PI_cib)
14. State=4 $\wedge RCV_C(Message8'.Amt_m.TIDm'.Amt_m.Payment_notify'_KC)$
= $|>$

State' :=6 $\wedge Amt_m' := new()$
$\wedge Message8' := Hash(Amt_m, TIDm', Amt_m, Payment_notify)$
$\wedge secret(TIDm', tidm, ibank, M)$
$\wedge$ witness(C,M,cust_merchant_idm,Mer_id)
end role
****

Figure A.8: Role specification for the customer in HLPSL

role merchant (C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func,
SND_M,RCV_C,RCV_Ib,RCV_M,SND_Ab,SND_C :channel(dy))
played_by M def=
local State : nat,
NCu1,NM1,TSc1,Message1,Message2,Message3,Message4,Message10,PI_cib :text,
KC,KM,Kab : public_key
const customer,merchant,ncu1,nm1,ts1,km,cust_merchant_idm,merchant_cust_km,
cust_ibank_picib,merchant_cust_nm1 :protocol_id
init State := 0
transition
2. State = 1 $\wedge$ RCV_M(start) = $|>$
State' := 3 $\wedge$ NM1':=new()
$\wedge$ TIDm':=new()
$\wedge$ Mer_id':=new()
$\wedge$ NCu1':=new()
$\wedge$TSc1':=new()
$\wedge$ Amt_m':=new()
$\wedge$ Message2' :=Hash(POIm,Amt_c,Amt_m,TIDm',NM1',TIDm',
Mer_id',NCu1',TSc1')

∧ SND_C(Message2'.POIm.Cust_id.IssuerBank_info.Amt_m.NCu1'.
NM1'.Mer_id.TIDm'.TSc1'_KC)
∧ secret(NM1',nm1,M,C)
∧ witness(M,C,merchant_cust_nm1,NM1)
4. State = 3 ∧ RCV_M(Message1'.Cust_id.IssuerBank_info'.NCu1'.TSc1'_KM) =
|>
State':= 5 ∧ Cust_id':=(Cust_id,Custph_no,Cust_pin)
∧ IssuerBank_info':=(Issuerbank_name,Issuerbank_code)
∧ Message1' :=Hash(Cust_id',NCu1,IssuerBank_info',TSc1)
∧ TIDm':=new()
∧ POIm':=new()
∧ Mer_id':=new()
∧ PI_cib':=(POIm'.Amt_m.NCu1'.TSc1'_Kcib)
∧ Message4' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',
Amt_c,Mer_id',PI_cib')
∧ SND_Ab(Message4'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.Amt_c.
Mer_id'.PI_cib'_Kab)
∧ secret(NCu1',ncu1,C,M)
∧ witness(C,M,cust_merchant_ncu1,NCu1)
∧ witness(C,M,cust_merchant_idm,Mer_id)
∧ witness(C,Ib,cust_ibank_picib,PI_cib)
5. State = 5 ∧ RCV_M(Message3'.PI_cib'.POIm.Cust_id'.IssuerBank_info'.
NCu1'.NM1'.
Mer_id.TIDm'.TSc1'_KC) = |>
State' := 7 ∧ Message3' :=Hash(POIm,Amt_m,NCu1',TSc1',TIDm',Amt_c,Mer_id,
PI_cib')
∧ secret(NCu1',ncu1,C,M)
∧ witness(C,M,customer_merchant_ncu1,NCu1)
∧ witness(C,M,cust_merchant_idm,Mer_id)
∧ witness(C,Ib,cust_ibank_picib,PI_cib)
17. State =7 ∧ RCV_M(Message10'.Amt_m.TIDm'.Amt_m.Payment_notify'_KM)
= |>
State' :=9 ∧ Amt_m':=new()

∧ Message10' :=Hash(Amt_m,TIDm,Amt_m',Payment_notify)
∧ secret(TIDm',tidm,Ib,M)
∧ witness(C,M,cust_merchant_idm,Mer_id)
end role
****

Figure A.9: Role specification for the merchant in HLPSL

In response, the merchant receives the request from the customer and decrypt the received message with merchant's private key, and verify the signature message of a customer, if all the verification is successful then merchant constructs a payment order information which is shown in Figure A.9. The message 2 is $\langle Message2'.POIm.Cust_id.IssuerBank_info.Amt_m.NCu1'.NM1'.Mer_id\rangle$ in state 3. Which is encrypted with the customer's public key using SND_Ab operation. At this moment the merchant also sends the same information to encrypt with the acquiring bank public key and send to acquiring the bank in state 5. Now merchant constructed message is received by the acquiring bank as shown in Figure A.10. After receiving a message from the merchant, decrypt with a secret key and perform signature validation procedure. If all verifications are successful then acquiring bank sends payment order information to the payment gateway for settle the transaction and amount between customer and merchant. The message5 is for payment gateway using SND_Pg operation in state 3. The message5 is $\langle Message5'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.Amt_c.Mer_id'.PI_cib'\rangle$.

****
role aquirebank(C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func,
SND_M,RCV_C,RCV_Ib,RCV_Ab,SND_Pg :channel(dy))
played_by Ab def=
local State : nat,
NCu1,NM1,TSc1,Message1,Message2,Message3,Message5,Message9,Message4,

Message10,PI_cib :text,
KC,KM,Kab,Kpg : public_key
const customer,merchant,ncu1,nm1,ts1,km,cust_merchant_ncu1,
cust_merchant_idm,merchant_cust_km,cust_ibank_picib :protocol_id
init State := 0
transition
6. State = 1 ∧ RCV_Ab(start) = |>
State' := 3 ∧ PI_cib':=(POIm'.Amt_m.NCu1'.TSc1'_Kcib)
∧TIDm':=new()
∧ Mer_id':=new()
∧TSc1':=new()
∧POIm':=new()
∧ NCu1':=new()
∧ Message4' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',
Amt_c,Mer_id,PI_cib')
∧ RCV_Ab(Message4'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.Amt_c.
Mer_id.PI_cib'_Kab)
∧ secret(NCu1',ncu1,C,M)
∧ witness(C,M,cust_merchant_ncu1,NCu1)
∧ witness(C,M,cust_merchant_idm,Mer_id)
∧ witness(C,Ib,cust_ibank_picib,PI_cib)
7. State =3 ∧ SND_Pg(Message5'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.Amt_c.
Mer_id'.PI_cib'_Kpg) = |>
State' :=5 ∧ NM1':=new()
∧ Amt_m':=new()
∧ Message5' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',
Amt_c,Mer_id',PI_cib')
∧ secret(NCu1',ncu1,C,M)
∧ witness(C,M,cust_merchant_ncu1,NCu1)
∧ witness(C,M,cust_merchant_idm,Mer_id)
∧ witness(C,Ib,cust_ibank_picib,PI_cib)
16. State =5 ∧ RCV_Ab(Message9'.Amt_m.TIDm'.Amt_m.Payment_notify'_Kab)
= |>

State' :=7 ∧ Amt_m':=new()
∧ Message9' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
∧ secret(TIDm',tidm,Ib,M)
∧ witness(C,M,cust_merchant_idm,Mer_id)
∧ Message10' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
∧ SND_M(Message10'.Amt_m.TIDm'.Amt_m.Payment_notify'_KM)
end role
****

Figure A.10: Role specification for the acquiring bank in HLPSL

After receiving the message from the acquiring bank, the payment gateway decrypts using a secret key and checks the authenticity of the message, if verifications are successful then payment gateway sends message6 to the issuing bank for confirmation of the customer account details and tally the customer account amount should be greater than the purchased item amount. The payment gateway message6 as shown in Figure A.11 using SND_Ib() operation in state 3, with the secret $(NCu1', ncu1, C, M)$. Upon receiving the message6 from the payment gateway, the issuing bank decrypts the information with his secret key and verifies authentication of the message. Then verifying the customer account like valid client account, sufficient funds are available, checks the nonce of N_C equal to the N_M, and other verifications. If all checks are valid, then the issuing bank sends payment notification whether successful/failure to the customer as well as issuing bank as shown in Figure A.12. The issuing bank constructed messages and encrypted with the public keys of respective entities of one for payment gateway and another one for customer is $Snd_Pg\,\langle\{Message7'.Amt_m.TIDm'.Amt_m.Payment_notify'\}_Kpg\rangle$
and $Snd_C\,\langle\{Message8'.Amt_m.TIDm'.Amt_m.Payment_notify'\}_KC\rangle$.
Upon receiving above messages from issuing bank, the payment gateway informs to the acquiring bank through message9 in state 9 and transition 15 which is shown in Figure A.11. Using secret $(TIDm', tidm, \{Ib, M\})$ along with witness $(C, M, customer_merchant_idm, Mer_id)$ specified in goals section. The acquiring bank receives message9 from payment gateway and decrypts with bank secret key which as shown in Figure A.10 the state is 5 and transition number 16 using RCV_Ab()

operation. Finally, the acquiring bank sends valid information to the merchant using message10 in state 7 with the SND_M() operation. Now the merchant verifies this information from the acquiring bank. The verifications are: whether purchased item order valid, customer account details. If all these verifications are valid, then amount credited to the merchant account. For this simulation, there is no question replay and man-in-the-middle attacks because each time verification of nonce and timestamps from entities and it is working in the mode of Dolev-Yao attacker model [350].

```
**********************************************************************
role paygate(C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func,
SND_M,RCV_C,RCV_Ib,SND_Ab,RCV_Pg,SND_Ib :channel(dy))
played_by Ab def=
local State : nat,
NCu1,NM1,TSc1,Message1,Message2,Message3,Message5,Message6,Message7,
aMessage9,PI_cib :text,
KC,KM,Kpg,Kib,Kab : public_key
const customer,merchant,ncu1,nm1,ts1,km,customer_merchant_ncu1,
merchant_customer_km :protocol_id
init State := 0
transition
8. State =1     ∧ RCV_Pg(start) = |>
State' :=3      ∧ NM1':=new()
                  ∧ TIDm':=new()
                  ∧ NCu1':=new()
                  ∧ POIm':=new()
                  ∧ PI_cib':=new()
                  ∧ Mer_id':=new()
                  ∧ TSc1':=new()
                  ∧ Amt_m':=new()
                 ∧ Message5' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',Amt_c,
```

```
Mer_id',PI_cib')
                ∧ RCV_Pg(Message5'.POIm'.Amt_m.NCu1'.TSc1'.
TIDm'.Amt_c.Mer_id'.PI_cib'_Kpg)
                ∧ NM1':=new()
                ∧ TIDm':=new()
                ∧ Mer_id':=new()
                ∧ Amt_m':=new()
                ∧ Message6' :=Hash(POIm',Amt_m,NCu1',TSc1',TIDm',
Amt_c,Mer_id',PI_cib')
9. State =3   ∧ SND_Ib(Message6'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.Amt_c.
Mer_id'.PI_cib'_Kib) = |>
State' :=5    ∧ secret(NCu1',ncu1,C,M)
                ∧ witness(C,M,customer_merchant_ncu1,NCu1)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
                ∧ witness(C,Ib,customer_issuerbank_picib,PI_cib)
13. State =7  ∧ RCV_Pg(Message7'.Amt_m.TIDm'.Amt_m.Payment_notify'_Kpg)
= |>
State' :=9    ∧ Amt_m':=new()
                ∧ Message7' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
                ∧ secret(TIDm',tidm,Ib,M)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
15. State =9  ∧ SND_Ab(Message9'.Amt_m.TIDm'.Amt_m.Payment_notify'_Kab)
= |>
State' :=11   ∧ Amt_m':=new()
                ∧ Message9' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
                ∧ secret(TIDm',tidm,Ib,M)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
end role
**********************************************************************
```

Figure A.11: Role specification for the payment gateway in HLPSL

In Figure A.13 describes the role of session and environment. The diagram depicts that it includes intruder knowledge with all participating entities for the standard Dolev-Yao model. Finally, we described goals section. In this section, there are four security goals, and four authentications are verified in this figure, which is presented subsequently.

- Secrecy_of ncu1:
- Secrecy_of nm1:
- Secrecy_of kcibb:
- Secrecy_of picib:
- Authentication_on cust_issuerbank_picib:
- Authentication_on cust_merchant_ncu1:
- Authentication_on merchant_cust_nm1:
- Authentication_on cust_merchant_idm:

****

```
role ibank(C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func,
SND_C,SND_M,RCV_C,RCV_Ib,SND_Pg :channel(dy))
played_by Ab def=
local State : nat,
NCu1,NM1,TSc1,Message1,Message2,Message3,Message6,Message7,
Message8,PI_cib :text, KC,KM,Kib,Kpg : public_key
const customer,merchant,ncu1,nm1,tidm,ts1,km,customer_merchant_ncu1,
merchant_customer_km :protocol_id
init State := 0
transition
```

```
10. State =1    ∧ RCV_Ib(start) = |>
State' :=3      ∧ NM1':=new()
                ∧ TIDm':=new()
                ∧ Mer_id':=new()
                ∧ PI_cib':=new()
                ∧TSc1':=new()
                 ∧ NCu1':=new()
                ∧POIm':=new()
                ∧ Amt_m':=new()
                ∧ Message6' :=Hash(POIm',Amt_m,
NCu1',TSc1',TIDm',Amt_c,Mer_id',PI_cib')
                ∧ RCV_Ib(Message6'.POIm'.Amt_m.NCu1'.TSc1'.TIDm'.
Amt_c.Mer_id'.PI_cib'_Kib)
                ∧ secret(NCu1',ncu1,C,M)
                ∧ witness(C,M,customer_merchant_ncu1,NCu1)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
                ∧witness(C,Ib,customer_issuerbank_picib,PI_cib)
11. State =3    ∧ SND_Pg(Message7'.Amt_m.TIDm'.Amt_m.Payment_notify'_Kpg)
= |>
State' :=5      ∧ Amt_m':=new()
                ∧ Message7' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
                ∧ secret(TIDm',tidm,Ib,M)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
12. State =5    ∧ SND_C(Message8'.Amt_m.TIDm'.Amt_m.Payment_notify'_KC) =
|>
State' :=7      ∧ Amt_m':=new()
                ∧ Message8' :=Hash(Amt_m,TIDm',Amt_m,Payment_notify')
                ∧secret(TIDm',tidm,Ib,M)
                ∧ witness(C,M,customer_merchant_idm,Mer_id)
end role
**************************************************************
```

Figure A.12: Role specification for the issuing bank in HLPSL

```
**********************************************************************
role session (C,M,Ab,Pg,Ib :agent,
Kcib :symmetric_key,
Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,Amt_m,Issuerbank_name,
IssuerBank_info,Issuerbank_code,Payment_notify :text,
Hash :hash_func)
def=
local SND_M,RCV_C,RCV_Ib,RCV_Ab,SND_Pg,SND_C,RCV_M,SND_Ab,RCV_Pg,
SND_Ib :channel(dy)
composition
cust(C,M,Ab,Pg,Ib,Kcib,Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,
Amt_m,Issuerbank_name,IssuerBank_info,
Issuerbank_code,Payment_notify,Hash,SND_M,RCV_C,RCV_Ib)
∧ merchant(C,M,Ab,Pg,Ib,Kcib,Cust_id,Custph_no,Cust_pin,Mer_id,POIm,
TIDm,Amt_c,Amt_m,Issuerbank_name,IssuerBank_info,Issuerbank_code,
Payment_notify,Hash,SND_M,RCV_C,RCV_Ib,RCV_M,SND_Ab,SND_C)
∧ aquirebank(C,M,Ab,Pg,Ib,Kcib,Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,
Amt_c,Amt_m,
Issuerbank_name,IssuerBank_info,Issuerbank_code,Payment_notify,Hash,
SND_M,RCV_C,RCV_Ib,RCV_Ab,SND_Pg)
∧ paygate(C,M,Ab,Pg,Ib,Kcib,Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,Amt_c,
Amt_m,Issuerbank_name,IssuerBank_info,Issuerbank_code,Payment_notify,
Hash,SND_M,RCV_C,RCV_Ib,SND_Ab,RCV_Pg,SND_Ib)
∧ibank(C,M,Ab,Pg,Ib,Kcib,Cust_id,Custph_no,Cust_pin,Mer_id,POIm,TIDm,
Amt_c,Amt_m,Issuerbank_name,IssuerBank_info,Issuerbank_code,
Payment_notify,Hash,SND_C,SND_M,RCV_C,RCV_Ib,SND_Pg)
end role
role environment()
def=
local
Hash : hash_func
```

```
const
cust1,mer,abb,pgg,ibb :agent,
kcibb :symmetric_key,
cust_id,custph_no,cust_pin,mer_id,poim,tidm1,amt_c,amt_m,issuerbank_name,
issuerBank_info,issuerbank_code,payment_notify :text
intruder_knowledge=i,cust1,mer,abb,pgg,ibb
composition
session(cust1,mer,abb,pgg,ibb,kcibb,cust_id,custph_no,cust_pin,mer_id,poim,tidm1,
amt_c,amt_m,issuerbank_name, issuerBank_info,issuerbank_code,payment_notify,Hash)
/\session(i,mer,abb,pgg,ibb,kcibb,cust_id,custph_no,cust_pin,mer_id,poim,tidm1,
amt_c,amt_m,issuerbank_name, issuerBank_info,issuerbank_code,payment_notify,Hash)
/\session(cust1,i,abb,pgg,ibb,kcibb,cust_id,custph_no,cust_pin,mer_id,poim,tidm1,
amt_c,amt_m,issuerbank_name, issuerBank_info,issuerbank_code,payment_notify,Hash)
end role
**********************************************************************
```

Figure A.13: Role specification for the session, environment in HLPSL

A.4 HLPSL specification for security against phishing attack

One of the formal model checking tools is called AVISPA the web interface is [31], [34]. AVISPA is a commonly used verification tool to analyze the network security features of the protocols. AVISPA supports a language known as the HLPSL [36] to describe the cryptographic protocols and to specify their intended security properties, along with a set of tools to formally validate them. It provides a modular and expressive formal language to specify cryptographic protocols that implement a variety of state-of-the-art automatic security analysis techniques. To analyze with AVISPA tool by automated security protocol used in mobile devices [33], it concludes that one can work with any network models especially in the environments of ad hoc and wireless. The proposed model of HLPSL specification for our protocol is as given below. The

HLPSL specification uses participants: mobile user, bank, an authentication server and DNS server of the proposed protocol that is being run. In this section we describe the HLPSL codes in the environment role, specifying an intruder's initial knowledge, overall three sessions of the protocol such as role session, the role of environment and finally goals.

role mobuser (Mu,B,As :agent,
Ku,Kas :public_key,
Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,Link,Rc_mobuser,B_name,Place_branch,B_code,
Bank_webinfo,Auth_info,As_info,As_name,As_id :text,
Hash :hash_func,
SND_B,RCV_Mu,RCV_B,SND_As :channel(dy))
played_by Mu def=
local State :nat,
NMu1,Alpha,Beta,NAs1 :text
const mobuser,bank,authser,nas1,mobuser_authser_nas1,
bank_webinfo,auth_info,ts,rc_mobuser :protocol_id
init State :=0
transition
1. State =0 ∧ RCV_Mu(start) = |>
State' :=2 ∧ NMu1' :=new()
∧ Mu_id' :=(Mu_ph)
∧ Rc_mobuser' :=(Mu_usr,Mu_pwd)
∧ Bank_webinfo':=(Url,Link)
∧ Auth_info':=(B_name,Place_branch,B_code)
∧ SND_B(Bank_webinfo'.Mu_id')
∧ secret(Bank_webinfo',bank_webinfo,Mu,B)
4. State =2 ∧ SND_As(Mu_id'.Rc_mobuser') = |>
State' :=3 ∧ NAs1' :=new()
∧ As_info' :=(As_name,As_id)
∧ secret(Rc_mobuser',rc_mobuser,Mu,As)
8. State =3 ∧ RCV_Mu(NAs1'.As_info') = |>
State' :=5 ∧ Auth_info':=(B_name,Place_branch,B_code)
∧ Alpha' :=Hash(NAs1'.Auth_info')

```
        ∧ SND_As(Alpha'.Auth_info'_inv(Ku))
        ∧ secret(NAs1,nas1,Mu,As)
end role
**********************************************************************
```

Figure A.14: Mobileuser HLPSL code for secure against phishing

In our proposed framework of security against phishing attack, there are three basic roles: mobile user (Mu), bank (B) and an authentication server (As), which represent the participants. We have moreover specified the roles of a session, and the environment and the goals in our implementation. Figure A.14, depicts the implemented role of Mu in HLPSL. Initially, the Mu sends the request to bank $\langle Bank_webinfo, Mu_{id} \rangle$ by using $SND_B()$ operation. For every role, we must specify that the type declaration as a channel $channel(dy)$ indicates the channel follows the Dolev-Yao intruder model [350]. The adversary $\mathcal{A}$ has the capability to modify, intercept and analyze the secure messages transmitted over the insecure channel. Now, Bank server (B) redirects this information to As which as depicted in Figure A.15.

```
**********************************************************************
role bank (Mu,B,As :agent,
Ku,Kas :public_key,
Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,Link,B_name,Place_branch,B_code,
Bank_webinfo,Auth_info,As_info,As_name,As_id :text,
Hash :hash_func,
SND_B,RCV_Mu,RCV_B,SND_As :channel(dy))
played_by Mu def=
local State :nat,
NMu1,Alpha,Beta,NAs1,Ts :text
const mobuser,bank,authser,bank_authser_ts,bank_authser_auth_info,
auth_info,ts :protocol_id
init State :=0
transition
2. State =0     ∧ RCV_B(start) = |>
```

```
State' :=2      ∧ Mu_id' :=(Mu_ph)
                ∧ Bank_webinfo':=(Url,Link)
                ∧Auth_info':=(B_name,Place_branch,B_code)
                ∧ SND_As(Bank_webinfo'.Auth_info')
                ∧ secret(Auth_info',auth_info,B,As)

   7. State =2  ∧ RCV_B(Bank_webinfo'.Mu_id') = |>
State' :=5      ∧ Ts' :=new()
                ∧ Auth_info':=(B_name,Place_branch,B_code)
                ∧ Beta':=Hash(Mu_id',Auth_info',Ts')
11. State =5    ∧ RCV_B(Beta'.Auth_info'.Ts'_inv(Kas)) = |>
State' :=7      ∧ secret(Auth_info',auth_info,B,As)
                ∧ secret(Ts',ts,B,As)
end role
**********************************************************************
```

Figure A.15: Bank's HLPSL code for secure against phishing

The As in Figure A.16, received a request from the bank server, As sends a nonce value to the mobile user which is to be signed. Now mobile user constructs a message called 'Alpha', which contains a hashing of nonce value and authentication information
$Alpha = hash\,\langle NAs1, Auth_info\rangle$ to the SND_As(). The mobile user waits for an authentication success acknowledgment from the bank via the As.
Figure A.15, depicts the implementation of the role of the bank server in HLPSL. After receiving the request message from the mobile user $\langle Bank_webinfo, Mu_{id}\rangle$ by using $SND_B()$, bank server (B), redirects to As over secure channel and waits for context information message. For our implementation, the context information (CI) renamed as Beta and verifies this message for authentication and integrity purpose. Finally, the bank sends authentication successful response as an acknowledgment to a mobile user, if all of the information specification is correct.
In Figure A.16, we implemented the role of the As. The mobile user sends the alpha message to As during state 5, now, As verifies this signature message for

the purposes of authentication and data integrity and confirms that whether mobile user Mu and bank information are the same or not. Then, As constructs a Beta message based on hashing of Mu id, timestamp, and authentication information. $Beta = hash\,\langle Mu_id, Auth_info, Ts\rangle$

Send Beta message to the bank server (B). In HLPSL specification contains the term secret $(Auth_info, \{Mu, As\})$ related to describing security goals. We would like to ensure that the exchanged message and the generated nonce N are kept secret among participating entities such as As and Mu. role authser (Mu,B,As :agent,

Ku,Kas :public_key,

Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,Link,Rc_mobuser,B_name,Place_branch,B_code,

Bank_webinfo,

Auth_info,As_info,As_name,As_id :text,

Hash :hash_func,

SND_B,RCV_Mu,RCV_B,RCV_As,SND_Mu :channel(dy))

played_by As def= local State :nat,

NMu1,Alpha,Beta,NAs1,Ts :text

const mobuser,bank,authser,nas1,ts1,mobuser_authser_nas1,authser_bank_ts1,

auth_info,ts :protocol_id init State :=0 transition

3. State =0 ∧ RCV_As(start) = |>

State' :=2 ∧ Bank_webinfo':=(Url,Link)

∧ Auth_info':=(B_name,Place_branch,B_code)

∧ Mu_id' :=(Mu_ph)

∧ Rc_mobuser' :=(Mu_usr,Mu_pwd)

∧ RCV_As(Bank_webinfo'.Auth_info')

∧ secret(Bank_webinfo',bank_webinfo,B,As)

5. State =2 ∧ RCV_As(Mu_id'.Rc_mobuser') = |>

State' :=3 ∧ NAs1' :=new()

∧ As_info' :=(As_name,As_id)

∧ secret(Rc_mobuser',rc_mobuser,Mu,As)

6. State =3 ∧ SND_Mu(NAs1'.As_info') = |>

State' :=5 ∧ Auth_info':=(B_name,Place_branch,B_code)

∧ Alpha' :=Hash(NAs1'.Auth_info')

∧ secret(NAs1',nas1,Mu,As)

```
9. State =5    ∧ RCV_As(Alpha'.Auth_info'_inv(Ku)) = |>
State' :=7     ∧ Ts' :=new()
                ∧ Mu_id' :=(Mu_ph)
                ∧ Beta':=Hash(Mu_id',Auth_info',Ts')
                ∧ secret(Auth_info',auth_info,Mu,As)
10. State =7    ∧ SND_B(Beta'.Auth_info'.Ts'_inv(Kas)) = |>
State' :=9     ∧ secret(Auth_info',auth_info,B,As)
                ∧ secret(Ts',ts,B,As)
end role
**************************************************************
```

Figure A.16: Authentication server HLPSL code for secure against phishing

In addition to basic roles implemented in HLPSL language, composition roles are also defined. These composition roles are used to combine several basic roles i.e. the roles that are executed in parallel without affecting the execution of the protocol steps and it defines the actual protocol sessions. The session initiates the single specimen of every role and thus defines one protocol run. In our implementation session role, as depicted in Figure A.17, all the primary roles: mobile user, bank, authentication server are instanced with concrete arguments. We gave a top-level role in HLPSL, called here as an environment as depicted in Figure. A.17. In this role, the stated cryptography goals are fulfilled for a bounded number of sessions, and they contain global variables specified in environment role. The intruder (i) also exists in the protocol execution as a particular session. Finally, we define goals of the protocol as depicted in Figure A.17, the goals section HLPSL language represents the goals. The security analysis is carried out against the mentioned cryptography goals and results indicate whether the protocol satisfies the goals or not. The goals section just describes which combinations of these goal facts indicate an attack [33].

```
**************************************************************
role session (Mu,B,As :agent,
Ku,Kas :public_key,
Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,Link,Rc_mobuser,
B_name,Place_branch,
```

```
B_code,Bank_webinfo,Auth_info,As_info,As_name,As_id :text,
Hash :hash_func)
def=
local SND_B,RCV_Mu,RCV_B,SND_As,SND_Mu: channel(dy)
composition
mobuser(Mu,B,As,Ku,Kas,Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,
Link,Rc_mobuser,B_name,
Place_branch,B_code,Bank_webinfo,Auth_info,As_info,As_name,
As_id,Hash,SND_B,RCV_Mu,RCV_B,SND_As)
∧
bank(Mu,B,As,Ku,Kas,Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,Link,B_name,
Place_branch,B_code,
Bank_webinfo,Auth_info,As_info,As_name,As_id,Hash,SND_B,
RCV_Mu,RCV_B,SND_As)
∧
authser(Mu,B,As,Ku,Kas,Mu_id,Mu_ph,Mu_usr,Mu_pwd,Url,
Link,Rc_mobuser,B_name,
Place_branch,B_code,Bank_webinfo,Auth_info,
As_info,As_name,As_id,Hash,
SND_B,RCV_Mu,RCV_B,SND_As,SND_Mu)
end role
role environment()
def=
const
mu,ubank,authserver :agent,
ku,kas :public_key, mu_id,mu_ph,mu_usr,mum_pwd,url,link,rc_mobuser1,b_name,
place_branch,b_code,bank_webinfo1,auth_info1,as_info,
as_name,as_id :text,
h1 : hash_func
intruder_knowledge=mu,ubank,authserver,i
composition
session(mu,ubank,authserver,ku,kas,mu_id,mu_ph,mu_usr,mum_pwd,
url,link,rc_mobuser1,b_name,place_branch,b_code,bank_webinfo1,
```

auth_info1,as_info,as_name,as_id,h1)
∧ session(mu,ubank,authserver,ku,kas,mu_id,mu_ph,mu_usr,mum_pwd,
url,link,rc_mobuser1,b_name,place_branch,b_code,bank_webinfo1,
auth_info1,as_info,as_name,as_id,h1)
end role
goal
secrecy_of nas1
authentication_on bank_authser_ts,bank_authser_auth_info
weak_authentication_on auth_info
end goal
****

Figure A.17: Session HLPSL code's for secure against phishing

Appendix B

Elliptic Curve Cryptography

B.1 Introduction

In 1985, Victor Miller and Neal Koblitz individually suggested the use of points of an elliptic curve over a finite field for cryptosystems. Various standards have approved the use of elliptic curve cryptosystems. In an elliptic curve, cryptography is based on a harder mathematical problem than other cryptosystems. For this reason, ECC has certain advantages over cryptosystem scheme. The elliptic curve equation is defined as $y^2 = x^3 + ax + b$, where $4a^3 + 27b^20$ and value of the a and b generates a different curve. Points (x,y) satisfies the above equation of the elliptic curve. The public key is an elliptic curve point, and the private key is a random integer. The generator point G, the coefficient a and b are domain parameters of ECC. The public key is generated by multiplying the private key with the point G. The generator point in the elliptic curve.

B.2 Elliptic Curves Over Finite Fields

Elliptic curve cryptography is based on elliptic curves which contain points. These points are used to generate keys. The points coordinates must be in the finite field for positive numbers. The field prevent the problem of infinitely repeating fractions and round-off errors. The different fields that elliptic curves defined in, GF(p) prime fields and GF(2^k) binary polynomial fields are used in cryptographic implementations.

B.2.1 Field overview of GF (2^k)

The characteristic-two finite field, F(2^k), referred to as a binary field, contains (2^k) elements for the degree k of the field. Bit strings of length k make up the elements of the field. All arithmetic in F(2^k) is done concerning operations on the bits. A basis, either polynomial or normal, is chosen to interpret the bit strings. The correct arithmetic operations are then chosen depending on which basis is used. The elliptic curve group over GF(2^k) consists of the point at infinity ∞, and all points (x,y), where x,y $\in$ F(2^k), that satisfy the following equation of the form $y^2 + xy \models x^3 + ax^2 + b$, where b $\neq 0$ and a,b $\in$ F(2^k).

B.2.2 Field overview of GF (P)

The finite field F(p), referred to as a prime field, consists of integers in the interval [1,p-1] where p is a prime number. The arithmetic of this field is modulo p so that any calculation results will fall into the finite space defined. An elliptic curve over Galois field GF(p) is defined by the following equation:
$y^2 = x^3 + ax + b$
In order to form a group on the elliptic curve consisting of integers, $x^3 + ax + b$ can contain no repeated factors. This can be guaranteed by ensuring that $(4a^3 + 27b^2)$ mod p $\neq 0$. All points (x,y) that satisfy the equation $y^2 = x^3 + ax + b$ where x,y $\in$ GF(p) , together with an extra point ∞, create a group on the curve.

B.3 Domain Parameters

Typically, domain parameters are shared by a group of entities; however, in some applications, they may be specific to each user Domain parameters D=(q,FR,S,a,b,P,n,h)are comprised of:

1. The field order q.

2. An indication field representation (FR) of the representation used for the elements of F_q.

3. A seed S if the elliptic curve was randomly generated in accordance with standard curve.

4. Two coefficients a,b $\in F_q$ that define the equation of the elliptic curve E over F_q (i.e., $y^2 = x^3 + ax + b$ in the case of a prime field or an OEF, and $y^2 + xy = x^3 + ax + 2$ bin the case of a binary field).

5. Two field elements x_P and y_P in F_q that define a finite point P=(x_P, y_P) $\in$ E(F_q) in affine coordinates. P has prime order and is called the base point.

6. The order n of P.

7. The cofactor h = #E(F_q)/n

B.4 Elliptic Curve Digital Signature Algorithm (ECDSA)

The elliptic curve digital signature algorithm is the elliptic curve similar of DSA which serves the same purposes of key generation, signature generation, and signature verification. ECDSA got accepted in 1998 as an ISO standard (ISO 14888-3), as an ANSI standard (ANSI X9.62) in 1999, and as an IEEE standard (IEEE 1363-2000) and as a NIST standard (FIPS 186-2) in 2000.

B.4.1 Security of ECDSA

The generation of the public key in ECDSA involves computing the point, Q, where Q = dP. To solve the elliptic curve key, the adversary would have to discover the secret key d. Given that the order of the curve E is a prime number n, then computing d given dP and P would take roughly $2_{n/2}$ operations. For example, if the key length n is 192 bits (the smallest key size that NIST recommends for curves defined over GF(p)), then Eve will be required to compute about 296 operations. If Eve had a supercomputer and could perform one billion operations per second, it would take her around two and a half trillion years to find the secret key. This is the elliptic curve discrete logarithm problem behind ECDSA.

B.4.2 Key pair generation of ECC

1. Select an elliptic curve E defined over a finite field F_p such that the number of points in E(Fp) is divisible by a large prime n.
2. Select a base point P, of order n such that P $\in (F_q)$
3. Select a unique and unpredictable integer d, in the interval [1, n-1]
4. Compute Q = dP
5. d is private key.
6. Q is public key .

B.4.3 Signature generation using ECDSA

Using A's private key, A generates the signature for message M using the following steps:

1. Select a unique and unpredictable integer k in the interval [1, n-1]
2. Compute kP = (x1; y1), where x1 is an integer
3. Compute r = x_1 mod n; If r = 0, then go to step 1
4. Compute h = H(M), where H is the Secure Hash Algorithm (SHA-1)
5. Compute s = (k^{-1} h + dr) mod n; If s = 0, then go to step 1
6. The signature of A for message M is the integer pair (r,s)

B.4.4 Signature verification using ECDSA

The receiver B can verify the authenticity of A's signature (r,s) for message M by performing the following: Obtain signatory A's public key (E, P, n, Q)

1. Verify that values r and s are in the interval [1, n-1]
2. Compute w = s^{-1} mod p

3. Compute h = H(M), where H is the same secure hash algorithm used by A
4. Compute u_1 = hw mod n
5. Compute u_2 = rw mod n
6. Compute u_1P + u_2Q = (x_0, y_0)
7. Compute v = x_0 mod n

B.5 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Based on the in-feasibility of computing discrete logarithm on elliptic curves over finite fields, the elliptic curve discrete logarithm problem is the security behind elliptic curve cryptography. Cryptosystems based on the computational complexity of this mathematical problem include ECDSA, EC El-Gamal, and elliptic curve Diffie–Hellman (ECDH). In the ECDLP, however, the subgroup Zp^* is replaced by the group of points on an elliptic curve over a finite field. In addition, unlike the DLP and the integer factorization problem, no subexponential-time algorithm is known for the ECDLP. ECDLP is considered to be significantly harder than the DLP, thus giving elliptic curve cryptosystems a greater strength-per-key-bit than their non-analogue discrete logarithm counterparts.

B.6 Elliptic Curve Integrated Encryption Scheme (ECIES)

The elliptic curve integrated encryption scheme (ECIES) was proposed by Bellare and Rogaway and is a variant of the ElGamal public-key encryption scheme. ECIES is a public-key encryption scheme based on ECC. It is designed to be semantically secure in the presence of an adversary capable of launching chosen-plaintext and chosen-ciphertext attacks.

B.6.1 Scheme Setup

Entities U and V should perform the following setup procedure to prepare to use ECIES:

1. Entity V should establish which of the key derivation functions. Let KDF denote the key derivation function chosen.

2. Entity V should establish which of the MAC schemes supported to use, and select any options involved in the operation of the MAC scheme. Let MAC denote the MAC scheme chosen, mackeylen denote the length in octets of the keys used by MAC, and maclen denote the length in octets of tags produced by MAC .

3. Entity V should establish which of the symmetric encryption schemes supported to use, and select any options involved in the operation of the encryption scheme. Let ENC denote the encryption scheme chosen, and enckeylen denote the length in octets of the keys used by ENC.

4. Entity V should establish whether to use the elliptic curve Diffie-Hellman primitive, or the elliptic curve cofactor Diffie-Hellman primitive.

5. Entity V should establish EC domain parameters T = (p, a, b, G, n, h) or (m, f(x), a, b, G, n, h) at the desired security level. The elliptic curve domain parameters T should be generated using the primitive specified in Section B.3. Entity V should receive an assurance that the elliptic curve domain parameters T are valid using one of the methods specified in B.3.

6. Entity U should obtain in an authentic manner the selections made by V — the key derivation function KDF, the MAC scheme MAC , the symmetric encryption scheme ENC, the elliptic curve domain parameters T, and an indication whether to use the elliptic curve DiffieHellman primitive or the cofactor Diffie-Hellman. Entity U should also receive an assurance that the elliptic curve domain parameters T are valid using one of the methods.

7. Entity U or V should establish whether or not to represent elliptic curve points using point compression.

8. Entities U and V should establish an expected format of $SharedInfo_2$ such that $EM \,||\, SharedInfo_2$ can be uniquely parsed, which may be done if $SharedInfo_2$ is suffix-free, meaning that two different validly formatted values of $SharedInfo_2$ cannot be such that one is a suffix (i.e. tail) of the other. For example, this would be generally the case if $SharedInfo_2$ ended in a counter giving its length.

9. If the XOR symmetric encryption option is selected, then entities U and V should establish whether the backwards compatibility mode (that is, compatibility with version 1.0 of this standard) is desired.

B.6.2 Key Deployment

Entities U and V should perform the following key deployment procedure to prepare to use ECIES:

1. Entity V should establish an elliptic curve key pair (dV , QV) associated with the elliptic curve domain parameters T established during the setup procedure. The key pair should be generated using the primitives.

2. Entity U should obtain in an authentic manner the elliptic curve public key QV selected by V . If the elliptic curve Diffie-Hellman primitive is being used, U should receive an assurance that QV is valid using one of the methods, and if the elliptic curve cofactor Diffie-Hellman primitive is being used, U should receive an assurance that QV is at least partially valid.

B.6.3 Encryption Operation

Entity U should encrypt messages using ECIES using the keys and parameters established during the setup procedure and the key deployment procedure as follows:
Input: The input to the encryption operation is:

- An octet string M which is the message to be encrypted.

- (Optional) Two octet strings $SharedInfo_1$ and $SharedInfo_2$ which consist of some data shared by U and V .

Output: An octet string C which is an encryption of M, or "invalid".
Actions: Encrypt M as follows:

1. Select an ephemeral elliptic curve key pair (k, R) with $R = (x_R, y_R)$ associated with the elliptic curve domain parameters T established during the setup procedure. Generate the key pair using the key pair generation primitive.

2. Decide whether or not to represent R using point compression according to the convention established during the setup procedure. Convert R to an octet string $\bar{R}$ using the conversion routine.

3. Decide whether to use the elliptic curve Diffie-Hellman primitive or the elliptic curve cofactor Diffie-Hellman primitive according to the convention established during the setup procedure. Use the chosen Diffie-Hellman primitive to derive a shared secret field element $z \in \mathbb{F}_q$ from the ephemeral secret key k and V's public key Q_V obtained during the key deployment procedure. If the Diffie-Hellman primitive outputs "invalid", output "invalid" and stop.

4. Convert $z \in \mathbb{F}_q$ to an octet string Z using the conversion routine.

5. Use the key derivation function KDF established during the setup procedure to generate keying data K of length *enckeylen* + *mackeylen* octets from Z and $[sharedinfo_1]$. If the key derivation function outputs "invalid", output "invalid" and stop.

6. Parse the leftmost *enckeylen* octets of K as an encryption key EK and the rightmost *mackeylen* octets of K as a MAC key MK. If symmetric encryption method is XOR and backwards compatibility mode is not selected, then instead parse the rightmost *enckeylen* octets of K as an encryption key EK and the leftmost *mackeylen* octets of K as a MAC key MK.

7. Use the encryption operation of the symmetric encryption scheme ENC established during the setup procedure to encrypt M under EK as cipher-text EM. If the encryption scheme outputs "invalid", output "invalid" and stop.

8. Use the tagging operation of the MAC scheme MAC established during the setup procedure to compute the tag D on $EM \,||\, [sharedinfo_2]$ under MK. If the MAC scheme outputs "invalid", output "invalid" and stop.

9. Output $C = (\bar{R}, EM, D)$. Optionally, the cipher-text may be output as $C = (\bar{R} \,||\, EM \,||\, D)$.

B.6.4 Decryption Operation

Entity V should decrypt ciphertext using the keys and parameters established during the setup procedure and the key deployment procedure as follows:
Input: The input to the decryption operation is:

- A triple of octet strings C = (R, EM, D) or an octet string C, which is the cipher-text.

- (Optional) Two octet strings $SharedInfo_1$ and $SharedInfo_2$ which consist of some data shared by U and V .

Output: An octet string M which is the decryption of C, or "invalid".
Actions: Decrypt C as follows:

1. If C is an octet string and the leftmost octet of C is 02_{16} or 03_{16}, parse the leftmost $\lceil (log_2\ q)\ /8 \rceil + 1$ octets of C as an octet string $\bar{R}$, the rightmost *maclen* octets of C as an octet string D, and the remaining octets of C as an octet string EM. If the leftmost octet of C is 04_{16}, parse the leftmost $2\lceil (log_2\ q)\ /8 \rceil + 1$ octets of C as an octet string $\bar{R}$, the rightmost *maclen* octets of C as an octet string D, and the remaining octets of C as an octet string EM. If the leftmost octet of C is not 02_{16}, 03_{16}, or 04_{16}, output "invalid" and stop.

2. Convert the octet string $\bar{R}$ to an elliptic curve point $R = (x_R, y_R)$ associated with the elliptic curve domain parameters T established during the setup procedure using the conversion routine. If the conversion routine outputs "invalid", output "invalid" and stop.

3. If the elliptic curve Diffie-Hellman primitive is being used, receive an assurance that R is a valid elliptic curve public key. If the elliptic curve cofactor Diffie-Hellman primitive is being used, receive an assurance that R is at least a partially valid elliptic curve public key. If an appropriate assurance is not obtained, output "invalid" and stop.

4. Decide whether to use the elliptic curve Diffie-Hellman primitive or the elliptic curve cofactor Diffie-Hellman primitive according to the convention established during the setup procedure. Use the chosen Diffie-Hellman primitive. To derive a shared secret field element $z \in \mathbb{F}_q$ from V's secret key d_v established during the key deployment procedure and the public key R. If the Diffie-Hellman primitive outputs "invalid", output "invalid" and stop.

5. Convert $z \in \mathbb{F}_q$ to an octet string Z using the conversion routine.

6. Use the key derivation function KDF established during the setup procedure to generate keying data K of length *enckeylen* + *mackeylen* octets from Z and $[sharedinfo_1]$. If the key derivation function outputs "invalid", output "invalid" and stop.

7. Parse the leftmost *enckeylen* octets of K as an encryption key EK and the rightmost *mackeylen* octets of K as a MAC key MK, with the following exception: if the symmetric encryption method is XOR and backwards compatibility mode is not selected, then instead parse the rightmost *enckeylen* octets of K as an encryption key EK and the leftmost *mackeylen* octets of K as a MAC key MK.

8. Use the tag checking operation of the MAC scheme MAC established during the setup procedure to check that D is the tag on $EM \,||\, [sharedinfo_2]$ under MK. If the MAC scheme outputs "invalid", output "invalid" and stop.

9. Use the decryption operation of the symmetric encryption scheme ENC established during the setup procedure to decrypt EM under EK as M. If the encryption scheme outputs "invalid", output "invalid" and stop.

10. Output M.

Appendix C

Fact Sheet for Publications

Table C.1: Fact sheet for publication [1]

Title	A secure end-to-end SMS-based mobile banking protocol
Authors	Sriramulu Bojjagani, and V.N.Sastry
Publication	International Journal of Communication Systems, Vol. 30, No. 15, pages 1- 19, 2017
ISBN/ISSN	1099-1131
DOI	10.1002/dac.3302
Status	Published
Publisher	Wiley
Publication Type	Journal

Figure C.1: Publication [1]

Table C.2: Fact sheet for publication [2]

Title	SSMBP: A Secure SMS-based Mobile Banking Protocol with Formal Verification
Authors	Sriramulu Bojjagani, and V.N.Sastry
Publication	Proceedings of the 11^{th} IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. (WiMob'2015) Abu Dhabi, UAE, pages 252-259, Oct 19-21, 2015
ISBN/ISSN	978-1-4673-7701-0
DOI	10.1109/WiMOB.2015.7347969
Status	Published
Publisher	IEEE
Publication Type	Conference

SSMBP: A secure SMS-based mobile banking protocol with formal verification

Sign In or Purchase to View Full Text

2 Paper Citations

176 Full Text Views

Related Articles

Internet suspend/resume

Mobile agents protection in the Internet environment

View All

2 Author(s)

Sriramulu Bojjagani ; V. N. Sastry

View All Authors

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
Emerging M-Commerce (Mobile-Commerce) activity opened scope for the design of new protocols. Now a days SMS (Short Message Service) as a mobile payment channel is gaining prominence because it supports low end devices and low of cost. In this paper we proposed a novel protocol for SMS based mobile banking and payment framework with formal verification. The framework of A Secure SMS-based Mobile Banking Protocol with Formal Verification (SSMBP) involves authentication and payment. This model provides secure SMS communication between the customer and the bank through a mobile phone banking application. Further, its verified that the protocol for its correctness and security properties because most of the protocols lack of achieving them. Our proposed framework is experimentally validated by a formal method of model checking tool called Automated Validation of Internet Security Protocols and Applications (AVISPA) and Burrows Abadi and Needham logic (BAN). Security analyses shows that, the proposed mechanism works better, compared with existing SMS payment protocols for real-world applications.

Published in: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)

Date of Conference: 19-21 Oct. 2015

INSPEC Accession Number: 15651220

Date Added to IEEE *Xplore*: 07 December 2015

DOI: 10.1109/WiMOB.2015.7347969

ISBN Information:

Publisher: IEEE

Conference Location: Abu Dhabi, United Arab Emirates

Figure C.2: Publication [2]

Table C.3: Fact sheet for publication [3]

Title	STAMBA: Security Testing for Android Mobile Banking Apps
Authors	Sriramulu Bojjagani, and V.N.Sastry
Publication	In Proceedings of the 2^{nd} International Symposium on Signal Processing and Intelligent Recognition Systems (SIRS'2015) Trivandrum, India, pages 671-683, December 16-19, 2015
ISBN/ISSN	978-3-319-28658-7
DOI	https://doi.org/10.1007/978-3-319-28658-7_57
Status	Published (Available online)
Publisher	Springer
Publication Type	Conference

Figure C.3: Publication [3]

Table C.4: Fact sheet for publication [4]

Title	VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps
Authors	Sriramulu Bojjagani, and V.N.Sastry
Publication	In Proceedings of the 3^{rd} IEEE International Conference on Collaboration and Internet Computing (CIC'2017) San Jose, California, USA, pages 77-86, Oct 15-17, 2017
ISBN/ISSN	978-1-5386-2565-1
DOI	10.1109/CIC.2017.00022
Status	Published
Publisher	IEEE
Publication Type	Conference

VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps

Sign In or Purchase to View Full Text

218 Full Text Views

Related Articles

Abusing mobile devices for Denial of Service attacks

A fuzzing test for dynamic vulnerability detection on Android Binder mechanism

View All

2 Author(s) | Sriramulu Bojjagani ; V.N. Sastry | View All Authors

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
Mobile devices are becoming targets for hackers and malicious users due to the multifold increase in its capabilities and usage. Security threats are more prominent in mobile payment and mobile banking applications (MBAs). As these MBAs, store, transmit and access sensitive and confidential information, so utmost priority should be given to secure MBAs. In this paper, we have analyzed MBAs of several banks running on two dominant platforms of Android & iOS using both static and dynamic analysis. We have proposed threat model, to detect various vulnerabilities rigorously. We have done a systematic investigation of different unknown vulnerabilities particularly in mobile banking applications and showed how MBAs are vulnerable to MitM attacks. We observe that some MBAs are using simple HTTP protocol to transfer user data without concerning about security requirements. In Most of the cases, MBAs are receiving the fake or self-signed certificates. These are blindly maintaining all certificates as sound and valid, which leads to SSL/TLS Man-in-the-Middle (MitM) attacks. We present a detailed analysis of the security of MBAs which will be useful for application developers, security testers, researchers, bankers and bank customers.

Published in: 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)

Date of Conference: 15-17 Oct. 2017

INSPEC Accession Number: 17431195

Date Added to IEEE *Xplore*: 14 December 2017

DOI: 10.1109/CIC.2017.00022

ISBN Information:

Publisher: IEEE

Conference Location: San Jose, CA, USA

Figure C.4: Publication [4]

Appendix D

Similarity Report