DESIGN OF PROTOCOLS FOR SECURE MOBILE PAYMENTS AND THEIR FORMAL VERIFICATION

A thesis submitted to the University of Hyderabad in partial fulfillment of the requirements for the award of the Ph.D. Degree in Computer Science

by
Shaik Shakeel Ahamad (07MCPC09)

School of Computer and Information Sciences

University of Hyderabad,
(P.O.) Central University, Gachibowli
Hyderabad – 500 046
Andhra Pradesh

INDIA

MAY 2013

School of Computer and Information Sciences (SCIS)
University of Hyderabad, Hyderabad - 500 046, INDIA

CERTIFICATE

This is to certify that the thesis entitled "**Design of Protocols for Secure Mobile Payments and their Formal Verification**" submitted by **Mr. Shaik Shakeel Ahamad** bearing **Reg. No 07MCPC09** in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in **Computer Science** is a record of bonafide research work carried out by him at IDRBT under our supervision and guidance. The thesis has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Prof. V.N. Sastry (IDRBT)	**Dr. Siba K. Udgata (UoH)**
Supervisor	Supervisor

Prof. Arun K. Pujari
Dean, School of Computer and Information Sciences (SCIS)
University of Hyderabad, Hyderabad - 500046, INDIA

DECLARATION

I, Shaik Shakeel Ahamad hereby declare that this thesis entitled "Design of Protocols for Secure Mobile Payments and their Formal Verification" submitted by me under the guidance and supervision of Prof. V.N. Sastry (IDRBT) and Dr. Siba K. Udgata (UoH) is a bonafide research work. 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.

Date: Shaik Shakeel Ahamad

Signature of the Student

Regd. No. 07MCPC09

ABSTRACT

The unprecedented growth of mobile communication technology stimulated by the ever increasing demand for personal mobility in communications has lead researchers to develop new technologies such as Near Field Communication (NFC) Technology and Universal Integrated Circuit Cards (UICC). This thesis proposes protocols which improve the currently prevalent mobile payment protocols as wireless technology is extremely complex, compared with wired networks. In order to fully deploy m-commerce for business, two levels of security requirements must be satisfied. The lower level requirement is the need for a secure wireless infrastructure to protect each individual wireless communication and the higher level requirement is for a secure protocol with which to conduct mobile payment and business transactions, thus protecting the legitimate security concerns of all the parties involved namely the Client, Merchant, Issuer, Acquirer and Payment Gateway. Although for lower level security requirement, wireless communication security, is the topic of considerable ongoing research and is vital for the deployment of all wireless applications, this thesis has focused on the higher level security, mobile payment and transaction security. We observe that existing security protocols at the communication layer in mobile environment are not adequate. Therefore, security at the application layer must be added in order to achieve end-to-end protection for mobile financial services. This thesis proposes protocols for personalizing the Secure Element (such as Universal Integrated Circuit Card (UICC)) by the client and mobile payment applications (which are on UICC) by the bank through OTA (Over The Air). We propose Mobile Payment Protocols by adopting Wireless PKI, using UICC as secure element, Mobile Agent technology, Signcryption and DSMR (Digital Signature with Message Recovery) mechanisms which helps in achieving end to end security thereby achieving all the security properties. Mobile Payment protocols proposed in this thesis consume fewer resources by reducing communication cost and computational cost. All the proposed Mobile Payment Protocols are successfully verified using Manual Formal verification methods (such as BAN Logic) and using Automated Methods (such as AVISPA and Scyther Tools).

ACKNOWLEDGEMENTS

This thesis is the result of research work carried out at IDRBT and University of Hyderabad. I would like to thank Allah, Prophet Mohammad (S.A.W), KGN, Kaliyar Sharif, KBN, Yousuffain Baba, Sharif Baba and Baji Baba for their blessings showered on me. I would like to thank my supervisors Professor V.N. Sastry (IDRBT) and Dr. Siba K. Udgata (UoH) (who has always cheered me up) who introduced me into the exciting field of Security in Mobile Payments research with the focus on improving real time security in mobile commerce. I gratefully acknowledge their prompt, insightful, clear, and frank feedback; their foresight on identifying values, shortcomings, and obstacles in a particular proposal; their aptitude for putting an idea into context; their extra-ordinary insight for details; for always being positive and reminding me of the bright side of life especially in difficult times; and above everything, for being patient and helping me to make the long process of this work a meaningful and worthwhile endeavor. I would also like to thank Professor C.R. Rao and Dr. Rajeev Wankar for reviewing me in DRC meetings in spite of their busy schedule and for their valuable feedback (which improved this thesis) and for their constructive suggestions. I wish to thank the Dean, School of Computer and Information Sciences, Prof. Arun K. Pujari for his cooperation. I would like to thank Prof. Arun Agarwal (Former HOD, DCIS), Prof. C. Raghavendra Rao (Former HOD, DCIS), Prof. P.N. Girija (Former HOD, DCIS) and Mr. Sambamurthy (Director IDRBT) for their constant support throughout my research tenure. I would like to acknowledge the insightful feedback, and constructive suggestions from anonymous reviewers of Journals and Conferences of works in this thesis. I have received substantial financial support from IDRBT and I like to thank IDRBT for easing the financial burden while doing this research, and I would like to thank SCIS, UoH for providing me this opportunity. Being the spouse of a Research Scholar is not easy. I would like to thank my wife Fatima and my daughter Shaik Sheena Ahmed for coping with my unpredictable schedules; for understanding the challenges in a Research Scholar's life. And last, but not the least, my special thanks to all my colleagues (Mohd.Farquad Haque, N. Naveen and Illiah) at IDRBT and (Sridhar Bhukya, Rushdie Umar) at University of Hyderabad for numerous helpful talks and discussions in corridors and at lunch tables.

TABLE OF CONTENTS

List of Tables

List of Figures

Abbreviations

Abbreviation	Description
3G	3rd Generation Mobile Network
3GPP	Third Generation Partnership Project
AES	Advanced Encryption Standard
AI	Account Information
Amt	Amount
APDU	Application Protocol Data Unit
API	Application Programming Interface
AS	Assumptions
ASN.1	Abstract Syntax Notation
AuthOI	Authorization of OI
AuthPI	Authorization of PI
AVISPA	Automated Validation of Internet Security Protocols and Applications
BAN Logic	Burrows, Abadi and Needham Logic
BIP	Bearer Independent Protocol
CA	Certifying Authority
CDMA	Code Division Multiple Access
CL-AtSe	Constrained-Logic-based Attack Searcher
CMP	Certificate Management Protocol

CRL	Certificate Revocation List
DES	Data Encryption Standard
DIC	Digital Invoice Certificate
DSMR	Digital Signature with Message Recovery
EAC	Enrolment Activation Code
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read Only Memory
EMSET	Enhanced Mobile Secure Electronic Transaction
EMV	Europay-Mastercard-Visa
ENDOR	Endorsement
ETSI	European Telecommunications Standards Institute
ETSI SCP	European Telecommunications Standards Institute Smart Card Platform
FV	Face Value
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
GSMA	Global System for Mobile communications Alliance
HLPSL	High Level Protocol Specification Language
HMAC	Hashed Message Authentication Code
ICCID	Integrated Circuit Card Identity
Iden	Identity

IF	Intermediate Format
In	Intruder
INAg	Information & Negotiation Agent
LI	Location Information
MA	Mobile Agent
MAC	Message Authentication Code
MANET	Mobile Ad hoc Networks
MD5	Message Digest 5
MIDP	Mobile Information Device Profile
MNO	Mobile Network Operator
MNSP	Mobile Network Service Provider
MPA	Mobile Payment Application
MPPU	Mobile Payment Protocol based on UICC
MQ	Merchant Quote
MS	Message
MS ISDN	Mobile Station International Subscriber Directory Number
MTC	Mobile Traveler’s Check
MWMA	Mobile Wallet Manager Application
NFC	Near Field Communication
NFC CLF	Near Field Communication Contactless Front-end
NFC WI	Near Field Communication Wired Interface
NR	Not Referred

NRL	Naval Research laboratory
NRP	Non Repudiation PIN
OCSP	Open Certificate Status Protocol
OFMC	On the Fly Model Checker
OTA	Over The Air provisioning
PAg	Payment Agent
PC	Proxy Certificate
PDA	Personal Digital Assistant
PG	Payment Gateway
PI	Payment Information
PIN	Personal Identification Number
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PO	Purchase Order
POS	Point-of-Sale Device
PR	Payment Requirement
PSE	Personal Security Environment
PUK	PIN Unblocking Key
RA	Registration Authority
RFID	Radio Frequency Identification
RSA	Rivest, Shamir and Adleman Algorithm
SAG	Secure Access Gateway

SAT	Satisfiability
SATMC	SAT- based Model Checker
SE	Secure Element
SET	Secure Electronic Transaction
SFS	Signcryption scheme with Forward Secrecy
SHA	Secure Hash Algorithm
SID	Subject Identifier
SIM	Subscriber Identity Module
SMC	Secure Memory Card
SMS	Short Message Service
SMWP	Secure Mobile Wallet Protocol
SNO	Serial Number
SOMMP	Secure and Optimized Merchant based Mobile Payment
SOPMP	Secure and Optimized Proximity based Mobile Payment
SP	Service Provider
SPAN	Security Protocol Analyzer
SPDL	Security Protocol Definition language
SSCD	Secure Signature Creation Device
SSL	Secure Socket Layer
SWP	Single Wire Protocol
TA4SP	Tree Automata-based Protocol Analyzer
TCP	Transmission Control Protocol
TELCO	Telecommunication Company

TID	Transaction Identity
TLS	Transport Layer Security
TPE	Trusted Processing Environment
TS	Time Stamps
TTP	Trusted Third Party
UAS	User Authentication Server
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
URL	Universal Resource Locator
USSD	Unstructured Supplementary Services Data
WIM	Wireless Identity Module
WML	Wireless Markup Language
WMA	Wireless Messaging API
WPKI	Wireless Public Key Infrastructure
WSLC	WPKI Short Lived Certificate
X.509 SLC	X.509 Short Lived Certificate
XHTML	eXtensible Hyper Text Markup Language

Notations

Notation	Description
A	Acquirer
$Anonid_X$	Anonymous Identity of Client of entity ‘X’
Amt_X	Amount generated by entity ‘X’
$AuthOI_X$	Authorization of OI generated by entity ‘X’
$AuthPI_X$	Authorization of PI generated by entity ‘X’
C	Client
$Cert_X$	Certificate of entity ‘X’
DIC_X	Digital Invoice Certificate generated by entity ‘X’
$DSMR_{X_Y}(MS)$	DSMR of message MS generated by entity ‘X’ intended to be verified by entity ‘Y’
$DS_Y^X(MS)$	Digital Signature Message Recovery generated by entity ‘X’ intended to be verified by entity ‘Y’ on message ‘MS’
$ENDOR_X$	Endorsement of entity ‘X’
HMQ_X	Hashed Merchant Quote of entity ‘X’
HOI_X	Hashed OI generated by entity ‘X’
HPI_X	Hashed PI generated by entity ‘X’
I	Issuer
K_{CI}	Symmetric Key Shared between Client (C) and Issuer (I)
K_X	Public Key of entity ‘X’
$K_X^{'}$	Private Key of entity ‘X’
LI_X	Location Information of entity ‘X’
M	Merchant

$MAC(X)$	Message Authentication Code of ‘X’
$MobPayAppl$	Mobile Payment Application
MQ	Merchant Quote
N_X	Nonce generated by entity ‘X’
$PayeeID$	Payee Identity
PG	Payment Gateway
$PhNo$	Mobile Phone Number
$Pubkey_X$	Public Key Parameters of entity ‘X’ in DSMR
R_X	Random Number generated by entity ‘X’
$SIG_{X_Y}(MS)$	Signcryption of message MS generated by entity ‘X’ intended to be verified by entity ‘Y’
T_X	Timestamp of entity ‘X’
$T_X^{'}$	Validity Period of Certificate
TID_X	Transaction Identity generated by entity ‘X’
$TranCertC$	Transaction Certificate of Client
$T_{\exp}$	Time for Exponentiation with Modulo N
T_{mm}	Time for Multiplication with Modulo N
T_m	Time for Multiplication without Modulo N
T_h	Time for calculating the adopted one way function ‘h’
T_{XOR}	Time for performing XOR operation
T_{SYMM}	Time for Encryption/Decryption using DES
$T_{sign(gen)}$	Time for generating Digital Signature using ECDSA
$T_{sign(verf)}$	Time for verifying Digital Signature using ECDSA
T_{ECPM}	Time for ECPM (Elliptic Curve Point Multiplication)
UC	UICC

$X.509SLC$	X.509 Short Lived Certificate
$X \rightarrow_{(Ag)} Y : MS$	Entity ‘X’ generates a Mobile Agent and sends message MS to Entity ‘Y’ with Mobile Agent (Ag)

Chapter 1: Introduction

1.1. Overview

Mobile payments are defined as wireless transactions of monetary value from one party to another where a mobile device (e.g., mobile phone, PDA (Personal Digital Assistant), smart phone or any mobile payment terminal) is used in order to initialize, activate and/or confirm the payment (Karnouskos,2004). An alternative definition for an m-payment is that it is a type of electronic payment transaction procedure in which at least the payer employs mobile communication techniques in conjunction with mobile devices for the initiation, authorization or realization of payment (Turowski and Pousttchi, 2004). Mobile payments provide consumers with ubiquitous payment possibilities, timely access to financial assets and an alternative to cash payments. The advantage of mobile payments compared to other means is banking anywhere and anytime. Mobile Payment is an integral part of M-Commerce. M-Commerce refers to transactions using a wireless mobile device resulting in the transfer of value in exchange for information, services, or goods and includes services such as banking, payment, and ticketing. Due to its inherent characteristics such as ubiquity, reachability, flexibility, and localization, mobile commerce promises high business potential, greater productivity and higher profitability. On account of personalized service and mass adoption there is need for novel approaches to evaluate Mobile Commerce applications and its security. Important channels of wireless communication useful in mobile commerce are: (1) Short Message Service (SMS), (2) Unstructured Supplementary Services Data (USSD), (3) General Packet Radio Service (GPRS), (4) Near Field Communications (NFC) and (5) Mobile Agents (MA). Existing security solutions do not necessarily ensure end- to- end security. For guaranteeing end- to- end security in mobile transaction, we need PKI but it cannot be used in the present form because mobile phone has less powerful CPU, less memory size, restricted battery power and small display. In addition to these there are several problems on wireless networks such as less bandwidth, more latency and insecure connection.

So WPKI (Wireless Public Key Infrastructure) is adopted for ensuring end-to-end security in wireless environment. In addition to security, mobile payment protocol should consume fewer resources as mobile devices have less capabilities and bandwidth limitations on wireless communication channels. So Mobile payment protocols should be lightweight (less communication cost and computational cost) and ensure all the security properties. A protocol is a set of rules that define the conventions to be followed to establish semantically correct communication between the participating entities. A security protocol on a communication protocol makes the message exchange secure using the defined Cryptographic mechanisms. Secure Mobile payment protocols should ensure all the security properties such as Integrity, Confidentiality, Authentication and Non Repudiation and achieve Identity protection from merchant and Eavesdropper, achieve Transaction privacy from Eavesdropper and Payment Gateway, achieve Payment Secrecy, Order Secrecy, forward secrecy, and should avoid Double Spending, Overspending and Money laundering. In addition, secure mobile payment protocols should withstand Replay, Man in the Middle, Impersonation and Multiprotocol attacks. Security protocols are error prone and it is not so easy to identify errors and prove their correctness. Merely using cryptographic mechanisms, does not guarantee semantically secure operation of the protocol, even if it is correct. There have been instances of reported breaches in some security protocols, after being published and accepted as a safe protocol. The design of secure mobile payment protocol is an intuitive process which is likely to be error-prone, so a robust protocol is required within which one can safely design secure protocols (Muhammad et al., 2006). Formal verification of security protocols is essential because it detects flaws which lead to protocol failure. Formal methods of verification can be classified as Manual and Automated such as Belief logics (BAN logic (Burrows, M. et al., 1989)), Theorem proving (NRL) and Model Checking (AVISPA (Armando et al., 2005) and Scyther (C. J. F. Cremers, 2006)).

1.1.1. Wireless technologies for Mobile Payments

Mobile payments are classified as (A) Remote Mobile Payments and (B) Proximity Mobile Payments based on the wireless technologies used for communicating. Remote Mobile Payments uses long range wireless technologies such as GSM, GPRS, CDMA and Mobile Agents (MA). Proximity Mobile Payments uses short range wireless technologies such as Bluetooth, RFID and NFC.

A) Remote Mobile Payments:

There are many long range wireless technologies such as GSM, GPRS, CDMA and Mobile Agents. This thesis uses GPRS and Mobile Agent technology for remote mobile payments.

Global System for Mobile Communication (GSM) is a second generation standard for Mobile communication developed by the European Telecommunications Standards Institute (ETSI) with final ownership belonging to the Third Generation Partnership Project (3GPP). GSM is the most widespread mobile standard currently utilized in Europe and the Asia- Pacific region. GSM is limited in its low data transmission speed and low bandwidth of data services. Other limitations include the fact that the charge for using GSM is based on the on-line duration and reconnection is required for each session. These limitations make it difficult to use GSM as the main stream network environment for real-time mobile payment transactions.

General Packet Radio Service (GPRS) is considered as an overlay network on the GSM networks, adding the extra network elements on the GSM infrastructure. It charges only for data received and have much more stable connectivity. GPRS is classified as a 2.5G technology, which means a technology between 2G and 3G. It is an extended service of the GSM network that offers the ability of surfing Internet using a phone at a slightly higher speed than GSM. GPRS Internet surfing speeds range from 9.6 Kbps to 171.2 Kbps. A phone with support capacities and a subscription from a supporting network operator is required to operate with GPRS.

The benefit of higher speed and more stable connectivity makes GPRS more suitable for operating mobile applications in real time.

Mobile Agent (MA) technology advances the distributed computing paradigm one step further to offer two extra properties: client customization and autonomy. End users are now able to virtually install new software in targeted foreign hosts by creating and launching a personalized mobile agent onto the Internet, thereby automatically accomplishing the assigned mission without the need for interactive guidance from the user. A mobile agent acts as a smart software agent that can be executed in foreign hosts on behalf of its owner. It can make decisions autonomously, based on the decision logics it contains. Once it has been launched, it is independent from its owner. During its life, it may visit many foreign hosts, communicate with other agents, and finally return to its owner with the results. On the other hand Mobile Agent technology has many benefits such as bandwidth conservation, reduction of latency, reduction of completion time, Asynchronous (disconnected) communications. Mobile agent overcomes low bandwidth and disrupted network.

B) Proximity Mobile Payments:

There are many short range wireless technologies such as Bluetooth, RFID and NFC. Near field communication (NFC) is a short-range wireless technology which is the advanced development of RFID technology. NFC's fundamental advantages compared to other wireless technologies like Bluetooth is the availability of the data storage facility known as the NFC tag. NFC is not just a replacement data cable as Bluetooth, but also as a means of storage of data. Referring to the NFC Forum, NFC technology is currently used in three areas, namely sharing, pairing, and transaction. NFC operates between two devices over a very short communication range. NFC communication uses the 13.56 MHz spectrum as in RFID. Currently data transfer speed options are 106, 212, and 424 kbps. NFC technology operates in different operating modes; reader/writer, peer-to peer, and card emulation where communication occurs between an NFC mobile on one side, and a passive RFID tag (NFC tag), an NFC mobile or an NFC reader on the other side.

NFC is a bidirectional and short range wireless communication technology. The communication occurs between two NFC compliant devices within a few centimeters. A 13.56 MHz signal with a bandwidth not more than 424 kbps is used. NFC technology operates in different operating modes; reader/writer, peer-to-peer, and card emulation where communication occurs between an NFC mobile on one side, and a passive RFID tag (NFC tag), an NFC mobile or an NFC reader on the other side (Vedat Coskun et al., (2012)). The NFC interface is composed of an analog/digital front-end called NFC Contactless Front-end (NFC CLF), an NFC antenna and an NFC controller to enable NFC communication. The NFC controller enables NFC communication of the mobile phone with the external NFC device. An NFC enabled mobile phone requires a Secure Element (SE) for performing secure transactions with the external NFC devices. The SE provides a secure environment for related programs and data. It enables storage of sensitive data of the user. It also enables secure storage and execution of NFC enabled services such as contactless payments. Various standards have already been defined for NFC communication between two NFC enabled devices, and data transfer within the NFC mobile phone such as Single Wire Protocol (SWP) or the NFC Wired Interface (NFC-WI) (Vedat Coskun et al., (2012)). SWP is a specification for a single wire connection between the SIM card and an NFC chip in the handset standardized by European Telecommunications Standards Institute (ETSI). Following are the motivations for using NFC as Proximity technology for Proximity based Mobile payments

a) NFC technology is better when compared with the existing wireless technologies in terms of Security, Personalization, Flexibility and Power Consumption parameters as given in table 1.1.

b) The main motivation for NFC is the integration of personal and private information such as credit card or debit card data into mobile phones. Therefore, security is the most important concern, and the wireless communication range provided even by RFID technology is considered too long. Mechanisms such as shielding are necessary to prevent unauthorized people from eavesdropping on private information because even no powered, passive tags can be read over 10 m. This is where NFC comes in (Vedat Coskun et al., (2012)).

c) NFC communication between two NFC enabled devices, and data transfer within the NFC mobile phone such as Single Wire Protocol (SWP) or the NFC Wired Interface (NFC-WI). In order to provide secure storage and execution of NFC enabled applications, an SE is essential.

Wireless Technologies / Parameter	RFID	Bluetooth	ZigBee	NFC
Security	High	Low	Low	High
Personalization	High	Medium	Low	High
Flexibility	Low	High	High	High
Power Consumption	No	High	Medium	Low

Table 1.1 Comparison of NFC with other Wireless Technologies Chang, Y. et al. (2010)

d) The SE is actually a combination of hardware, software, interfaces, and protocols. UICC based SE provides an ideal environment for NFC applications because it is personal, secure, portable and easily managed remotely via OTA technology. The card holder can be assured that transactions are executed with their personal information protected. It has appropriate card structure based on the Global Platform card specification that allows multiple security domains for different applications on the same smart card. By using OTA, new NFC applications can remotely be installed onto the UICC easily, personalized afterwards, and the life cycle of the SE can be managed easily thereafter.

e) One of the major properties of NFC technology is its implicit security because of the short communication distance involved. The close proximity of the two devices makes the signal interception probability very low. The other property is the implicit pairing capability of NFC.

f) In the case of a payment, user's intervention is essential and the payment cannot proceed without the user's approval and without details such as a payment password being entered. Automatically firing the NFC application allows automatic pairing of the related NFC components, which is not true in Bluetooth. The user shows her intention for this automatic pairing by bringing her device close to the other device. In order to show intention in a peer-to-peer communication, two users need to bring their devices close to each other. So, pairing two NFC devices is completely different from the processes required in Bluetooth. In addition to this pairing of devices in Bluetooth includes searching, waiting, pairing, and authorization.

1.1.2. Formal Verification of Protocols

A protocol is a set of rules that follows the defined conventions to establish semantically correct communications between the participating entities. A security protocol is an ordinary communication protocol in which the message exchanged is often encrypted using the defined Cryptographic mechanisms. The mechanisms of Symmetric Key Cryptography or Asymmetric Key Cryptography are used to satisfy various security properties such as Confidentiality, Entity Authentication, Message Integrity, Non-repudiation, Message Freshness etc. However, merely using cryptographic mechanisms, does not guarantee security-wise semantically secure operation of the protocol, even if it is correct. There indeed have been reported breaches in the security protocols, after being published and accepted as a safe protocol. The design of security protocol which is an intuitive process is severely error-prone so a more rigid protocol is required within which one can safely design secure protocols. The network is assumed to be hostile as it can contain intruders with the capabilities to encrypt, decrypt, copy, forward, delete, and so forth. Considering an active intruder with such powerful capabilities, it becomes extremely difficult to guarantee proper working of a security protocol. Several examples show how carefully designed protocols were later found to have security breaches

(Muhammad et al., 2006). Formal methods are then necessary for verification of secure mobile payment protocols.

The formal methods of verification can be classified as Belief logics, Theorem proving and Model Checking as shown in figure 1.1.

a) **Belief logics:** These logics are composed by a group of modal operators, a set of believers and a set of inference rules. The operators describe the relationship among the model participants and the data. The beliefs are held by the participants. The inference rules allow one to derive new believes from the previous ones. Belief logics work in a higher abstraction level than other approaches. Thus, they are less precise than the other ones. On the other hand, their main advantage is their simplicity.
They are decidable and offer efficient computation algorithms. In spite of their simplicity, they can point out serious errors in security protocols. The most popular example of these belief logics is BAN logic (Burrows, M. et al., 1989).

b) **Theorem proving:** They are based on an inference process by means of several theorems to prove the protocol properties. They are interactive in nature.

c) **Model Checking:** They follow the standard Dolev-Yao model (where a state space is defined) in which the intruder is assumed to have control of the network. This state space is explored by means of an algorithm to find out a path which leads to a state where the properties are not held. Casper, AVISPA Tool and Scyther are examples in this category.

The first two approaches are focused on proving the correctness of the protocol according to the properties. In contrast, the model checking concentrates on searching incorrect traces. AVISPA tool offers a high level language that is translated into an intermediate format. This intermediate format is the input for four different back ends. This way the analysis can be performed by four different although complementary approaches. Scyther Tool offers a high level language SPDL (Security Protocol Definition Language).

Drawbacks of BAN Logic

a) It is difficult to verify large protocols and is usually error prone.

b) Assumes that all the entities involved are honest.

c) Based upon Assumptions.

d) Not good for finding attacks.

e) Slow and not good for analyzing large/complex protocols

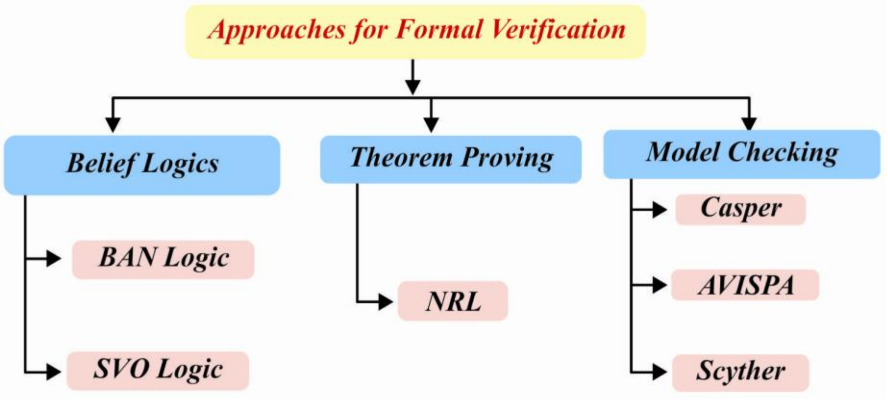


Figure 1.1 Approaches for the Formal Verification of Security Protocols

Need for Automated Tools (Model Checking)

a) The number and scale of new security protocols under development is out-pacing the human ability to rigorously analyze and validate them.

b) To Speed up the development of new security protocols and to improve their security it is important to have tools that support the rigorous analysis of security protocols, by either finding flaws (or) establishing their correctness.

c) Good in finding attacks.

d) Optimally, these tools should be completely Automated, Robust, Expressive and easily usable, so that they can be integrated into the protocol development and Standardisation process.

The successful use of the formal methods for verification has led to the upsurge in devising similar tools for verifying the security properties of a cryptographic protocol, too. In order to gain confidence in the cryptographic protocol employed, it has been found desirable that the protocol be subjected to an exhaustive analysis that verifies its security properties. Some of the tools developed for the purpose are Scyther (C. J. F. Cremers, 2006)) and AVISPA (Armando et al., 2005). These tools differ in their input language and also in the way they verify the protocols and provide the output. Scyther and AVISPA tools eliminate the possibility of human error and help in verifying the correctness of security protocols. Very few automated tools explore all the possible behaviors, whereas others explore strict subsets. Ignoring these kinds of differences will lead to completely wrong interpretations of the output of a tool. Cas Cremers and Pascal Lafourcade's have applied state space relations in performance comparison of several well-known automatic tools for security protocol verification, After the analysis of performances of tools over comparable state spaces, they concluded that the efficiency of the Scyther and ProVerif are superior, their approximation techniques are effective, and both can handle unbounded verification. Scyther tool has the advantage of not using approximations. We notice that some mobile payment protocols available in the literature were verified using only manual verification such as BAN Logic but are not verified with Automated Tools.

AVISPA Tool

AVISPA (Automated Validation of Internet Security Protocol Applications) protocol provides a high-level formal language HLPSL (High-Level Protocol Specification Language) for specifying protocols and their security properties. AVISPA also offers a graphical interface SPAN that helps in specifying task. The language used for writing protocols in AVISPA is HLPSL (High-Level Protocol Specification Language). We have carried out formal validation of our proposed protocols using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. The validation process is as follows.

a) We specify our protocols in the High Level Protocol Specification Language (HLPSL).
b) AVISPA tool translates it into the Intermediate Format (IF) specification.

c) Intermediate Format (IF) specification is analyzed invoking state-of-the-art back-ends that this tool provides, which are currently: On-the-Fly Model Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model Checker (SATMC), and Tree Automata-based Protocol Analyzer (TA4SP).

d) As a result, these back-ends return attacks (if any) in a readable output format.

AVISPA Tool	Scyther Tool
AVISPA assumes that the protocols are used in isolation of other protocols (i.e., there is only a single protocol using a network at a given time)	Scyther assumes that there are multiple protocols that are working in parallel (i.e., there are many protocols using a network at a given time)
It uses HLPSL language	It uses SPDL language
Does not verify Multi Protocol attacks	Good in verifying Multi Protocol attacks
Attack graphs are not generated	Attack graphs are generated when attacks are found in the protocol.

Table 1.2: Differences between AVISPA Tool and Scyther Tool

Scyther Tool

Traditionally, verification methods for security protocols typically assume that the protocols are used in isolation of other protocols (i.e., there is only a single protocol using a network at a given time). However, in practice it is unrealistic to assume that a security protocol runs in isolation in an insecure network. Scyther is a tool used for security protocol verification, where it is assumed that all the cryptographic functions are perfect. The tool provides a graphical user interface that makes it easier to verify and understand a protocol. In addition, attack graphs are generated whenever an attack is found corresponding to the claim mentioned. The tool can also verify all the possible claims on the protocol. The tool can be used to find problems that arise from the way the protocol is constructed. It can also be used to generate all the possible trace patterns. The verification can be done here using a bounded or an unbounded number of sessions. The language used to write protocols in Scyther is SPDL (Security Protocol Description Language) (C. J. F. Cremers, 2006). As per our knowledge we are the first to verify the correctness of Mobile Payment Protocol using Scyther Tool. Scyther is a tool for the verification, falsification and the analysis of security protocols, where it is assumed that all cryptographic functions are perfect. Scyther provides a number of novel features that include the possibility of unbounded verification with guaranteed termination, analysis of infinite sets of traces in terms of patterns and support for multi-protocol analysis. Scyther is based on a pattern refinement algorithm, providing concise representations of (infinite) sets of traces. This allows the tool to assist in the analysis of classes of attacks and possible protocol behaviors, or to prove correctness for an unbounded number of protocol sessions. Scyther verifies bounded and unbounded number of runs, using a symbolic analysis with a backward search based on (partially ordered) patterns. Scyther does not require the input of scenarios. Table 1.2 shows the differences between AVISPA and Scyther tool.

1.2. Objectives of the Thesis

The objectives of this research work are as follows

a) To propose protocols for the Personalization of Universal Integrated Circuit Card (UICC) by the client and personalization (Mutual Authentication & Key Agreement Protocol) of Mobile Payments Application (which is on UICC) by the Bank.

b) To propose mobile wallet solution by overcoming the flaws in the existing mobile wallet protocols.

c) To design mobile payment protocols in UICC by applying cryptographic techniques such as Digital Signature, DSMR (Digital Signature with Message Recovery) and Signcryption based on Elliptic Curve Cryptography (ECC) which offers higher transaction performance than the existing solutions when operating on low end computational mobile devices.

d) To propose a secure payment instrument (MTC (Mobile Traveler's Check)) in the realm of mobile commerce which has the merits of both e-cash and e-check, i.e., it can be used freely as an e-cash and it is as secure as an e-check.

e) To propose an enhanced mobile SET protocol by overcoming the flaws in the existing mobile SET protocols using DSMR (Digital Signature with Message Recovery) mechanism and Mobile Agents.

f) To design mobile payment protocols which ensure transaction security properties including authentication, integrity, confidentiality, non repudiation, anonymity, avoids double spending, over spending, and money laundering.

g) To verify the proposed mobile payment protocols using formal verification methods such as BAN Logic and formal verification tools such as AVISPA Tool and Scyther Tool thereby confirming that it satisfies all the security properties and is free from attacks.

1.3. Motivation of the Research Work

The present study was undertaken to provide a number of improvements to the currently prevalent mobile payment protocols as wireless technology is more complex, compared to wired networks. In order to fully deploy m-commerce for business, end to end security needs to be ensured. There are two levels of security (a) Communication level Security and (b) Application level security. Communication level Security at wireless infrastructure protects each individual wireless communication and is the topic of considerable ongoing research. Application level security is at the application level with which we conduct mobile payments thus protecting the legitimate security concerns of all the parties involved, namely the client, the merchant, Issuer, Acquirer and Payment Gateway. We have observed that current security solutions at the communication layer in mobile environment are not adequate for financial transactions. Therefore, security at the application layer must be added in order to achieve end-to-end protection for mobile financial services. Our research work focuses on the higher level security requirement of mobile payment and transaction security. In order to ensure end to end security for mobile payment protocols WPKI should be adopted. Some mobile payments protocols in the literature adopt WPKI for ensuring end to end security but these protocols are not proposed in the Secure Element (SE). So the digital signatures generated cannot be considered as valid signatures because these are not generated in Secure Signature Creation Device (SSCD). Some mobile payment protocols proposed in the literature use SIM as a Secure Element (SE) but the SIM is personalized by the MNO and it is very difficult to install and personalize these mobile payment applications by the Issuer (I) using Over The Air (OTA). Existing Mobile payment protocols in the literature consume more clients' resources (i.e. communication cost and computation cost). Several mobile wallet solutions found in the literature are implemented in the memory of high end mobile phones, significant amount of data is stored unencrypted and they do not necessarily ensure non repudiation. Another reason is the idea of formally verifying mobile payment protocols because designing of mobile payment protocols does not guarantee that it is free from flaws. Several examples in the literature show that carefully designed protocols even were found to have security breaches later (Muhammad et al., 2006) & (Muhammad et al., 2007). So in order to detect flaws in the protocol, formal verification of the mobile payment protocols needs to be done, as it can detect flaws that have lead to protocol failure using manual (such as BAN Logic) and automated verification tools (such as AVISPA and Scyther). Mobile Payment Protocols proposed in the literature were verified using Manual Formal verification methods (such as BAN Logic) but not with Automated Methods (such as AVISPA and Scyther Tools), which has inspired us to take up this research work.

1.4. Literature Survey

This study was started by looking into the existing mobile payment protocols in wireless environment. Mobile payment protocols are broadly classified as Remote mobile payments (using GSM/GPRS and Mobile Agents) and Proximity mobile payments (using NFC, RFID and Bluetooth) which are sub-classified into Application domain, Security domain and Transaction Optimization (Computational cost and Communication cost) domain. Security domain is again classified into Verification of mobile payment protocols and Personalization of mobile payment application. Verification of mobile payment protocols are classified into (i) BAN (Burrows, Abadi and Needham) Logic based (ii) AVISPA (Automated Validation of Internet Security Protocols and Applications) and Scyther Tools based. The review of the literature on Remote and Proximity mobile payments and their security is presented in the following six categories (A to F).

A) Application Domain in Remote Mobile Payments

There are many mobile payment protocols proposed in Application domain of Remote mobile payments such as by Hsien et al., (2001), Supakorn Kungpisdan et al. (2003) (SET/A+), Chung-Ming Ou & C.R.Ou (2010) (SETNR/A), Xiaolin Pang et al. (2002) (LITESET/A+). Hsien et al., (2001) proposed a payment instrument i.e. Traveler's check in wired networks. Traveler's check (a payment instrument) was not proposed in the realm of mobile commerce. Traveler's check has the merits of both e-check and e-cash (i.e. it is as secure as an e-check and can be used freely as an e-cash). Horng-Twu et al. (2007) proposed, "A new electronic traveler‘s Check Scheme based on one-way hash function". This scheme is based on utilization of one-way hash function with the properties of security and efficiency. Mastercard and VISA have introduced SET protocol which is a very popular credit-card payment protocol. SET was successfully implemented on fixed networks but it is not easy to implement it on wireless networks because of the nature of the SET itself and the problems in wireless networks. SET is a complex protocol which is implemented using public-key infrastructure (PKI). Adopting PKI to wireless environment is a non trivial task because of the limitations of the wireless communication environment when implementing wireless PKI. Due to these problems, it is very difficult to apply wired PKI system to the wireless environment (OMA, April 2001 and OMA, July 2001). To overcome these limitations, Romao A. and da Silva M. M. (1998) proposed an agent-based SET payment system (SET/A). With SET/A, client is not required to stay connected to the Internet during the whole period of the transaction. An agent containing SET wallet plays the client‘s role in SET payment session. Thus, the client needs to connect to the Internet for short periods during the entire transaction. However, SET/A is vulnerable to attacks because the agent is required to bring SET wallet with it to perform cryptographic operations at the merchant environment which is considered to be hostile so Non repudiation property is not ensured. Wang X. F. et al. (1999) proposed SET/A+ which is a modified version of SET/A in order to solve the problems and limitations of SET/A.

SET/A+ is operated in the larger scenario than that of SET/A, in that, it includes the brokering and negotiation phase which naturally requires the capability of agent in SET protocol. Client‘s Purchase Request is completely generated on the client‘s mobile device before it is brought with an agent to merchant. SET/A+ solves the problem of key compromise at the merchants site but performing all the cryptographic operations at clients site results in the problem of high computational load for the client. Moreover signature can be abused easily in malicious merchant environment, so Non repudiation property is not ensured. In order to overcome the limitations of SET, SET/A and SET/A+ Supakorn Kungpisdan et al. (2003) have proposed to employ the combination of proxy –based solution and the agent technology to secure transactions and solve the problems of implementing SET payment in wireless environments. But the solution provided by Chung-Ming Ou & C.R.Ou (2010) has to trust a proxy server, the client need to validate certificate of the issuer every time it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client. If digital signatures are generated in the memory of the mobile phone then these signatures should not be considered as valid signatures because they are not generated in a tamper resistant device. Chung-Ming Ou & C.R.Ou (2010) have proposed SETNR/A protocol to improve the weakness of lacking non-repudiation mechanism from SET and SET/A for credit card-based transactions; on the other hand, agent-based protocol is ideal for complicated payment system. Broker generates a mobile agent for Buyer (i.e. client) which carries encrypted purchase order to Seller (i.e. merchant). A trusted third party (TTP) acts as a lightweight notary for evidence generation. But the solution provided by Chung-Ming Ou & C.R.Ou (2010) has to trust a proxy server, the client need to validate certificate of the Broker and merchant every time, it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client. If digital signatures are generated in the memory of the mobile phone these signatures should not be considered as signatures because they are not generated in a tamper resistant device. Xiaolin Pang et al. (2002) propose LITESET/A+ which does not ensure non repudiation property. Protocols proposed in the literature (Application Domain of Remote mobile payments) are not formally verified. So these protocols cannot be considered as truly secure protocols.

This thesis overcomes these shortcomings by proposing MTC (Mobile Travelers Check) in chapter 4 **(P1)** and EMSET (Enhanced Mobile SET) in chapter 6 **(P6)** in the Application domain of Remote mobile payments.

B) Security Domain in Remote Mobile Payments

Remote mobile payment protocols proposed in the literature are formally verified using BAN logic in the security domain. Remote Mobile payment protocols proposed in the thesis are formally verified using BAN logic given in **P1, P2, P4, P6 and P7.** Remote mobile payment protocols proposed in the literature are not formally verified using AVISPA and Scyther tools but the protocols proposed in this thesis are formally verified using AVISPA and Scyther tools such as **P1, P2, P4, P6 and P7.**

Personalization of Secure Element (UICC) and mobile payment application falls under the category of security domain of Remote mobile payments. There are many mobile payment solutions proposed in the memory of mobile phones or SIM i.e. client's credentials are stored in the memory of Mobile Phones or on the SIM. Mobile Phones and SIM with PKI functionality are personalized by the Issuer (Usually by Mobile Network Service Provider (MNSP)), Service Providers like Banks install Mobile Payment Applications with the help of MNSP's on the SIM. Following are the solutions proposed in the literature of Personalization of Secure Element (UICC) and mobile payment application and their drawbacks. SafeSMS (Hassinen, 2005) is an application aimed for confidentiality, integrity and authentication in Short Message Service (SMS) software java and runs on mobile devices with the Java runtime environment (MIDP 2.0, WMA) installed. It doesn‘t demonstrate how to get the shared secrets and how to generate the session keys among two communicating parties. In addition to these all the secret keys stored in the memory of Mobile Phone could be infected by viruses or can be maliciously replaced.

In J.L.C. Lo et al. (2008) the authors focuses on designing a SMSSec, protocol by making minimal use of computing resources and ensures end-to-end encryption between a WMA client and a server but this mechanism is also implemented in the memory of a mobile phone and faces the same problems as mentioned for M. Hassinen, (2005). S. Chanson & T. Cheung (2001), use a dedicated smart card for mobile authentication. But the phone in the system must have an additional smart card slot where the PKI application, implemented as the SAT (SIM Application Toolkit) application, used for authentication is located in a dedicated java card. An enhancement solution with a single slot was also proposed in their succeeding research. However, the system needs a trusted third party server, namely the User Authentication Server (UAS) in the Internet and requires the mobile user to place his/her full trust in the remote network entity. This means that the system is vulnerable if the trusted third party is compromised. In addition, the article doesn‘t discuss the portability of the application (M. Hassinen & K. Hypponen (2005)).In the solution proposed by He R et al. (2008) user's credentials are not generated by the user using On Board Key Generation (OBKG) procedure, user is issued with a readymade PKI-SIM containing his key pair and SAG‘s (Secure Access Gateway) public key which is a root public key on the chip in addition to these SMS is used as a bearer which is an unidirectional transport mechanism so mutual authentication is not possible. In Yong Lee et al. (2007) the system needs a dual slot mobile phone and does not ensure end to end security. Protocol proposed by Gu et al. (2003) does not ensure end to end security. Lee C.S. et al. (2006) & Juul,N.C. & N.H.Jorgenson, (2002) did not explain how end to end security is ensured from mobile phone to service provider by applying WPKI architecture. In Chao-Wen Chang (2008) the client has to trust the Gateway because it generates the shared key so if the Gateway is not trustworthy then the confidentiality is lost, this work cannot resist Man in the middle attack, and authors did not explain how the shared key is generated. We have proposed a unique solution by overcoming these drawbacks in chapter 2 of this thesis (**P4 and P7**).

C) Transaction Optimization Domain in Remote Mobile Payments

Protocols proposed by Téllez, J., & Sierra, J. (2007), Téllez, J. et al. (2006) and Téllez, J. et al. (2008) falls under the category of Transaction Optimization domain of Remote mobile payments (i.e. these protocols consume low computational and communication cost). Following are the solutions proposed in the literature of Transaction Optimization domain of Remote mobile payments and their drawbacks

a) Protocols proposed by Tellez et.al (Téllez, J., & Sierra, J. 2007c; Téllez, J., et al., 2006a) employs symmetric-key operations and (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) employs Digital Signature with Message Recovery using Self-Certified public key schemes based on RSA (which consumes more computational and communication cost compared with ECC).

b) The number of Client interactions with other engaged parties are more.

c) Protocols proposed by Tellez et.al (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., & Sierra, J. 2007c; Téllez, J., et al., 2006a; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) do not ensure forward secrecy and Public Verification.

d) In (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., & Sierra, J. (2007c); Téllez, J., et al., 2006a; Téllez, J., et al., 2006b) protocols every Client C needs to register itself with merchant in merchant registration protocol thereby consuming lot of resources.

e) Téllez, J., & Sierra, J.2007c; Téllez, J., et al., 2006a protocols does not ensure non repudiation.

f) Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., et al., 2006b protocols cannot withstand Replay attack, Impersonation attack, and MITM attack. Téllez, J., et al., 2008 Protocol cannot withstand MITM attack.

g) Security protocols are error prone and are not easy to identify errors and prove their correctness. Mobile Payment Protocols proposed by Tellez et.al. were not verified using Manual Formal verification methods (like BAN Logic) or using Automated Formal verification Tools like AVISPA, Scyther and CryptoVerif.

This thesis proposes an efficient mobile payment protocol SOMMP **(P2)** in chapter 5 of this thesis which optimizes computational and communication cost compared with the protocols proposed by Téllez et al.

D) Application Domain in Proximity Mobile Payments

a) Mobile Wallets were proposed by NTT DoCoMo (NTT DoCoMo, 2012) and Google Wallet (Google Wallet, 2011) but there are many flaws in their proposed mobile wallets

  a) NTT DoCoMo‘s Mobile Wallet is implemented in the memory of mobile phone

  b) NTT DoCoMo‘s Mobile Wallet needs a special mobile phone which supports NTT DoCoMo‘s Mobile Wallet.

  c) Digital forensics firm via Forensics has found many flaws in Google Wallet (Block, R., 2011) such as

    a) Significant amount of data is stored unencrypted within Google Wallet.

    b) While Google Wallet does a decent job by securing all users credit cards numbers (it is not insecurely stored and a PIN is needed to access the cards to authorize payments), the amount of data that Google Wallet stores unencrypted on the device is significant (pretty much everything except the first 12 digits of your credit card) which can lead to social engineering attack.

    c) Privacy of the consumers is not ensured

b) Yannis Labrou et al. (2004) propose a wireless wallet in the memory of mobile phone

c) Ernst-Joachim Steffens et al. (2009) propose a SIM based mobile wallet in SIM which cannot be personalized by the bank using OTA.

d) Current mobile wallet solutions (Yannis Labrou et al. (2004); Ernst-Joachim Steffens et al. (2009); Hao Zhao & Sead Muftic (2011) store client‘s credentials in the memory of Mobile Phones, on the SIM or UICC, Mobile Phones and SIM with PKI functionality is personalized by the Issuer (Usually by MNSP) and Service Providers like Banks install Mobile Payment Applications with the help of MNSP‘s on the SIM. Mobile Payment Applications cannot be personalized by Banks without the intervention of MNSP‘s.

e) Current mobile wallet solutions (Yannis Labrou et al. (2004); Ernst-Joachim Steffens et al. (2009); Hao Zhao & Sead Muftic (2011) do not ensure end to end security in the application layer.

f) Current mobile wallet solutions (Yannis Labrou et al. (2004); Ernst-Joachim Steffens et al. (2009); Hao Zhao & Sead Muftic (2011) was not formally verified using manual verification methods like BAN logic nor with any Automated Tools like Scyther (Cremers, C. J. F (2006); Cremers, C. and Lafourcade, P (2009) and AVISPA (Armando.A et al., 2005).

E) Security Domain in Proximity Mobile Payments

As per our knowledge Proximity Mobile Payment protocols in the literature for example (Yannis Labrou et al. (2004); Ernst-Joachim Steffens et al.(2009); Hao Zhao & Sead Muftic, (2011), Huda Ubaya (2012), Emir Husni et al. (2012), Tan Soo Fun et al. (2008), Chen, W et.al (2010), M. Massoth and T. Bingel (2009) and Weidar Chen et.al (2010)) are neither formally verified using manual verification method (BAN logic) nor with automated tools (AVISPA and scyther). We have verified our proposed proximity mobile payment protocols using manual verification methods such as BAN logic and using automated tools such as AVISPA and Scyther tools in chapter 3 and 7 of this thesis **(P3, P5).**

F) Transaction Optimization Domain in Proximity Mobile Payments

Protocols proposed by Huda Ubaya (2012), Emir Husni et al. (2012), Tan Soo Fun et al. (2008), Chen, W et.al (2010), M. Massoth & T. Bingel (2009) and Weidar Chen et.al (2010) falls into the category of transaction optimization domain of proximity mobile payment protocols which optimizes computational and communication cost. But these solutions have the following drawbacks.

a) Huda Ubaya (2012) solution is a prototype application designed for the implementation of security in e-commerce transactions by using Tag-to-Tag protocol so that the user needs security and comfort during the financial transaction. But this solution has the following flaws

   a. Does not ensure all the security properties (especially non repudiation)
   b. Application is in the memory of mobile phone (i.e. Android)
   c. It is not clear where the session key (shared session key) generated for encryption and decryption during the payment process between merchant and user is stored
   d. The protocol is prone to attacks
   e. The protocol was not verified with formal methods and tools.

b) Emir Husni et al. (2012) solution proposes a secure and efficient protocol that will reduce physical activity of the device owners and reduce transaction time. The data exchange between merchant and payer will be executed without waiting for each other and one transaction will require two data transmissions which are performed by the merchant and payer. Transactions are secured by the use of encryption on each data sent by the merchant and payer. This solution has the following drawbacks

   a. Does not ensure non repudiation
   b. The protocol is prone to attacks
   c. The protocol was not verified with formal methods and tools

c) M. Massoth & T. Bingel (2009) discusses about different mobile payment services compared with a NFC based solution. They showed that NFC is a trend for mobile payment solution. But it’s not clear how NFC technology improves the security of mobile payments.

d) Tan Soo Fun et al. (2008) proposed a symmetric key centric mobile payment system that constructs on the mobile operator protocol. Pointing out the symmetric mobile payment system has performance advantages on the limited computation platform than the PKI based system, e.g. the handset. Their symmetric system reduces the communications steps between engaging parties meanwhile the security is still not compromised. Their solution does not ensure non repudiation and moreover their solution is proposed in the memory of mobile phone.

e) Chen, W et al. (2010) & Weidar Chen et al. (2010) propose mobile payment system for merchants, which can be built on existing GSM and NFC architecture. Their proposal leverages the SIM’s authentication and identification capabilities and uses GSM and 3G cryptographic primitives, which simplifies integration into the current mobile infrastructure but their solution does not ensure non repudiation, and their solution is proposed in the memory of mobile phone. Proposed mobile payment system trusts MNO for ensuring all the security properties. The client needs to keep his trust MNO this means the system is vulnerable if MNO is compromised.

f) Mateja Jovanovic and Mario Muñoz Organero et al. (2011) proposes new mobile commerce proximity payment architecture, based on the analysis of existing solutions and current and future market needs. But they did not propose any protocol.

g) In the existing proximity mobile payments solutions client’s credentials are generated and stored in the memory of mobile phone and could be infected by viruses or can be maliciously replaced. It does not ensure end to end security in the application layer.

h) Protocols proposed in the realm of Proximity Mobile Payments were formally verified using manual verification methods like BAN logic but not with Automated Tools like Scyther ((Cremers, C. J. F.(2006)) & (Cremers, C. & Lafourcade (2007))) and AVISPA (Armando.A et al., 2005).

This thesis proposes a Secure and Optimized Proximity Mobile Payment protocol (SOPMP) in chapter 7 **(P5)** which optimizes computational and communication cost compared with other protocols in the literature of proximity mobile payments. In the figure 1.2, the tree structure of the reviewed literature on mobile payments is presented.

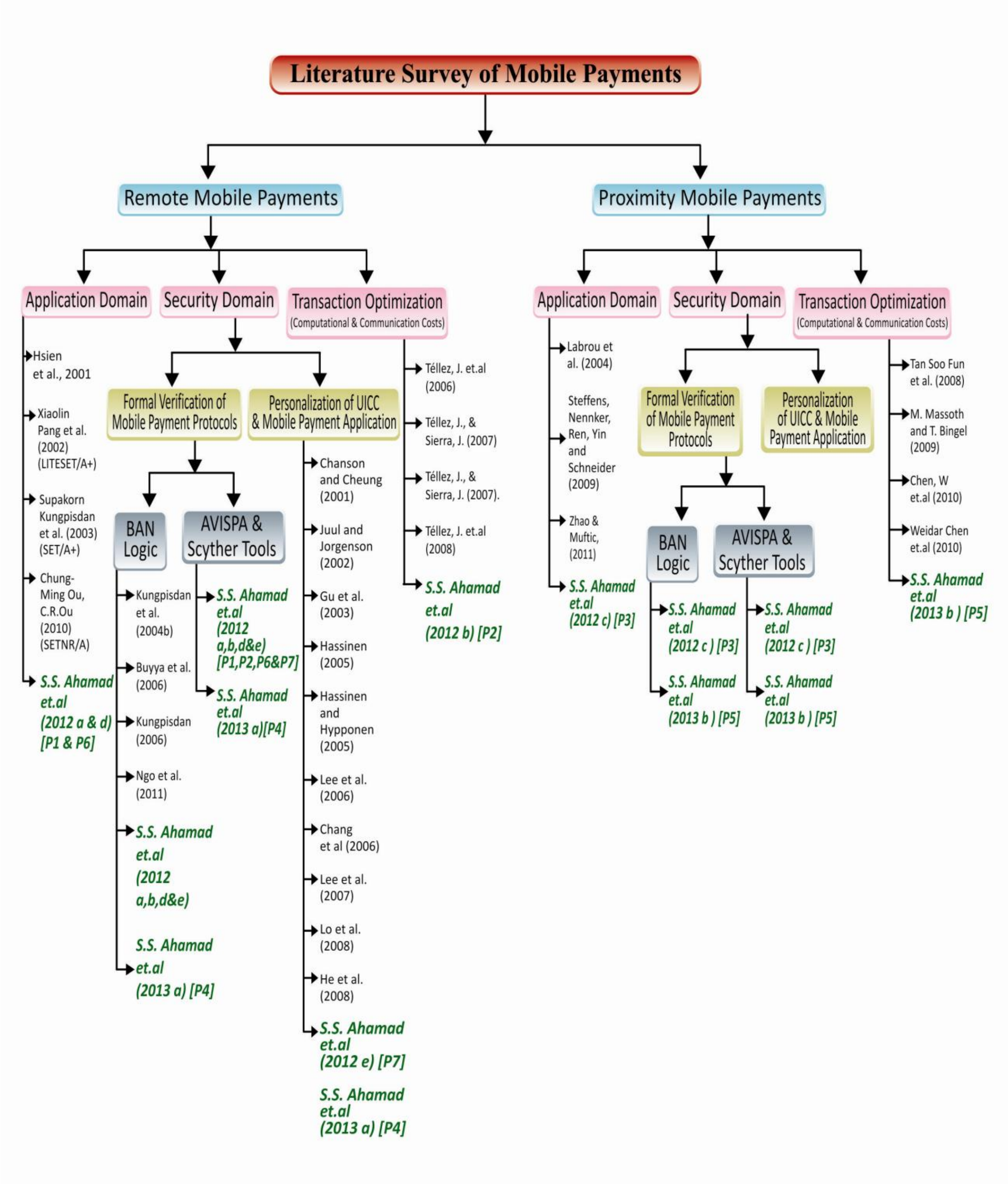

Literature Survey of Mobile Payments
Remote Mobile Payments
Proximity Mobile Payments
Application Domain
Security Domain
Transaction Optimization (Computational & Communication Costs)
Hsien et al., 2001
Xiaolin Pang et al. (2002) (LITESET/A+)
Supakorn Kungpisdan et al. (2003) (SET/A+)
Chung-Ming Ou, C.R.Ou (2010) (SETNR/A)
S.S. Ahamad et.al (2012 a & d) [P1 & P6]
Formal Verification of Mobile Payment Protocols
Personalization of UICC & Mobile Payment Application
BAN Logic
AVISPA & Scyther Tools
Kungpisdan et al. (2004b)
Buyya et al. (2006)
Kungpisdan (2006)
Ngo et al. (2011)
S.S. Ahamad et.al (2012 a,b,d&e)
S.S. Ahamad et.al (2013 a) [P4]
S.S. Ahamad et.al (2012 a,b,d&e) [P1,P2,P6&P7]
S.S. Ahamad et.al (2013 a)[P4]
Chanson and Cheung (2001)
Juul and Jorgenson (2002)
Gu et al. (2003)
Hassinen (2005)
Hassinen and Hypponen (2005)
Lee et al. (2006)
Chang et al (2006)
Lee et al. (2007)
Lo et al. (2008)
He et al. (2008)
S.S. Ahamad et.al (2012 e) [P7]
S.S. Ahamad et.al (2013 a) [P4]
Téllez, J. et.al (2006)
Téllez, J., & Sierra, J. (2007)
Téllez, J., & Sierra, J. (2007).
Téllez, J. et.al (2008)
S.S. Ahamad et.al (2012 b) [P2]
Application Domain
Security Domain
Transaction Optimization (Computational & Communication Costs)
Labrou et al. (2004)
Steffens, Nennker, Ren, Yin and Schneider (2009)
Zhao & Muftic, (2011)
S.S. Ahamad et.al (2012 c) [P3]
Formal Verification of Mobile Payment Protocols
Personalization of UICC & Mobile Payment Application
BAN Logic
AVISPA & Scyther Tools
S.S. Ahamad et.al (2012 c) [P3]
S.S. Ahamad et.al (2013 b) [P5]
S.S. Ahamad et.al (2012 c) [P3]
S.S. Ahamad et.al (2013 b) [P5]
Tan Soo Fun et al. (2008)
M. Massoth and T. Bingel (2009)
Chen, W et.al (2010)
Weidar Chen et.al (2010)
S.S. Ahamad et.al (2013 b) [P5]

Figure 1.2: Literature Survey of Mobile Payments

1.5. Preliminaries of the Thesis

1.5.1. Entities Involved in the proposed mobile payment protocols of the Thesis

There are five entities (which are directly involved) and two entities (which are indirectly involved) involved in the proposed protocols of the Thesis. Entities which are directly involved are Payment Gateway (PG), Issuer (I), Acquirer (A), Client (C) and Merchant (M). PG, A and I are connected through secure private banking network and messages are securely exchanged through this network. Entities which are indirectly involved in the protocol are CA and MNSP/MNO. Figure 1.3 shows the entities involved in the proposed mobile payment protocols of the thesis. In order to ensure end to end security UICC and WPKI are used in the protocol thereby ensuring Confidentiality, Integrity, Authentication, and Non-repudiation.

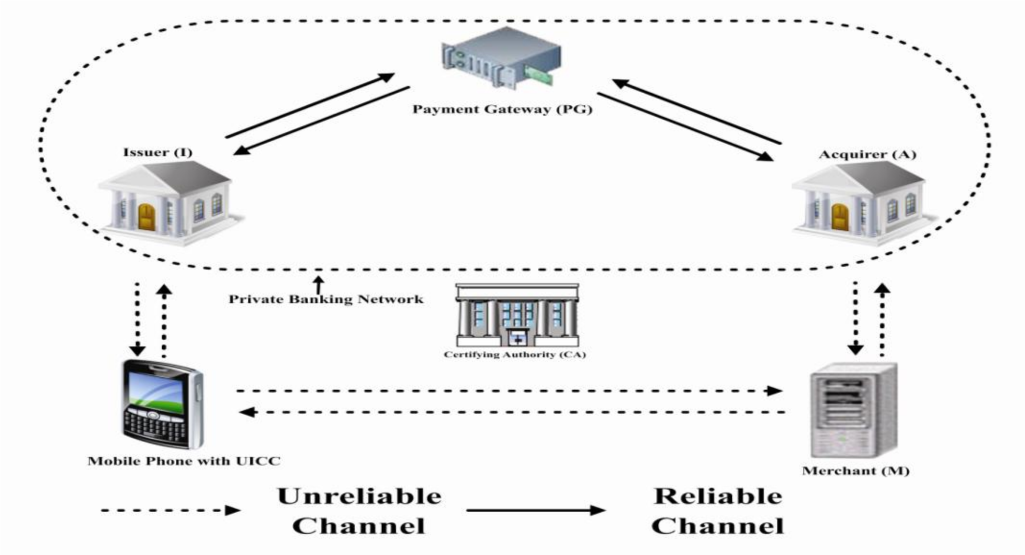


Figure 1.3: Entities involved in the proposed Mobile Payment Protocols of the Thesis

In our proposed protocols each entity has its own certificates or WSLC (WPKI Short Lived Certificates) to prove their authenticity and validity of public keys, certificates or WSLC are maintained by CA in its directory. A public-key certificate is a data structure consisting of a data part and a signature part. Table 1.3 provides a list of entities involved in the protocols of this thesis with their keys and their certificates.

Payment Gateway (PG): Payment Gateway (PG) is an entity that acts as a medium between A and I at Private banking network whose function is to clear and settle funds (i.e. it acts as an arbitrator). PG is trusted by all the engaging parties. A secure private banking network connecting (I, A and PG) ensures security of the messages exchanged among them.

Issuer (I): Banks are Key Stakeholders in this ecosystem as the actual revenue generation is through a transaction which starts from and ends at the bank. Banks play a bigger role by acting as an Application Issuer (AI)/ Service Providers (SP) providing secure banking services to clients through their application implemented on a UICC. In our proposed mobile payment protocol Bank/Issuer personalizes mobile payment application. Issuer (I) is the financial institution of the Client (C) and is trusted by the Client (C). It provides the payment instrument to the Client (C) and manages Client's accounts including funds transfer. A symmetric secret key is shared between Issuer (I) and Client (C) (K_{CI}). Payment Instruction (PI) contains sensitive information such as credit/debit card account number and is encrypted using shared secret key between C and I i.e. Shared secret key is known only to Client (C) and Issuer (I).

Acquirer (A): Acquirer (A) is the financial institution of the beneficiary or payee or Merchant (M) and is trusted by the Merchant. It manages M's account.

Client (C): Client is an entity which is directly involved in our proposed protocols of this thesis. She/he possesses a mobile phone with UICC as a Secure Element with GPRS, Mobile Agent and NFC as communication channel.

Merchant (M): Merchant is an entity which is directly involved in our proposed protocols of this thesis. Merchant (M) sells goods or services and is paid by the Client (C).

Certifying Authority (CA): CA is an entity which is indirectly involved in our proposed protocols of this thesis. CA is responsible for establishing and vouching for the authenticity of public keys. In certificate-based systems, this includes binding public keys to distinguished names through signed certificates, managing certificate serial numbers, and certificate revocation. In addition to these functions CA also acts as a Trusted Service Manager (TSM) who establishes the link between the Service Providers' (SP's) and MNSP's worlds from a technical perspective. The TSM role is also to ensure a level of trust and confidentiality between the entities.

Mobile Network Service Provider (MNSP)/ Mobile Network Operator (MNO): MNSP/MNO is an entity which is indirectly involved in our proposed protocols of this thesis. Mobile Network Service Provider (MNSP) is the crucial stakeholders of this ecosystem as it provides and controls connectivity and enables the entire over-the-air (OTA) transaction system and services. MNSPs provide UICC which can be used as a Secure Element with some modification so they can potentially be an issuer of SE.

UICC: Universal Integrated Circuit Card **(**UICC) is the Secure Element used in this thesis which is a generic platform for smart card applications. It has been standardized by ETSI SCP (ETSI Project Smart Card Platform). The UICC can host a number of different applications, either from the UICC issuer or from other parties, each defining and controlling its own application(s). The UICC has a separate security domain for each application, administered by the application issuer and based on the use of secret administrative keys. The card's operating system implements a firewall that will prevent applications from accessing or sharing data between them. Applications are loaded/ stored in the card's EEPROM memory and also can usually be removed if the application is no longer needed. The freed space can then be used to store some other application if required. UICC hosts the Subscriber Identification Module (SIM) and is supported by the GSMA, has many benefits. UICC smartcards are expected to be globally the most widely distributed Secure Element and the UICC has been well standardized, with well-proven enrolment processes in practice MobeyForum (2008). New service channels and concepts have created new requirements where cross-industry collaboration is needed for the creation of successful business models of Mobile Payments. MNSPs and Banks need close collaboration thereby offering cost-effective and compelling services to the clients. Although UICC is the property of MNSP there are several alternatives for ownership, issuance and management of a shared UICC based secure element. Shared use of the SE has been compared with the property world.

Different property management models have different agreements between parties; they are Hotel Concept, Rental Building Concept and Ownership Concept as given in (MobeyForum 2008). We adopt Rental Building Concept in which the Issuer (I) will hire space on the UICC from MNSP to install Mobile Payment Applications. Certifying Authority will act as a Trusted Service Manager (TSM). This thesis focuses on the enhancement of application level security thereby ensuring end to end security. Application level security is ensured by our proposed secure mobile payment protocol based on UICC. The list of entities with their key pairs and certificate are given in table 1.3 where $T_X^{'}$ is the validity period of the certificate issued for entity 'X'.

Principal/Entities	Key pair	Certificate
Payment Gateway (PG)	(K_{PG}, K_{PG}^{-1})	$Cert_{PG} = (PG, K_{PG}, T_{PG}^{'})_{K_{CA}^{-1}}$
Issuer (I)	(K_I, K_I^{-1})	$Cert_I = (I, K_I, T_I^{'})_{K_{CA}^{-1}}$
Acquirer (A)	(K_A, K_A^{-1})	$Cert_A = (A, K_A, T_A^{'})_{K_{CA}^{-1}}$
Client (C)	(K_C, K_C^{-1})	$Cert_C = (C, K_C, T_C^{'})_{K_{CA}^{-1}}$
Merchant (M)	(K_M, K_M^{-1})	$Cert_M = (M, K_M, T_M^{'})_{K_{CA}^{-1}}$
Certifying Authority (CA)	(K_{CA}, K_{CA}^{-1})	$Cert_{CA} = (CA, K_{CA}, T_{CA}^{'})_{K_{CA}^{-1}}$

Table 1.3: List of Entities with their Keys and related Certificates

1.5.2. Technologies and Cryptographic Mechanisms used in the Thesis

a) Wireless Public Key Infrastructure (WPKI) Technology:

PKI used for wired networks has limitations for its use in resource constrained devices like mobile phones because of less bandwidth, more latency, and insecure connection, less computational power, less memory size, low battery power, small display and input device of mobile device. Due to these problems, it is very difficult to apply wired PKI system in wireless environment. A mobile phone lacks computing capabilities of PKI services such as key generation, digital signature generation and verification, certificate validation, and Certificate Revocation List (CRL) verification, and memory size of storing certificate and CRL. Due to less wireless communication bandwidth, processing of CMP (Certificate Management Protocol) for certificate life cycle such as certificate issue in the mobile phone, and downloading CRL required for certificate verification can be a considerable burden (OMA, April 2001 and OMA, July 2001). Wireless Public Key Infrastructure (WPKI) is the core cryptographic mechanism for non-repudiation protocol; it consists of two parts, one is the operation; the other is the entity. UICC (Universal Integrated Circuit Card) is personalized by the client. UICC personalization involves the following steps i) Generation of Private Keys (one for Authentication and the other for Signing) using On Board Key Generation (OBKG) procedure and ii) Two PINs and One PUK (PIN Unblocking Key) code are used protecting Private Keys. There are two major WPKI operations in our non-repudiation protocol, one is the PKI encryption and decryption, the other is the digital signature-based evidence generation and verification.

b) Over-The-Air (OTA) Technology:

Over-the-Air (OTA) technology contributes dynamic spirit of the NFC based system adaptability to flexible environments. It enables loading and installation of new Mobile applications on SEs - especially on UICCs - remotely, activation and deactivation of SEs, remote service management, life-cycle management of Mobile Payment applications on the SEs, and other online services. High-capacity bearers those are being used in OTA technology are very important in providing an NFC solution. For instance, several kilobytes of data needs to be transferred to the UICC based SE when downloading an application activation data or a Mobile application. Using GPRS/UMTS and the BIP (Bearer Independent Protocol) protocol, applications are rapidly deployed OTA to the UICC card.

c) Signcryption Mechanism:

Signcryption is a new paradigm in public key cryptography that simultaneously fulfills both the functions of digital signature and public key encryption in a logically single step, and with a cost significantly lower than that required by the traditional “signature followed by encryption” approach. Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve consumes less computational and communication cost. Signcryption Scheme with forward secrecy (SFS) based on Elliptic curve combines digital signature and encryption functions (Hwang et al., 2005). This scheme takes lower computation and communication cost to provide security functions. SFS not only provides message confidentiality, authentication, integrity, unforgeability, and non-repudiation, but also forward secrecy for message confidentiality and public verification. In this scheme, the judge can verify sender’s signature directly without the sender’s private key when dispute occurs. This scheme can be applied to mobile communication environment more efficiently because of the low computation and communication cost.

d) DSMR (Digital Signature with Message Recovery) Mechanism:

For secure payments PKI should be used as it provides trust to the parties engaged in the transaction. PKI uses Digital signatures and a digital signature can be re-presented as a secure base because it provides authentication, data integrity, and non-repudiation cryptography services. However, the traditional digital signature schemes are based on asymmetric techniques which make signature computations very expensive and are not suitable for resource constrained devices like Mobile Phones. Moreover, these schemes suffer from the well-known authentication problem (i.e. an imposter may impersonate any innocent user with a valid but incorrect public Key) which requires the usage of certificates to avoid it (Menezes, Van Oorschot & Vanstone, 1997). The public-key certificate must be verified by a Certificate Authority (CA), and that verification causes an additional information exchange during a transaction which adds communication cost for mobile phones. The solution is to use non-traditional digital signature scheme to consume less resources i.e. Digital Signature with message recovery using self certified public keys. Digital Signature with message recovery using self certified public keys (Shiang-Feng Tzenga & Min-Shiang Hwang (2004); Zuhua Shao (2004); Chang, Chang & Huang (2005); Tseng, Jan, & Chien (2003)) provides an authenticated encryption scheme that integrates the mechanisms of signature and encryption, which enable only the specified receiver to verify and recover the original message. The authentication of the public key can implicitly be accomplished with signature verification. Digital Signature with Message Recovery (DSMR) based on ECDSA mechanism eliminates the need of certificates and removes the hurdle of PKI thereby reducing the consumption of resources. In addition to this DSMR requires smaller band width for data communications in order to achieve confidentiality, integrity, authentication and non repudiation properties. The authentication of public keys is implicitly being accomplished with DSMR verification. For reducing the size of messages and for greater efficiency in terms of key sizes and bandwidth DSMR mechanism based on ECDSA is used. So ECDSA is suitable for resource constrained devices. Both the public key and the identity of the user can be authenticated simultaneously when recovering messages. We use DSMR mechanism proposed by Zuhua Shao (2004) in this thesis.

1.5.3. Common Assumptions used in the Thesis

a) Issuer (I) is the Client's (C) financial institution and is trusted by the Client (C). It provides the payment instrument to the Client (C) and manages Client's accounts including funds transfer.

b) Acquirer (A) is the Merchant's (M) financial institution and is trusted by the Merchant. It manages M's account.

c) Payment Gateway (PG) is an entity that acts as a medium between A and I at Private banking network whose function is to clear and settle funds (i.e. it acts as an arbitrator). PG is trusted by all the engaging parties.

d) Payment Instruction (PI) is known only to Client (C) and Issuer (I).

e) A symmetric secret key (K_{CI}) is shared between Issuer (I) and Client (C).

f) Client (C) and Merchant (M) have agreed on the price and description of the goods (or) services

g) Single CA is used in our model which issues certificates for all the engaged parties and is trusted by all the engaged parties in the payment model.

h) CA's public-private key pair is denoted by (K_{CA}, K_{CA}^{-1}) and K_{CA} is distributed in an authenticated manner to every principal participant.

i) A secure private banking network connecting (I, A and PG) ensures security of the messages exchanged among them.

j) All the engaged parties/actors/entities have their own digital certificates.

1.6. Security Goals

To analyze our proposed Mobile Payment protocols a generic set of security goals are defined in subsequent subsections. Security goals are categorized into four sections namely Data security, Client Security, Bank (Issuer, Acquirer and PG) security and Merchant security. We present security goals in each section

Data Security

Third party

Goal 1: *Proposed Mobile Payment protocol should ensure transaction security (i.e. it should ensure Authentication, Confidentiality, Integrity and Non Repudiation properties) and should with stand known attacks such as Replay, Impersonation, MITM and Multi Protocol Attacks.*

Confidentiality: Confidential information must be protected from being viewed by an unauthorized entity. A system providing confidentiality removes the risks from an eavesdropper or attacker.

Authentication: Authentication ensures verification of two interacting parties to gain the right to access a system or resource to take part in a transaction. The purpose is to prevent anyone impersonating anyone else. For a network communication system it is important to ensure authentication as the two parties are not directly connected.

Integrity: The information and systems must be guaranteed against tampering by outside parties. With integrity the transaction information persists intact and unaltered during transmission. A system with a high level of integrity should be difficult to tamper with or to alter.

Non-repudiation: The client must not be able to deny the performed transaction and must provide proof in case this situation occurs. Non-repudiation ensures that the originator cannot falsely refuse or deny a transaction. Non-repudiation is an indispensable security requirement for systems in electronic business and mobile commerce where disputes of transactions can occur.

Replay Attacks: A replay attack is one in which an attacker obtains a copy of an authenticated message and later transmits it to the intended destination.

Impersonation attack: An Intruder (In) tries to impersonate a client (C) to CA/Bank/TELCO, which results in CA/Bank/TELCO being cheated and then the proposed protocol should resist those types of attacks.

Man In The Middle Attack: Man in the Middle Attack is a common attack of intercepting communication in banking protocols where the attacker is able to read, insert, and modify messages in the intercepted communication. The attack targets the integrity of the protocol.

Multi Protocol Attacks: A multi-protocol attack is an attack in which more than one protocol is involved. The verification methods for security protocols that assume a single protocol on a network will fail to verify a protocol’s resistance/vulnerability to multi-protocol attacks. Further, multiple security protocols that are verified to be correct in isolation can be susceptible to multiprotocol attacks when used over the same network.

Goal 2: *In a Mobile Payment system, from the view of the client, the merchant should not have access to client's Payment Information (PI) i.e. Payment Secrecy should be achieved and the bank (I,A,PG) should not have access to client's OI i.e. Order Secrecy should be achieved. In addition to this transaction privacy is to be achieved from PG and Eavesdropper.*

Order Secrecy: No one but the client and the merchant should know the Order Instructions (OI) / Purchase Order (PO) of any given purchase, i.e., order secrecy.

Payment Secrecy: No one but the client and the Issuer should know about Payment Information (PI) of any given purchase, i.e., payment secrecy.

Goal 3: *In a Mobile Payment system, every transaction processed should be unique*

Client Security

Goal 4: *In a Mobile Payment system, the client should obtain unforgeable proof of other participant's Authenticity before it engages in a payment protocol with that participant.*

Goal 5: *In a Mobile Payment system, the client should obtain unforgeable proof of Transaction authorization by the bank.*

Goal 6: *In a Mobile Payment system, the client should be able to achieve Identity Protection from merchant and eavesdropper*

Identity Protection/Anonymity: Merchant should not be able to know the real identity of the Client (C).

Bank Security (Issuer, Acquirer and PG)

Goal 7: *In a Mobile Payment system, the bank should be presented with an un forgeable proof, certifying the authenticity of the other participants.*

Goal 8: *In a Mobile Payment system, the bank before it authorizes a transaction should obtain unforgeable proof from the Client and Merchant, certifying that the Client and Merchant has agreed to the transaction details and authorized to proceed with the transaction.*

Order Instruction Authorization: The Acquirer (A) authorizes order instructions (OI) to the PG if and only if hashed OI sent by merchant (HOI_M) and hashed OI sent by client (HOI_C) are same.

Payment Information Authorization: The Issuer (I) authorizes payment authorization to the Payment Gateway (PG) if and only if it has valid PI.

Goal 9: *In a Mobile Payment system, the bank should be able to prevent Double Spending, Overspending and money laundering*

Goal 10: *In a Mobile Payment system if any one or all the entities involved in private banking network (Issuer/Acquirer/PG)* turns malicious *then they should not succeed in performing transaction on behalf of the client.*

Merchant Security

Goal 11: *In a Mobile Payment system, the Merchant should be presented with a unforgeable proof, certifying the authenticity of the other participants.*

Goal 12: *In a Mobile Payment system, the Merchant before it authorizes a transaction should obtain unforgeable proof from the Client*

1.7. Research Contributions

The contributions of the research work presented in the thesis are briefly explained below and diagrammatically shown in figure 1.4 we have:

a) Proposed protocols for the Personalization of Universal Integrated Circuit Card (UICC) and personalization (Mutual Authentication & Key Agreement Protocol) of Mobile Payments Application (which is on UICC) by the Bank.

b) Proposed a Secure Mobile Wallet Protocol (SMWP) based on WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card) which is a tamper-resistant security-sensitive device, thereby overcoming the shortcomings of the existing mobile wallet solutions.

c) Proposed a Secure and Optimized Mobile based Merchant Payment (SOMMP) protocol using Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve which consumes less computational and communication cost. SOMMP overcomes the demerits of Tellez et.al protocols.

d) Proposed a Mobile Traveler's check (a payment instrument) and its payment process which has the merits of both e-check and e-cash (i.e. it is as secure as e-check and can be used freely as an e-cash)

e) Proposed an enhanced version of Secure Electronic Transaction (SET) in mobile environment named EMSET (Enhanced Mobile SET) using Mobile Agents technology and DSMR (Digital Signature with Message Recovery) mechanism overcoming the shortcomings of SET/A, SET/A+, SETNR/A and LITESET/A+.

f) Proposed a Secure and Optimized Proximity Mobile Payment (SOPMP) Protocol using NFC by adopting WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card) thereby overcoming the demerits of the existing mobile proximity based mobile payment solutions.

g) All the mobile payment protocols proposed in this thesis are verified using BAN logic and with Automated Formal verification Tools like AVISPA and Scyther.

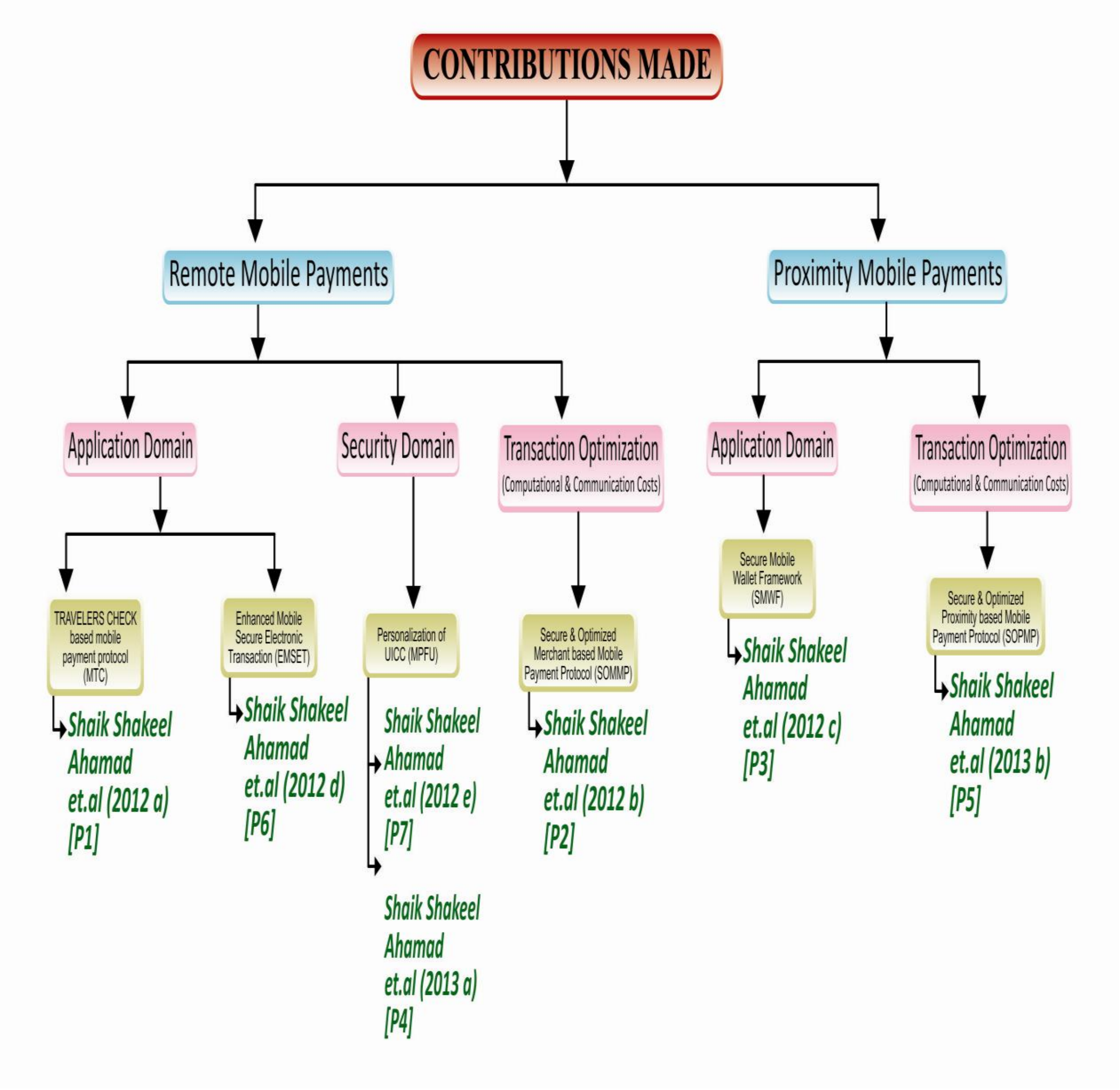

CONTRIBUTIONS MADE
Remote Mobile Payments
Proximity Mobile Payments
Application Domain
Security Domain
Transaction Optimization
(Computational & Communication Costs)
Application Domain
Transaction Optimization
(Computational & Communication Costs)
TRAVELERS CHECK based mobile payment protocol (MTC)
Shaik Shakeel Ahamad et.al (2012 a) [P1]
Enhanced Mobile Secure Electronic Transaction (EMSET)
Shaik Shakeel Ahamad et.al (2012 d) [P6]
Personalization of UICC (MPFU)
Shaik Shakeel Ahamad et.al (2012 e) [P7]
Shaik Shakeel Ahamad et.al (2013 a) [P4]
Secure & Optimized Merchant based Mobile Payment Protocol (SOMMP)
Shaik Shakeel Ahamad et.al (2012 b) [P2]
Secure Mobile Wallet Framework (SMWF)
Shaik Shakeel Ahamad et.al (2012 c) [P3]
Secure & Optimized Proximity based Mobile Payment Protocol (SOPMP)
Shaik Shakeel Ahamad et.al (2013 b) [P5]

Figure 1.4: Contributions made

1.8. Organization of the Thesis

The research work covering mobile payment protocols and focusing on how these protocols ensure end to end security is organized into eight chapters described below:

Chapter 1 provides an overview, aims and objectives, motivation, literature survey on mobile payments and formal methods for security analysis. Our research contributions, structure of thesis and conclusions.

Chapter 2 proposes a procedure for the Personalization of Universal Integrated Circuit Card (UICC), for secure mobile payments.

Chapter 3 proposes a Secure Mobile Wallet Protocol (SMWP) based on WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card).

Chapter 4 introduces a new payment instrument called Mobile Traveler's Check (MTC) in the realm of mobile commerce based on traveler's check concept which is as secure as an e-check and which can be used freely as an e-cash.

Chapter 5 proposes a Secure and Optimized Mobile based Merchant Payment (SOMMP) protocol using Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve which consumes less computational and communication cost.

Chapter 6 proposes an Enhanced Mobile Secure Electronic Transaction (EMSET) in Cellular Network using Mobile Agent technology and Digital Signature with Message Recovery (DSMR) mechanism based on Elliptic Curve Digital Signature Algorithm (ECDSA).

Chapter 7 proposes a Secure and Optimized Proximity Mobile Payment (SOPMP) Protocol using NFC (Near Field Communication) technology, WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card).

Chapter 8 concludes the thesis by summarizing the unique characteristics of our proposed protocols, important findings made by this thesis and scope for future work.

1.9. Conclusions

This chapter provides an overview, objectives, motivation of secure mobile payment discusses formal methods for security analysis of mobile payments, literature survey, research contributions, structure of thesis and conclusions. Mobile payment transactions demands end to end security i.e. all the transactions should ensure Confidentiality, Authentication, Integrity and Non repudiation properties. In addition to these proposed mobile payment protocols should withstand any type of attack (such as replay, Impersonation, MITM (Man In The Middle) and Multiprotocol attacks). In this study we have surveyed most of the mobile payment protocols both in remote and proximity domains. Despite the existence of many protocols in both Remote and proximity domains there are still considerable room for improvements as none of these protocols satisfy all the security requirements and withstand all the identified attacks. Based on the literature survey following observations are made

a) All the mobile payment protocols in the literature are proposed in the memory of the mobile phone, electronic signatures generated in the memory of mobile devices using keys generated in the memory of mobile devices cannot be considered as qualified signatures because they are not generated in Secure Signature Creation Device (SSCD). So these protocols do not ensure non repudiation as they are not generated in SSCD.

b) Most of the mobile payment protocols in the literature are not optimal in terms of consumption of resources (such as computational cost and communication cost).

c) Most of the mobile payment protocols in the literature are vulnerable to attacks.

d) Most of the mobile payment protocols in the literature are not formally verified using AVISPA and Scyther Tools.

Chapter 2: Personalization of Secure Element (Universal Integrated Circuit Card (UICC))

2.1. Introduction

In this chapter we propose a protocol for personalizing UICC by the client (Generation of private key and public key using On-Board Key Generation method), a protocol for personalizing (Authentication and Key-Agreement protocol) the Mobile Payment Application (which is on UICC) by the remote bank server. A mobile payment protocol is proposed between the personalized Mobile Payment Application and remote bank server. Authentication and Key-Agreement protocol and Mobile Payment protocols are formally verified using BAN logic, AVISPA and Scyther Tools.

2.2. Related Works

SafeSMS (Hassinen, 2005) is an application aimed for the confidentiality, integrity and authentication in Short Message Service (SMS) software using java and runs on mobile devices with the Java runtime environment (MIDP 2.0, WMA) installed. It doesn’t demonstrate how to get the shared secrets and how to generate the session keys among two communicating parties. In addition to these all the secret keys are stored in the memory of Mobile Phone and could be infected by viruses or can be maliciously replaced. J.L.C. Lo et al. (2008) focuses on designing a SMSSec, protocol by making minimal use of computing resources and ensures end-to-end encryption between a WMA client and a server but this mechanism is also implemented in the memory of a mobile phone and faces the same problems as mentioned for Hassinen, (2005). S. Chanson, T. Cheung (2001), uses a dedicated smart card for mobile authentication, but the phone in the system must have an additional smart card slot where the PKI application is implemented as the SAT (SIM Application Toolkit) application. An enhancement solution with a single slot was also proposed in their succeeding research, however, the system needs a trusted third party server, namely the User Authentication Server (UAS), in the Internet and requires the mobile user to place his/her full trust in the remote network entity.

This means that the system is vulnerable if the trusted third party is compromised. In addition, the article doesn't discuss about the portability of the application (M. Hassinen & K. Hypponen, 2005). In the protocol proposed by He .R et al. (2008), user credentials are not generated by the user using On Board Key Generation (OBKG) procedure, user is issued with a readymade PKI-SIM containing his key pair and SAG's (Secure Access Gateway) public key which is a root public key on the chip. In Yong Lee et al. (2007) the system needs a dual slot mobile phone and does not ensure end to end security. In Gu. J et al. (2003) the system does not ensure end to end security. Lee C.S. et al. (2006) & Juul,N.C. and N.H.Jorgenson (2002) did not explain as to how the end to end security is ensured from mobile phone to service provider by applying WPKI architecture. In the protocol proposed by Chang C.W. et al. (2006) the client has to trust the Gateway because it generates the shared key so if the Gateway is not trustworthy then the confidentiality is lost, this work cannot resist Man in the middle attack, and authors did not explain how the shared key is generated.

Limitations in the existing Mobile Payment solutions

a) User Credentials are stored in the memory of Mobile Phones or on the SIM
b) Mobile Phones and SIM with PKI functionality is personalized by the Issuer (Usually by MNSP)
c) Service Providers like Banks install Mobile Payment Applications with the help of MNSP's on the SIM. Mobile Payment Applications cannot be personalized by Banks without the intervention of MNSP's.
d) Protocols of Mobile Payments were experimentally verified using manual verification methods like BAN logic and SVO Logic but not with Automated Tools like Scyther (C. J. F. Cremers 2006) and AVISPA (A. Armando *et al.* 2005).

Contributions

a) We have proposed a protocol for personalizing UICC by the client (Generation of private key and public key using On-Board Key Generation method),

b) We have proposed a protocol for personalizing (Authentication and Key-Agreement protocol) of Mobile Payment Application (which is on UICC) by the remote bank server.

c) We have proposed a mobile payment protocol using the personalized Mobile Payment Application on UICC and Bank Server.

d) Proposed protocols were formally verified using BAN logic, AVISPA and Scyther Tools.

2.3. Proposed Protocol for the Personalization of UICC by the client

UICC production involves parties active in UICC card development and delivery to the MNSP. UICC development involves UICC card manufacturers, UICC Operating System developers and WPKI application developers. UICC personalization involves the following steps

i) Generation of Private Keys (one for Authentication and the other for Signing) using OBKG (On-Board Key Generation) procedure.

ii) Two PINs and One PUK (PIN Unblocking Key) code are used to protect Private Keys.

The private keys for Authentication and signing are protected with different non repudiation PINs (NRP). The length of NRP is of six digits. The client will set the NRP for both the private keys. A local application on the UICC handles the setting up of the NRP's in offline. The NRP object is blank when the UICC card is issued to the client by MNSP. The User sets the PIN values when the keys are generated by On Board Key Generation (OBKG) procedure. The process of setting the PIN values will be handled by a local application with no network connection (i.e. in offline mode). The authentication and signing function s of UICC card will be blocked after a

maximum of five consecutive failed attempts. If the user blocks authentication and signing functions by mistake then the user can use PUK (PIN Unblocking Key) code to unblock the PIN. The PUK (PIN Unblocking Key) code is also set by the user in the local application of the UICC which is also offline. The length of the PUK (PIN Unblocking Key) code is of eight digits and should be blocked after five consecutive failed attempts.

Personalization of UICC has two stages

a) Pre-Enrolment Stage

a) Enrolment Stage

a) Pre-Enrolment Stage (before the client is enrolled by the MNO)

MNO gets UICC from UICC manufacturers containing platform certificate (issued by CA) and CA's certificate (since CA is TSM). MNO verifies the platform certificate; if the platform certificate verification is successful then it distributes UICC to its clients. MNO acts as Registration Authority (RA) performs an original identification process to securely identify the client. The client's information such as name, identification card no, address, date of application and ICCID (Integrated Circuit Card ID) which is a 19 or 20-digit serial number of the UICC card of WPKI- UICC is enrolled in the database of the MNO and is sent to CA. After successful verification of client's credentials, MNO recommends CA to generate & issue Enrolment Activation Code (EAC) through SMS for the client.

b) Enrolment Stage

i) MNO sends a command to users WPKI- UICC to initiate On Board Key Generation (OBKG) process which includes setting up of PIN's and PUK codes for the generated keys. After the keys were generated, MNSP receives public keys for the newly generated private keys. User also sends a self signed request for the creation of a new certificate from WPKI- UICC.

ii) CA sends a signature request to the user via MNO where the input for the generation of signature is Enrolment Activation Code (EAC) which is issued by CA.

iii) CA receives the signed Enrolment Activation Code (EAC) and verifies it against the one given during pre-enrolment.

iv) CA requests client for its public keys

v) The client responds by sending platform certificates and self signed request by the user for the creation of client certificate.

vi) CA uses the received information from MNO in producing client's Certificates with the information about MSISDN (Mobile Station International Subscriber Directory Number). MSISDN is a number uniquely identifying a subscription in a GSM or a UMTS mobile network).

vii) CA sends a copy of certificate to MNO and Client.

2.4. Proposed Protocol for the Personalization of Mobile Payment Application (which is on UICC) by the Bank (Issuer/Issuing Bank) using OTA

Provisioning is the process of installing a payment application on a UICC. Personalization is the process of putting data specific to a client into the mobile payment application. Personalization and provisioning server is responsible for transferring the application and personalization data to UICC. This includes providing the necessary cryptographic material required by the UICC or application in order to allow installation or personalization. It is also responsible for providing a chain of trust between the bank and UICC, including appropriate logging to assist in audit, repudiation and forensic. The personalization of mobile payment application is done as follows

Assumptions

Every UICC will have its Platform Certificate issued by CA

Every Client will have his/her own Certificate issued by CA

Every Mobile Payment Application will have its own Certificate issued by CA

All the entities involved in the protocol have their own certificates and their public keys

We assume that there is only one CA which generates and issues certificates to all the entities involved in this protocol.

CA maintains Certificates in its directory, Online Certificate Status Protocol (OCSP) and Certificate Revocation List (CRL)

CA also acts a Trusted Service Manager (TSM)

Bank maintains Personalization and Provisioning server which is responsible for transferring the application and personalization data to UICC.

Provisioning and Personalization of Mobile Payment Application is done Over The Air (OTA) by the Issuer

Main Protocol

Step 1: Client sends its platform certificate and client certificate to the Issuer

$C \rightarrow I : C_P, C_C$ /* C_P, C_C are Platform and Client's certificates */

Step 2: Issuer (I) validates platform certificate and clients certificate using Certificate Validation Procedure given in (D.R. Stinson, 2006)

Certificate validation mainly includes (D.R. Stinson, 2006):

a) Verifying the integrity and authenticity of the certificate by verifying the CA's signature over it.
b) Verifying that the certificate is not expired.
c) Verifying that the certificate is not revoked.

Step 3: Client requests Bank for Mobile Payment Application certificate

Step 4: Issuer (I) sends either Mobile Payment Application certificate or URL of Mobile Payment Application certificate to the client

Step 5: Client delegates the process of Mobile Payment Application Certificate Validation to OCSP. OCSP validates the Certificate.

$C \rightarrow OCSP : C_{MPA}$ /* C_{MPA} is Mobile Payment Application Certificate */

Step 6: If the response from OCSP is positive, client requests Bank to install Mobile Payment Application on the UICC. /* this is the provisioning step */

Step 7: Issuer (I) installs Mobile Payment Application on the UICC

Step 8: Bank personalizes Mobile Payment Application on the client's UICC as per Protocol 1.

Protocol 1: Personalization of Mobile payment application

$Step1: C \rightarrow I: \{MS1, DS_I^C(MS1)\}_{K_I}, Cert_C \quad MS1 = \{PI, phno, NRP, T_C, N_C\}$

/* before initiating the process of personalization client validates bank's certificate using Certificate validation procedure given in (D.R. Stinson 2006), mutual authentication is done between client and bank in this step. UICC initiates the process to negotiate shared symmetric key with the Bank and sends $\{MS1, DS_I^C(MS1)\}_{K_I}, Cert_C$. Issuer (I) decrypts the received message from UICC using his private key and checks the authenticity of $DS_I^C(MS1)$ checks the timestamps and nonce if all the checks are successful then it generates a shared symmetric key K_{ci} between the I and C. Issuer sends $\{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I$ to C containing $MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$ Session keys are generated using hashing algorithms with one bit cyclic shift of a master secret each time a session key is generated as shown in Supakorn Kungpisdan et al. (2003). The key set K_{CI} (with {1, 2, 3, .n}) is generated from the secret key K_{ci} and is stored in Mobile Payment Application of the UICC at the client end and in the Issuer server.*/

$Step2: I \rightarrow C: \{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I \quad Where \quad MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$

/* upon receiving the message B checks the authenticity of the message, if the checks are successful then client sends $Step2$/

```
IF Verification (of digital signature on MS1)  $DS_I^C(MS1) = TRUE$ {
    /*Authenticity and Integrity of message MS2 is not compromised*/
    Go to Step 2}        Else  {
    /* Authenticity and Integrity of message MS1 is compromised*/
    Exit}
```

Bank decrypts the received message from UICC using his private key and checks the authenticity of $DS_I^C(MS1)$, checks the timestamps and nonce if all the checks are successful then it generates a shared symmetric key $\boldsymbol{K_{ci}}$ between the I and C. Issuer (I) sends $\{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I$ to C containing $MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$ session keys are generated using hashing algorithms with one bit cyclic shift of a master secret each time a session key is generated as shown in Supakorn Kungpisdan et al. (2003). The key set K_{CI} (with {1, 2, 3, .n}) is generated from the secret key K_{CI} and is stored in Mobile Payment Application of the UICC at the client end and in the Issuer server.

$Step3: C \rightarrow I: \{MS3, MAC0\}_{K_{CI}}$ $MS3 = \{PI, Ack, K_{CI}, T_I, N_C, N_I, T_C\}$

/* Upon receiving the message client C checks the authenticity of the message, if the checks are successful then client sends an acknowledgement response message $\{MS3, MAC0\}_{K_{CI}}$ to I for K_{CI} which is a shared symmetric key for this mobile payment application.*/

```
IF Verification of digital signature on MS2 $DS_C^I(MS2) = TRUE$ {
/*Authenticity and Integrity of message MS2 is not compromised*/
Go to Step 3 }     Else   {
      /*Authenticity and Integrity of message M2 is compromised*/
       Exit }
```

C sends $\{MS3, MAC0\}_{K_{CI}}$ to I containing $MS3 = \{PI, Ack, K_{CI}, T_I, N_C, N_I, T_C\}$

$MAC0 = h(K_{CI}, MS3)$

```
Bank verifies $MAC0$     IF   $(Verf(MAC0) = TRUE)$
{    /*Authentication of/Client, Confidentiality and Integrity of MS0 are successfully verified*/
Bank maps $K_{CI}$ to PI of Client
```

```
  }    Else    {
/* Authentication of Client, Confidentiality and Integrity of $MAC0$ are not verified*/
        Exit  }  }
```

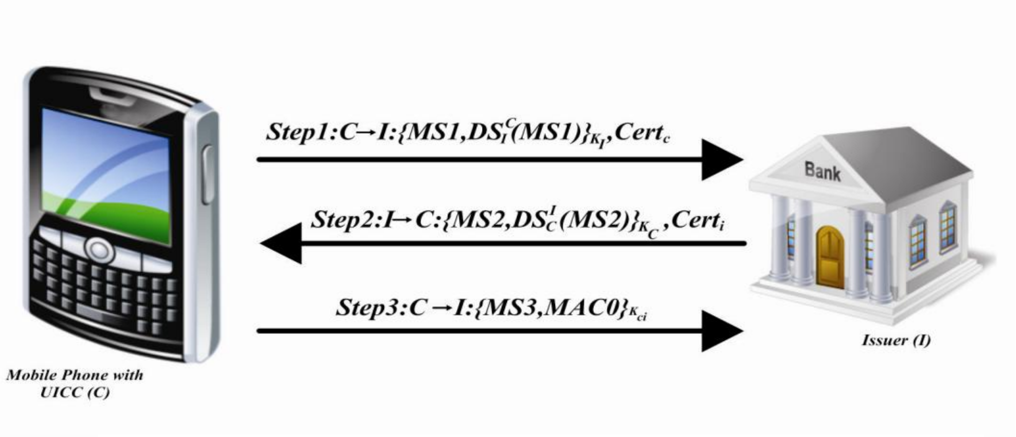

$Step1: C \rightarrow I: \{MS1, DS_I^C(MS1)\}_{K_I}, Cert_c$
$Step2: I \rightarrow C: \{MS2, DS_C^I(MS2)\}_{K_C}, Cert_i$
$Step3: C \rightarrow I: \{MS3, MAC0\}_{K_{ci}}$
Bank
Mobile Phone with UICC (C)
Issuer (I)

Figure 2.1: Personalization of Mobile Payment Application

2.5. Proposed Mobile payment protocol

$Step1: C \rightarrow UC: \langle MPA \rangle_{NRP}$

/* The Client (C) unlocks the Mobile Payment Application (MPA) residing in the UICC card by entering NRP. The Client (C) uses Non-Repudiation PIN to convince the UICC card of his/her identity ***/**

```
$WHILE\ (Attempts \leq 3\ times)$ { IF  $(Verf(NRP) = TRUE)$
{    /* Authentication is successful and User/Client is allowed to use Mobile Payment Application*/
           Go to Step 2}    ELSE    {
/* Authentication is unsuccessful and User/Client is  denied by ‘UC” from using Mobile Payment  Application */
           Exit}  }
```

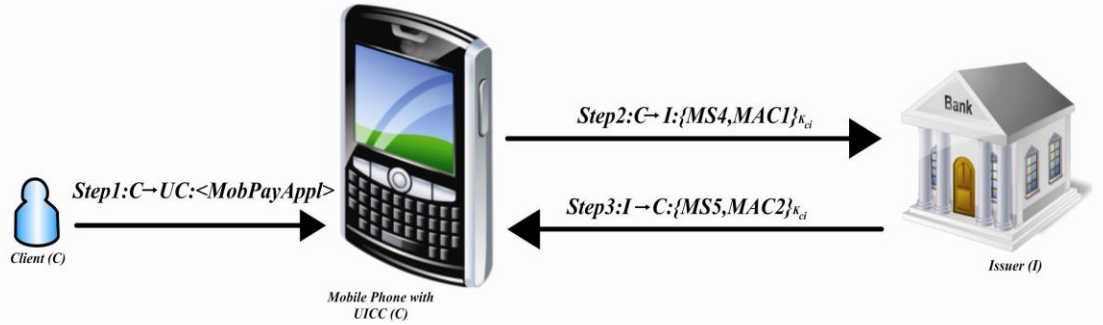

Step1:C→UC:<MobPayAppl>
Client (C)
Mobile Phone with UICC (C)
Step2:C→ I:{MS4,MAC1}$_{K_{ci}}$
Step3:I→C:{MS5,MAC2}$_{K_{ci}}$
Bank
Issuer (I)

Figure 2.2: Mobile Payment Protocol between Client and Issuer

$Step2 : C \to I : \{MS4, MAC1\}_{K_{CI}}$ $MS4 = \{PI, PayeeID, T_I, N_C, N_I, T_C Amt\}$

$MAC1 = h(K_{CI}, MS4)$

I receives $\{MS4, MAC1\}_{K_{CI}}$ message from the Client, decrypts the message using K_{CI} and gets message MS4.If all the verifications are successful then transfers funds from client's account to payee's account and informs both Client and Payee. Mobile Payment application maintains a number of counters and parameters. After the transaction mobile payment application will be updated i.e. these counters will be reset, and parameters are updated by the Update server.*/

```
IF $(Verf(MAC1) = TRUE)$ {
/* Authentication of the Client, Confidentiality and Integrity of MS4 are successfully verified */
  Go to Step 3   }
      Else   {   /* Authentication of/Client, Confidentiality and Integrity of MS4 are
                   not verified*/        Exit  }  }
```

$Step3 : I \to C : \{MS5, MAC2\}_{K_{CI}}$ $MAC2 = h(K_{CI}, MS5)$

$MS5 = \{PayeeID, T_I, N_C, N_I, T_C, Amt, Success\}$

/* C receives $\{MS5, MAC2\}_{K_{CI}}$ message from the Issuer, decrypts the message using K_{CI} and gets message MS5 & $MAC2 = h(K_{CI}, MS5)$ */

```
 IF $(Verf(MAC2) = TRUE)$
{
/* Authentication of/Client, Confidentiality and Integrity of MS5 are successfully verified*/
Issuer Transfers the amt to payee     }
   ELSE    {
        /* Authentication of the Client, Confidentiality and Integrity of MS5 are not verified*/
Exit }}
```

2.6. Security Analysis

We consider the goals defined in section 1.6 of Chapter 1 and perform the security analysis as below.

2.6.1. Data Security

Third party

Goal 1:

All the entities involved in the MPPPU protocol store their credentials (Private Keys, NRP and Certificates) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the MPPU protocol will exchange messages using encryption and digital signature so any third entity will not be able to gain access to participant’s transactional data thereby achieving Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. The Mobile Signature of data provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore if the mobile signature was generated by means of SSCD it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (MPPU) withstands the following attacks

Replay Attacks: Replay attack is avoided by including Timestamps, Nonce and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included in the messages exchanged ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C's private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating digital signature is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Master key for the generation of symmetric key is stored in the mobile payment application which is protected by NRP. Our proposed protocol MPPU withstands this attack because the intruder (In) does not have receiver's private key.

Multiprotocol Attack: Our proposed protocols in this chapter withstand this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2:

In our proposed Mobile Payment protocol (MPPU) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Eavesdropper cannot get PI because the messages are hashed and encrypted thereby achieving Transaction privacy from Eavesdropper.

Goal 3:

In our proposed Mobile Payment protocol (MPPU) every transaction is unique. Every transaction is linked to Timestamps, Nonce and Shared symmetric key.

2.6.2. Client Security

Goal 4:

Our proposed Protocol uses certification authorities, to certify the authenticity of public keys held by the Client and Issuer/Bank. The client's conversations are only with the Issuer/Bank. Client and Issuer's authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5:

In our proposed Mobile Payment Protocol (MPPU) the client obtains unforgeable proof of transaction authorization from the Issuer at the end of protocol (i.e. whether the transaction is success or failure).

Identity Protection from Merchant and Eavesdropper

Goal 6:

Client will be issued an anonymous identity $Anonid_C$ by the CA after successful verification of client's credentials. Client's certificate will have anonymous identity instead of his/her real identity thereby achieving identity Protection from Merchant and Eavesdropper.

2.6.3. Bank Security (Issuer, Acquirer and PG)

Authentication

Goal 7:

Our proposed Mobile Payment Protocol (MPPU) has certifying authority to certify the authenticity of the public keys held by the Client and Issuer. The client's conversations are only with the Issuer. Client and Issuer's authenticity is proved by verifying their certificates. In addition to this client and Issuer share a symmetric key between them.

Authorization

Goal 8:

The Issuer obtains an authorization proof for transaction from the client in the form of PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer.

Prevents double spending, overspending and money laundering

Goal 9:

Issuer (I) keeps the message it has received from Client in its archives. If the client tries to double spend the PI, I can detect this from the symmetric key, timestamp and nonce. So double spending is avoided in MPPU by Issuer (I). If the client tries to overspend, I avoids them in doing so since it checks Client's (C) balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Issuer (I) is always involved in every payment transaction and honoring account to account transfer thereby preventing money laundering.

2.7. Comparative Analysis with the related work

We have compared our proposed Personalization protocol with related works; the report of the analysis based on multiple features is presented in Table 2.1. The last column results are of our proposed protocol.

Protocols Key Features	M. Hassinen (2005 a)		S. Chanson, T. Cheung (2001)	J.L. C. Lo et al (2008)	He. R, Zheng Qin & Xi Qin (2008)	Yong Lee et al. (2007)	Gu .J et al. (2003)	Lee C.S. et al. (2006)	Juul,N. C. and N.H.Jorgenson, (2002)	Chang C.W. et al (2006)	MPPU
Client's credentials are generated using OBKG procedure	No	No	No	No	No	No	No	No	No	No	Yes
WPKI is implemented in the UICC which is personalized by the user	No	No	No	No	No	No	No	No	No	No	Yes
Mobile Payment Application is installed and personalized by the Bank on UICC without the intervention of MNSP	No	No	No	No	No	No	No	No	No	No	Yes
Ensures end to end Security at the Application layer	No	No	No	No	No	No	No	No	No	No	Yes
Withstands Replay Attack	No	No	No	No	No	No	No	No	No	No	Yes
Withstands Impersonation Attack	No	No	No	No	No	No	No	No	No	No	Yes
Withstands MITM Attack	No	No	No	No	No	No	No	No	No	No	Yes
Withstands Multi Protocol Attack	No	No	No	No	No	No	No	No	No	No	Yes

Table 2.1: Comparative analysis of Personalization Protocol in MPPU with Related Works

We have compared our proposed Mobile Payment protocol with related works; the report of the analysis based on multiple features is presented in Table 2.2. The last column results are of our proposed protocol thereby ensuring end to end security.

Protocols Features	Ngo et al. (2011)	Supakorn Kungpisdan (2004a)	Supakorn Kungpisdan (2004b)	Supakorn Kungpisdan (2006)	Gianluigi Me (2005a)	Gianluigi Me, Alex Schuster (2005b)	Buyya Rajkumar et al. (2006)	MPPU
Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Non- Repudiation	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Identity Protection from Eavesdropper	No	No	No	No	Yes	No	No	Yes
Transaction Privacy Protection from Eavesdropper	No	No	No	No	Yes	No	No	Yes
Prevents Double Spending	NR	NR	NR	NR	NR	NR	NR	Yes
Prevents Over spending	NR	NR	NR	NR	NR	NR	NR	Yes
Prevents Money Laundering	NR	NR	NR	NR	NR	NR	NR	Yes
Withstands Replay Attack	Yes	No	No	No	No	Yes	Yes	Yes
Withstands Impersonation Attack	Yes	No	No	No	No	No	Yes	Yes
Withstands MITM Attack	Yes	No	No	No	No	No	Yes	Yes
Withstands Multi Protocol Attack	No	No	No	No	No	No	No	Yes
Experimental Verification using BAN Logic, SVO Logic	Yes	No	Yes	Yes	No	No	Yes	Yes
Experimental Verification using AVISPA and SCYTHER TOOL	No	No	No	No	No	No	No	Yes

Table 2.2: Comparative analysis of our proposed mobile payment Protocol with Related Works

So our proposed mobile payment protocol from Mobile Payments Application (which is on UICC) to the Issuer/Bank ensures Confidentiality, Authentication, Integrity and Non

Repudiation, prevents double spending, over spending and money laundering, withstands replay, MITM, Impersonation and Multiprotocol Attacks.

2.8. Formal Verification of the Proposed Protocols (Mutual Authentication and Key Agreement, Mobile Payment protocols))

2.8.1. Formal Verification of the Proposed Protocol's Security using BAN logic

Assumptions for the verification of the proposed protocol is given in Appendix A

Security Proof of the Proposed Protocol

$Step1: C \to I: \{MS1, DS_I^C(MS1)\}_{K_I}, Cert_C \quad MS1 = \{PI, phno, NRP, T_C, N_C\}$

$Step2: I \to C: \{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I \quad Where \quad MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$

I validate the certificate of 'C' using the method given in D.R. Stinson (2006) which includes

a) Verifying the integrity and authenticity of the certificate by verifying the CA's signature over it.
b) Verifying that the certificate is not expired.
c) Verifying that the certificate is not revoked.

If the certificate validation is successful & from the assumptions given in **AS1, AS2 & AS6** we can conclude that I **believes C said** $Cert_C$ ……..statement (1)

I receives $\{MS1, DS_I^C(MS1)\}_{K_I}$ from C and decrypts the message using its private key from the assumptions **AS1& AS2** and can conclude that

I **believes** $\{MS1, DS_I^C(MS1)\}_{K_I}$ ………..statement (2)

I verifies the digital signature on Message MS1 i.e. $DS_I^C(MS1)$ if the verification is successful then I **believes C said** $DS_I^C(MS1)$ ………..statement (3)

From **AS7** I **believes fresh** T_C ………..statement (4)

From **AS5** I **believes fresh** N_C ………..statement (5)

From Statements (1) to (5) we can conclude that

“I **believes C said** $\{MS1, DS_I^C(MS1)\}_{K_I}, Cert_C$

Now I constructs $\{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I$ and sends it to C Where

$MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$

$Step3: C \rightarrow I: \{MS3, MAC0\}_{K_{CI}} \quad MS3 = \{PI, Ack, K_{CI}, T_I, N_C, N_I, T_C\}$

C validates the certificate of ‘I’ using the method given in D.R. Stinson (2006). If the certificate validation is successful & from the assumptions given in **AS1, AS2 & AS6** we can conclude that

C **believes I said** $Cert_I$ ………..statement (6)

C receives $\{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I$ from I and decrypts the message using its private key from the assumptions **AS1& AS2, we** can conclude that

C **believes** $\{MS2, DS_C^I(MS2)\}$..Statement (7)

C verifies the digital signature on Message MS2 i.e. $\{DS_C^I(MS2)\}$ if the verification is successful then

C **believes I said** $\{DS_C^I(MS2)\}$ ………..Statement (8)

From **AS5** C **believes fresh** T_I ………..Statement (9)

From **AS7** C **believes fresh** N_I …..Statement (10)

From Statements (6) to (10) we can conclude that

“C **believes I said** $\{MS2, DS_C^I(MS2)\}_{K_C}, Cert_I$

Now C constructs $\{MS3, MAC0\}_{K_{CI}} \quad MS3 = \{PI, Ack, K_{CI}, T_I, N_C, N_I, T_C\}$ and sends it to C

Proof of Step 3

I decrypts $\{MS3, MAC0\}_{K_{CI}}$ using the symmetric shared key K_{CI} between C and I as given in **AS4** and gets $\{MS3, MAC0\}$

I **believes C said** $\{MS3\}$ ………..statement (11)

I checks the authenticity of $MAC0 = h(K_{CI}, MS3)$ using shared session key between I and C and MS5 as given in **AS4,** if the check is successful then

I **believes C said** $\{MAC0\}$ …..statement (12)

From **AS5** I **believes fresh** T_C ………..statement (13)

From **AS7** I **believes fresh** N_C ……..statement (14)

From Statements (11) to (14) we can conclude that "I **believes** $\{MS3, MAC0\}_{K_{CI}}$

Mobile Payment Protocol

$Step4 : C \rightarrow UC : \langle MPA \rangle_{NRP}$

The Client C unlocks the Mobile Payment Application (MPA) residing in the UICC card by entering NRP. The Client C uses Non-Repudiation PIN to convince the UICC card of his/her identity as given in **AS3.** UC **believes** that the Client C is authorized to use the Mobile Payment Application loaded on 'UC'

$Step5 : C \rightarrow I : \{MS4, MAC1\}_{K_{CI}}$ $MS4 = \{PI, PayeeID, T_I, N_C, N_I, T_C Amt\}$

$MAC1 = h(K_{CI}, MS4)$

C Sends $\{MS4, MAC1\}_{K_{CI}}$ to I

$Step6 : I \rightarrow C : \{MS5, MAC2\}_{K_{CI}}$ $MAC2 = h(K_{CI}, MS5)$

$MS5 = \{PayeeID, T_I, N_C, N_I, T_C, Amt, Success\}$

$\{MS4, MAC1\}$

I decrypts $\{MS4, MAC1\}_{K_{CI}}$ which he has received using the shared session key between I and C as given in **AS4,** checks PI as given in **AS9** and gets $\{MS4, MAC1\}$

I **believes C said** $\{MS4, MAC1\}$ …..statement (15)

I checks the authenticity of $MAC1 = h(K_{CI}, MS4)$ using shared session key between I and C and MS4 as given in **AS4,** if the check is successful then

I **believes C said** $MAC1$ ………..statement (16)

From **AS5** I **believes fresh** T_C ………..statement (17)

From **AS7** I **believes fresh** N_C ……..statement (18)

From Statements (15) to (18) we can conclude that I **believes C said** $\{MS4, MAC1\}_{K_{CI}}$

Proof of Step 6

C decrypts $\{MS5, MAC2\}_{K_{CI}}$ $MAC2 = h(K_{CI}, MS5)$ using the symmetric shared key K_{CI} between C and I as given in **AS4** and gets $\{MS5, MAC2\}$

$MAC2 = h(K_{CI}, MS5)$ $MS5 = \{PayeeID, T_I, N_C, N_I, T_C, Amt, Success\}$

C **believes I said** $\{MS5\}$ ………..statement (19)

C checks the authenticity of $MAC2 = h(K_{CI}, MS5)$ using shared session key between I and C and MS5 as given in **AS4,** if the check is successful then

C **believes I said** $\{MAC2\}$ ……..statement (20)

From **AS5** C **believes fresh** T_I ………..statement (21)

From **AS7** I **believes fresh** N_I **…… …..** Statement (22)

From Statements (19) to (22) we can conclude that C **believes I said** $\{MS5, MAC2\}_{K_{CI}}$

2.8.2. Formal Verification of the Proposed Protocol's Security using AVISPA Tool

The validation of our proposed protocol (MPPU) specification, using the OFMC back-end in AVISPA Tool is given in Appendix C as figure 2.3.

The validation of our proposed protocol (MPPU) specification, using the CL-Atse back-end in AVISPA Tool is given in Appendix C as figure 2.4

2.8.3. Formal Verification of the Proposed Protocol's Security using SCYTHER Tool

The result of MPPU using "Verification Claim" Procedure of Scyther Tool is given in Appendix C as figure 2.5.

The result of MPPU using "Automatic Claim" Procedure of Scyther Tool is given in Appendix C as figure 2.6.

2.9. Conclusions

This chapter proposes protocols for personalizing UICC by the client, Mobile Payments Application (which is on UICC) by the Bank (Issuer) and a mobile payment protocol between the personalized Mobile Payment Application on UICC and the Bank Server. Our proposed protocols are ensures end to end security. Our proposed mobile payment protocol from Mobile Payment Application to the Bank ensures Confidentiality, Authentication, Integrity and Non Repudiation, prevents double spending, over spending and money laundering, and withstands replay, MITM, Impersonation and Multi Protocol attacks. Proposed protocols were successfully formally verified using BAN logic, AVISPA and scyther tools. We observe that for the successful implementation of this protocol there should be collaboration (i.e. understanding) among all the entities involved in the protocol (i.e. among UICC manufacturers, CA, MNO and Banks). There should be a regulatory authority to look into the functioning of each of these entities involved in the protocol.

Chapter 3: Secure Mobile Wallet Protocol based on UICC

3.1. Introduction

This chapter covers Application Domain of Proximity Mobile Payments. A mobile wallet is a complete payment application for NFC-enabled mobile phone that enables consumers to pay at stores at the point of sale with a mobile phone. The digital wallet, which is associated with a credit card, integrates all payment-related services like the management and storage of receipts, coupons and offers, and loyalty cards. In this chapter we propose a protocol for the personalization of Mobile Payment Application which is in the Mobile wallet. Then we consider a scenario in which client has a NFC enabled Mobile Phone with UICC as secure element, merchant has NFC enabled POS (Point Of Sale) and all the items (in the store) chosen by client are all tagged with NFC tags. Client selects some tagged items at a department store, and approaches one of the several NFC enabled POS in the store. We consider 6 entities: (i) Client with NFC-enabled Smartphone with UICC, (ii) Merchant with NFC enabled POS (iii) Issuer (client's Bank) (iv) Acquirer (merchant's Bank) (v) Payment Gateway (PG) and (vi) Certifying Authority (CA) in our ecosystem. Our proposed protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these our proposed protocol withstands Replay, Man in the Middle, Impersonation and Multi Protocol attacks. The security properties of the proposed protocol have been verified using BAN logic, AVISPA and Scyther Tools and presented with results.

Preliminaries:

UICC: UICC is explained in chapter 1 and is shown in figure 3.1.

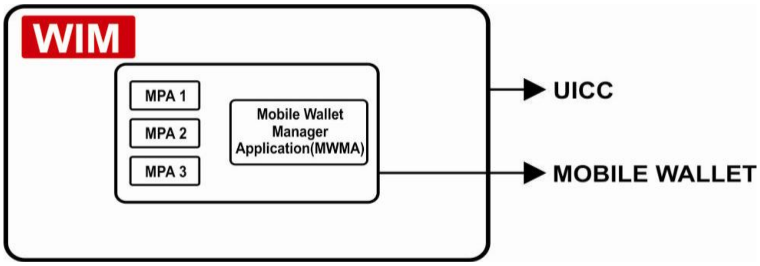


Figure 3.1: Proposed Architecture of UICC

***Mobile Payment Application (MPA*):** Mobile Payment Application module (MPA) is a separate application realizing the function of payment situated inside the UICC. After the successful authentication (using NRP) of the client by MWMA (Mobile Wallet Manager Application), the mobile phone (MP) sends a message to this Application requesting either the credit card information according to the APDU (Application Protocol Data Unit) command. MPA keeps client's personal sensitive information such as credit card number, shared symmetric keys and Issuer's Certificate.

Wireless Identity Module (WIM): Wireless Identity Module (WIM) is another SIM based solution which ensures that key pairs are generated inside the card and private keys never go outside the card. However, this technology requires that WIM application is included in the SIM by the manufacturer. WML/XHTML Script Wireless Markup Language (WML) and eXtensible Hyper Text Markup Language (XHTML) are languages for web application development on mobile sets. These languages contain cryptographic libraries that can be executed on mobile browser and can establish session with WIM application residing inside smart cards. For instance, these libraries contain a function named Sign Text" which supplies plain text to the WIM application. WIM returns the signed text after applying crypto functions with private key residing in the smart card. However, a major restriction with this solution is that the manufacturer has to provide functionality for communication between WIM and the script running in browser

Mobile Wallet Manager Application (MWMA): Mobile Wallet manager Application (MWMA) is a separate application installed by the TSM (in our protocol CA is TSM) which verifies the credentials of third party applications.

3.2. Related Works

a) Mobile Wallets were proposed by NTT DoCoMo (NTT DoCoMo, 2012) and Google Wallet (Google Wallet, 2011) but there are many flaws in their proposed mobile wallet solutions

   a. NTT DoCoMo‘s Mobile Wallet is implemented in the memory of mobile phone

   b. NTT DoCoMo‘s Mobile Wallet needs a special mobile phone which supports NTT DoCoMo‘s Mobile Wallet.

   c. Digital forensics firm via Forensics has found many flaws in Google Wallet (Block, R., 2011) such as

      i. Significant amount of data is stored unencrypted within Google Wallet.

      ii. While Google Wallet does a decent job by securing all users credit cards numbers (it is not insecurely stored and a PIN is needed to access the cards to authorize payments), the amount of data that Google Wallet stores unencrypted on the device is significant (pretty much everything except the first 12 digits of your credit card) which can lead to social engineering attack.

      iii. Privacy of the client’s is not ensured

b) Authors of (Yannis Labrou et al., 2004) propose a wireless wallet in the memory of mobile phone

c) Authors of (Ernst-Joachim Steffens et al., 2009) propose a SIM based mobile wallet in SIM which cannot be personalized by the bank using OTA.

d) Existing mobile wallet solutions (Yannis Labrou et al., 2004; Ernst-Joachim Steffens et al., 2009; Hao Zhao & Sead Muftic, 2011) store client‘s credentials in the memory of Mobile Phones, on the SIM or UICC, Mobile Phones and SIM with PKI functionality is personalized by the Issuer (Usually by MNSP) and Service Providers like Banks install Mobile Payment Applications with the help of MNSP‘s on the SIM. Mobile Payment Applications cannot be personalized by Banks without the intervention of MNSP‘s.

e) Existing mobile wallet solutions (Yannis Labrou et al., 2004; Ernst-Joachim Steffens et al., 2009; Hao Zhao & Sead Muftic, 2011) do not ensure end to end security in the application layer.

f) Existing mobile wallet solutions (Yannis Labrou et al., 2004; Ernst-Joachim Steffens et al., 2009; Hao Zhao & Sead Muftic, 2011) was not formally verified using manual verification methods like BAN logic nor with any Automated Tools like Scyther (Cremers, C. J. F, 2006; Cremers, C. and Lafourcade, P, 2009) and AVISPA (Armando.A et al., 2005).

CONTRIBUTIONS MADE

a) A protocol is proposed for the personalization of Mobile Payment Application which is in the Mobile Wallet of UICC.

b) A mobile payment protocol (SMWP) is proposed between the personalized Mobile Payment Application which is in the Mobile Wallet of UICC and the Issuing Bank Server.

c) Our protocol (SMWP) is proposed in the UICC of Mobile Device which is considered to be a tamper resistant device so UICC is a Secure Signature Creation Device (SSCD) because the signature processes are performed in the UICC and the private key never leaves the WIM. So non repudiation is ensured.

d) Our proposed mobile payment protocol (SMWP) originating from Mobile Payment Application which is in the Mobile Wallet of UICC to the issuer (I) Server realizes ensures Confidentiality, Authentication, Integrity and Non Repudiation, and prevents double spending, over spending and money laundering. In addition to these SMWP withstands Replay, Man in the Middle, Impersonation attacks and Multiprotocol attack.

e) Our proposed mobile payment protocol is modeled using the High Level Protocol Specification Language (HLPSL) and was verified successfully using AVISPA (Automated Validation of Internet Security Protocols and Applications) tool.

f) Our proposed mobile payment protocol is modeled using the high-level formal language SPDL (Security Protocol Description Language) and were verified successfully using Scyther tool

3.3. Proposed Protocol for the Personalization of Mobile Payment Application in the Mobile Wallet of UICC by the Issuer/Bank

$Step1$: Client gets authenticated by MWMA after entering NRP.

$Step2$: Mobile Wallet Manager Application (MWMA) allows the client to use Mobile Payment Application (MPA) (installed in the MW) after successful authentication of client.

$Step3: MPA \to I: \{MS1, DS_I^C(MS1)\}_{k_I}, Cert_C \quad MS1 = \{PI, phno, NRP, N_C, T_C\}$

/* before initiating the process of personalization client validates bank's certificate using Certificate validation procedure given in (D.R. Stinson, 2006) */

$Step4: I \to MPA: \{MS2, DS_C^I(MS2)\}_{k_C}, cert_I \quad MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$

/* upon receiving the message B checks the authenticity of the message, if the checks are successful then client sends $Step4: I \to MPA: \{MS2, DS_C^I(MS2)\}_{k_C}, cert_I$ */

```
IF Verification (of digital signature on MS1) $DS_I^C(MS1) = TRUE${
        /*Authenticity and Integrity of message MS2 is not compromised*/
        Go to Step 5}
        Else   {      /* Authenticity and Integrity of message MS1 is compromised*/
    Exit}
```

$\{MS1, DS_I^C(MS1)\}_{k_I}, Cert_C$

Issuer decrypts the received message from UICC using his private key and checks the authenticity of $DS_I^C(MS1)$, checks the timestamps and nonce if all the checks are successful then it generates a shared symmetric key K_{CI} between the I and C. Bank sends $\{MS2, DS_C^I(MS2)\}_{k_C}, cert_I$ to UC containing $MS2 = \{PI, phno, K_{CI}, T_I, N_C, N_I, T_C\}$ session keys are generated using hashing algorithms with one bit cyclic shift of a master secret each time a session key is generated as shown in (Supakorn Kungpisdan. et al., 2003). The key set K_{CI} (with {1, 2, 3, .n}) is generated from the secret key K_{CI} and is stored in Mobile Payment Application of the UICC at the client end and in the Issuer server.*/ All the five steps are shown in figure 3.2.

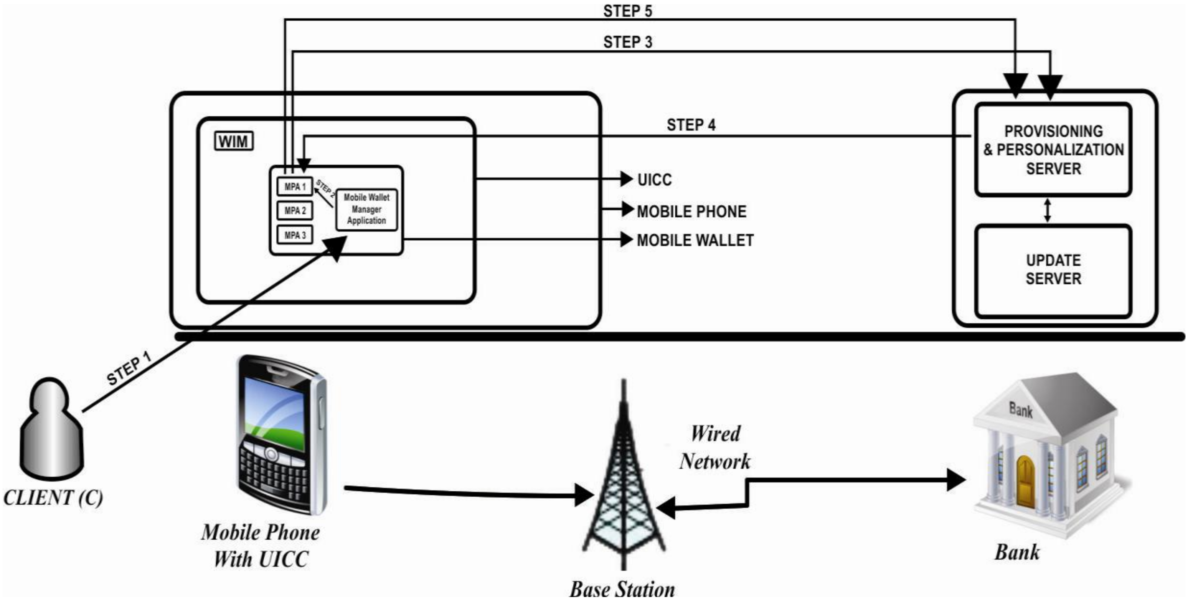


Figure 3.2: Personalization of Mobile Payment Application which is in the Mobile Wallet Application of UICC

$Step5: MPA \rightarrow I : \{MS3, MAC0\}_{K_{ci}}$ Where $MS3 = \{PI, Ack, K_{CI}, N_I, T_I, N_C, T_C\}$

/* B receives $\{MS3, MAC0\}_{K_{ci}}$ from C containing
$MS3 = \{PI, Ack, K_{CI}, N_I, T_I, N_C, T_C\}$ $MAC0 = h(K_{CI}, MS3)$ */

IF ($Verf(MAC0) = TRUE$)

```
{ /*Authentication of/Client, Confidentiality and Integrity of MAC0 are successfully verified*/
    Bank/Issuer maps $K_{CI}$ to PI of Client }
    Else { /* Authentication of/Client, Confidentiality and Integrity of MAC0 are not verified*/
    Exit } }
```

3.4. Proposed Secure Mobile Payment Protocol using Mobile Wallet (SMWP)

We consider a scenario in which a client has a NFC enabled Mobile Phone with UICC as secure element, merchant has NFC enabled POS (Point Of Sale) and all the items (in the store) chosen by client are all tagged with NFC tags. Client chooses some tagged items at a department store, and approaches one of the several NFC enabled POS in the store. The NFC enabled POS scans the tagged items in the shopping cart and generates an invoice. Merchant sends the generated invoice (OI and amount) to client. After successful verification of invoice (OI and amount) client gets authenticated by the mobile wallet application which is on the UICC. Client selects the relevant Mobile Payment Application (MPA) from the mobile wallet application in order to access the payment information (PI) from the Mobile Payment Application (MPA).

Step 0: $C \rightarrow UC : NRP$

Client unlocks the mobile payment application using (NRP) Non-repudiation PIN, which is loaded in the tamper resistant UICC (UC). The client uses Non-repudiation PIN to convince the UC of his identity. All the steps (i.e. from 1 to 10) involved in our proposed SMWP Payment Protocol are shown in figure 3.3.

$Step1 : C \rightarrow M : \{MS4, DS_M^C(MS4)\}_{K_M} \quad MS4 = \{ID_C, T_C, N_C\}$

Client (C) approaches one of the several NFC enabled POS in the store with chosen tagged items at a department store and sends $\{MS4, DS_M^C(MS4)\}_{K_M}$ from his NFC enabled Mobile Phone (with UICC as secure element) to NFC enabled POS (Point Of Sale) containing

$\{ID_C, T_C, N_C\}$.

$Step2: M \rightarrow C: \{MS5, DS_C^M(MS5)\}_{K_C}$

$MS5 = \{OI_M, HOI_M, TID_M, Amt_M, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M\}$

NFC enabled POS scans the tagged items in the shopping cart and generates an OI_M Merchant decrypts $\{MS4, DS_M^C(MS4)\}_{K_M}$ using his private key and gets $\{ID_C, T_C, N_C\}$. Then merchant generates $\{MS5, DS_C^M(MS5)\}_{K_C}$ and containing $\{OI_M, TID_M, Amt_M, LI_M\}$ sends it to Client (C).

$Step3: C \rightarrow M: \{MS6, DS_M^C(MS6)\}_{K_M}$

$MS6 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$

$PI = \{PI, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, LI_C\}$

Client (C) decrypts $\{MS5, DS_C^M(MS5)\}_{K_C}$ using his private key and gets $MS5$. After successful verification and validation client sends $\{MS6, DS_M^C(MS6)\}_{K_M}$ to the merchant.

$Step4: M \rightarrow A: \{MS7, DS_A^M(MS7)\}_{K_A}$

Merchant (M) decrypts $\{MS6, DS_M^C(MS6)\}_{K_M}$ using his private key and gets $MS6$ Merchant sends $\{MS7, DS_A^M(MS7)\}_{K_A}$ to the Acquirer (A). Where

$$MS7 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$$

$Step5: A \rightarrow PG: \{MS8\}$

Where

$MS8 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$

Acquirer decrypts $\{MS7, DS_A^M(MS7)\}_{K_A}$ and gets $MS7$

a) Checks if $HOI_M = HOI_C, TID_M = TID_C, Amt_M = Amt_C, LI_M = LI_C$

b) Checks if Timestamps $T_C = T_M$

c) Checks if nonces $N_C = N_M$

If all the checks are found to be successful then it keeps a copy of the received message MS7 and authorizes the Order Information. Then Acquirer forwards $MS8$ message to the PG via Secure Private Banking Network.

$$MS8 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$$

$$Step6: PG \to I: \{MS9\}$$
$$MS9 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$$

PG receives $MS8$ from the Acquirer through Private Banking Network which is very secure. Payment Gateway (PG) will perform the following verifications from the $MS8$ it has received.

a) Checks if $HOI_M = HOI_C, TID_M = TID_C, Amt_M = Amt_C, LI_M = LI_C$

b) Checks if Timestamps $T_C = T_M$

c) Checks if nonces $N_C = N_M$

if all the checks are found to be successful then it keeps a copy of the received message $MS8$ and forwards $MS9$ to the Issuer (I).

$$Step7: I \to PG: \{MS10\}$$
$$MS10 = AuthPI$$

Issuer receives $MS9$ from the PG through Private Banking Network which is very secure. Issuer (I) will perform the following checks from $MS9$ it has received. Where

$$PI = \{AI, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, LI_C\}$$

a) Decrypts the PI using the symmetric key shared between Issuer and Client

b) Checks the clients account for sufficient funds

c) Checks if $HOI_M = HOI_C, TID_M = TID_C, Amt_M = Amt_C, LI_M = LI_C$

d) Checks if Timestamps $T_C = T_M$

e) Checks if nonces $N_C = N_M$

If all the checks are successful it authorizes the PI and sends $MS10$ to PG

$$Step8: I \to C: \{MS11, DS_C^I(MS11)\}_{K_C}$$

$MS11 = \{TID, Amt, ID_M, Success / Failure\}$

Issuer (I) informs Client (C) about the success/failure of the transaction

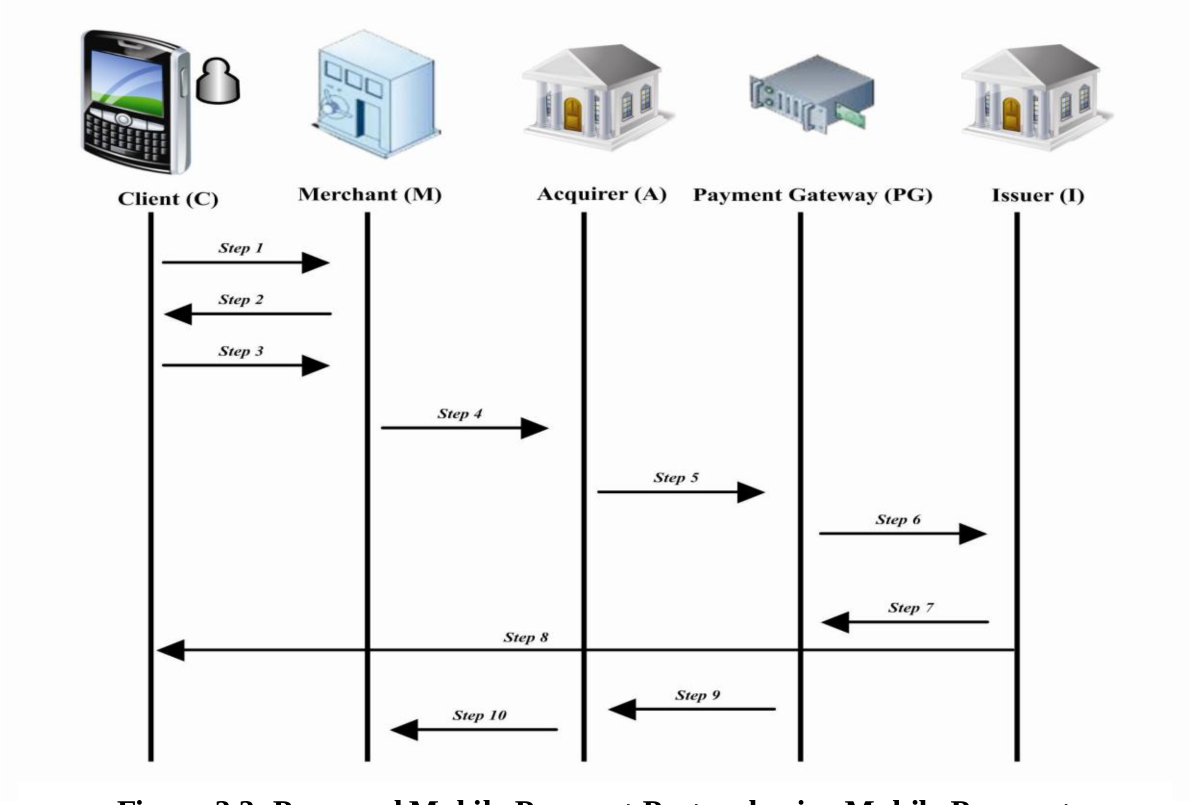


Figure 3.3: Proposed Mobile Payment Protocol using Mobile Payment Application personalized in MW Application

$Step9 : PG \rightarrow A : \{MS12\}$

$MS12 = \{TID, Amt, ID_M, Success / Failure\}$

PG informs Acquirer (A) about the success/failure of the transaction.

$Step10 : A \rightarrow M : \{MS13, DS_M^A(MS13)\}_{K_M}$

$MS13 = \{TID, Amt, ID_M, Success / Failure\}$

A informs merchant (M) about the success/failure of the transaction

3.5. Security Analysis of SMWP

We consider the goals defined in section 1.6 of Chapter 1 and perform the security analysis as below.

3.5.1. Data Security

Goal 1 (Third party):

All the entities involved in the SMWP protocol store their credentials (Private Keys, NRP and Certificates) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the SMWP protocol will exchange messages using encryption and digital signature so any third entity will not be able to gain access to participant's transactional data thereby achieving Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. The Mobile Signature of data provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore if the mobile signature was generated by means of SSCD it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (SMWP) withstands the following attacks

Replay Attacks: Replay attack is avoided by including Timestamps, Nonce, TID and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included in the messages exchanged ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C's private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating digital signature is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Master key for the generation of symmetric key is stored in the mobile payment application of the Mobile Wallet which is protected by NRP. Our proposed protocol SMWP withstands this attack because the intruder (In) does not have receiver’s private key.

Multiprotocol Attack: Our proposed protocol SMWP withstands this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2 (Secrecy):

In our proposed Mobile Payment protocol (SMWP) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I), merchant will not be able to decrypt Payment Information (PI). Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)). PG will not know about OI and PI thereby achieving Transaction privacy from PG. Eavesdropper cannot get OI and PI because the messages are hashed and encrypted thereby achieving Transaction privacy from Eavesdropper.

Goal 3 (Uniqueness):

In our proposed Mobile Payment protocol (SMWP) every transaction is unique. The uniqueness is achieved due to the generation of the fresh Transaction id (TID) by the merchant, and its verification by the bank. Every transaction is also linked to TID, Timestamps, Nonce, Location Information generated by the Client and Merchant and symmetric keys shared between Client and Issuer used for the encryption of PI.

3.5.2. Client Security

Goal 4 (Authentication):

Our proposed Mobile Payment Protocol (SMWP) uses certifying authority to certify the authenticity of public keys held by the Client, Merchant and Payment Gateway. The client’s conversations are only with the Merchant. Client, Merchant and Issuer’s authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5 (Authorization):

In our proposed Mobile Payment Protocol (SMWP) the client obtains un forgeable proof of transaction authorization from the Issuer at the end of protocol (i.e. whether the transaction is success or failure).

Goal 6 (Identity Protection from Merchant and Eavesdropper):

Client will be issued an anonymous identity $Anonid_C$ by the CA after successful verification of client's credentials. Client's certificate will have anonymous identity instead of his/her real identity thereby achieving identity Protection from Merchant and Eavesdropper.

3.5.3. Bank Security (Issuer, Acquirer and PG)

Goal 7(Authentication):

Our proposed Mobile Payment Protocol (SMWP) has certifying authority to certify the authenticity of the public keys held by the Client, Merchant and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer share a symmetric key between them.

Goal 8 (Authorization):

The Issuer obtains an authorization proof for transaction from the client in the form of PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer which contains payment details, merchant identity and hashed OI. The Merchant obtains OI and PI (Payment Information) encrypted with the shared symmetric key and forwards this information to Acquirer after successful verification of digital signature on the message, OI and amount. Merchant sends the hashed value of OI, which is verified by the Issuer, so Issuer authorizes the transaction after obtaining unforgeable proof from the Client and Merchant.

Goal 9(Prevents double spending, overspending and money laundering):

Issuer (I) keeps the message it has received from PG in its archives. If the client or merchant tries to double spend the PI, I can detect this from the TID, Timestamp, Nonce, Location Information and symmetric key shared between Client and Issuer. So double spending is avoided in SMWP by Issuer (I). If the client or merchant tries to overspend, I avoids them in doing so since it checks client's balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Banks (I, A, PG) are always involved in every payment transaction thereby preventing money laundering.

Goal 10(Issuer, Acquirer and PG turning malicious):

If any one or all of the entities in the private banking network (Issuer, Acquirer and PG) turns malicious then also they will not succeed in performing transaction on behalf of Client (C) because they have no knowledge about the private key of the Client (C).

3.5.4. Merchant Security

Goal 11(Authentication):

Our proposed Payment Protocol (SMWP) uses certification authorities, to certify the authenticity of public keys held by the Client, Acquirer, Issuer and PG.

Goal 12 (Authorization):

Merchant checks the authenticity and integrity of the messages received from C, validates digital signatures, location Information, checks timestamps and nonce. If the Merchant is convinced about the TID and OI then only M authorizes the OI thereby achieving order secrecy OI is not known to any of the engaging entities other than C and M. So Merchant authorizes the transaction after obtaining unforgeable proof from the Client.

3.6. Comparative Analysis of SMWP with Related Works

Protocols Features	Google Wallet 2011	NTT DoCoMo 2012	Yannis Labrou et al., 2004	Ernst-Joachim Steffens et al., 2009	Hao Zhao & Sead Muftic, 2011	SMWP
Authentication	Yes	Yes	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes
Non- Repudiation	Yes	Yes	Yes	Yes	Yes	Yes
Client's credentials are generated using OBKG procedure	No	No	No	No	No	Yes
WPKI is implemented in the memory of Mobile Phone	No	No	No	No	No	Yes
Ensures end to end Security at application level	No	No	No	No	No	Yes
Identity Protection from Eavesdropper	No	No	No	No	No	Yes
Transaction Privacy Protection from Eavesdropper	No	No	No	No	No	Yes
Transaction Privacy Protection from PG	No	No	No	No	No	Yes
Prevents Double Spending, Over Spending and Money Laundering	NR	NR	NR	NR	NR	Yes
Withstands Replay Attack	Yes	Yes	No	No	No	Yes
Withstands Impersonation Attack	Yes	Yes	No	No	No	Yes
Withstands MITM Attack	Yes	Yes	No	No	No	Yes
Withstands Multi Protocol Attack	No	No	No	No	No	Yes
Formal Verification using BAN Logic, AVISPA & SCYTHER TOOL	No	No	No	No	No	Yes

Table 3.1: Comparative Analysis of SMWP Protocol with the Related Works

3.7. Formal Verification of SMWP

3.7.1. Formal Verification of SMWP using BAN Logic

$Step1: C \rightarrow M : \{MS4, DS_M^C(MS4)\}_{K_M}, Cert_C \qquad MS4 = \{ID_C, T_C, N_C\}$

$Step2: M \rightarrow C: \{MS5, DS_C^M(MS5)\}_{K_C}$

$MS5 = \{OI_M, HOI_M, TID_M, Amt_M, ID_C, T_C, N_C, ID_M, T_M, N_M .LI_M\}$

Proof of Step 1

If the certificate validation is successful & from the assumptions given in **AS1, AS2 & AS6** we can conclude that M **believes C said** $Cert_C$ ……..statement (1)

M receives $\{MS4, DS_M^C(MS4)\}_{K_M}$ from C and decrypts the message using its private key from the assumptions **AS1& AS2** and can conclude that

M **believes** $\{MS4, DS_M^C(MS4)\}$ ..statement (2)

From **AS5 & AS7** we conclude “M **believes fresh** (T_C)”…statement (3).

“M **believes fresh** (N_C)”………..Statement (4)

From statements 1 and 4

M **believes C said** $\{MS4, DS_M^C(MS4)\}_{K_M}, Cert_C$ ………..statement (5)

$Step3: C \rightarrow M : \{MS6, DS_M^C(MS6)\}_{K_M}$

$MS6 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$

$PI = \{PI, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, LI_C\}$

Proof of Step 2

If the certificate validation is successful & from the assumptions given in **AS1, AS2 & AS6** we can conclude that C **believes M said** $Cert_M$ ……..statement (6)

C receives $\{MS5, DS_C^M(MS5)\}_{K_C}$ from Merchant (M) and decrypts the message using his private key and gets MS5 so from the assumptions **AS1, AS2, AS6**

C **believes** $\{MS5, DS_C^M(MS5)\}$ ……..statement (7)

From **AS5 & AS7** we conclude **C believes fresh** (T_m)…statement (8).

C believes fresh (N_m)..Statement (9)

From statements 6 to 9

C believes M said $\{MS5, DS_C^M(MS5)\}_{K_C}$ ………..statement (10)

$Step4: M \rightarrow A: \{MS7, DS_A^M(MS7)\}_{K_A}$

$Step4: M \rightarrow A: \{MS7, DS_A^M(MS7)\}_{K_A}$

$MS7 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, T_C, N_C, ID_M, T_M, N_M, LI_M, LI_C, (PI)_{K_{CI}}\}$

Proof of Step 3

M receives $\{MS6, DS_M^C(MS6)\}_{K_M}$ from C and decrypts the message using his private key and gets $\{MS6, DS_M^C(MS6)\}$ from the assumptions **AS1, AS2, AS6**

M **believes** $\{MS6, DS_M^C(MS6)\}$ ……..statement (11)

From **AS5 & AS7** we conclude M **believes fresh** (T_C)…statement (12)

M believes fresh (N_C)………..Statement (13)

From statements 11 and 13

M **believes C said** $\{MS6, DS_M^C(MS6)\}_{K_M}$ ………..statement (14)

$Step8: I \to C: \{MS11, DS_C^I(MS11)\}_{K_C}$ $MS11 = \{TID, Amt, ID_M, Success / Failure\}$

Proof of Step 4

A receives $[\{MS7, DS_A^M(MS7)\}_{K_A}, Cert_M]$ from M

A validates certificate of the merchant M, If the certificate validation is successful & from the assumptions given in **AS1, AS2 & AS6** we

A **believes M said** $Cert_M$ ………..statement (15)

M decrypts the message using his private key and gets$\{MS7, DS_A^M(MS7)\}, Cert_M$from the assumptions **AS1, AS2, AS6**

A **believes** $\{MS7, DS_A^M(MS7)\}_{K_A}$ ………..statement (16)

From **AS5 & AS7** and statement (16) we conclude

“A **believes fresh** (T_M)”…statement (17).

“A **believes fresh** (N_M)”………..Statement (18)

From **AS8 & AS9 &** From statements 15 and 18

A **believes M said** $[\{MS7, DS_A^M(MS7)\}_{K_A}, Cert_M]$ ………..statement (19)

Messages exchanged from step 5 to step 7 are transmitted through secure private banking network so we are not verifying these steps.

Proof of Step 6

Step 6 is sent through secure private banking network Issuer successfully checks PI using the assumptions **(AS4, AS10, AS11 & AS12)** given in Appendix B.

$Step10: A \rightarrow M\ :\{MS13, DS_M^A(MS13)\}_{K_M}$

Proof of Step 8

Client (C) receives $\{MS11, DS_C^I(MS5)\}_{K_C}$ from Issuer (I) and decrypts the message using his private key and gets $\{MS11, DS_C^I(MS5)\}$ so from the assumptions **AS1, AS2, AS6**

C **believes** $\{MS11, DS_C^I(MS5)\}$ ..statement (20)

C validates certificate of the Issuer, If the certificate validation is successful then

C **believes I said** $Cert_I$ ………..statement (21)

From **AS5 & AS7** we conclude C **believes fresh** (T_I)…..statement (22).

C believes fresh (N_I)....Statement (23)

From statements 20 to 23

C **believes I said** $\{MS11, DS_C^I(MS5)\}, Cert_I$ ……..statement (24)

Proof of Step 10

Merchant (M) receives $\{MS13, DS_M^A(MS13)\}_{K_M}$ from Acquirer (A) and decrypts the message using his private key and gets $\{MS13, DS_M^A(MS13)\}$ so from the assumptions **AS1, AS2, AS6**

M **believes** $\{MS13, DS_M^A(MS13)\}$ ………..statement (25)

M validates certificate of the A, if the certificate validation is successful then

M **believes A said** $Cert_A$ ………..statement (26) From **AS5 & AS7** we conclude

M **believes fresh** (T_A)…statement (27)

M believes fresh (N_A)……..Statement (28)

From statements 25 to 28

M **believes A said** $\{MS13, DS_M^A(MS13)\}_{K_M}, Cert_A$ ……….statement (29)

3.7.2. Formal Verification of SMWP using AVISPA Tool

The validation of our proposed protocol SMWP using the OFMC back-end of AVISPA Tool is given in Appendix C as figure 3.4
The validation of our proposed protocol SMWP using the CL-AtSe back-end of AVISPA Tool is given in Appendix C as figure 3.5

3.7.3. Formal Verification of SMWP using SCYTHER Tool

Result of SMWP using 'Verification Claim' Procedure of Scyther Tool is given in Appendix C as figure 3.6
Result of SMWP using 'Automatic Claim' Procedure of Scyther Tool is given in Appendix C as figure 3.7

3.8. Conclusions

This chapter proposes a protocol for the personalization of Mobile Payment Application of Mobile Wallet which is in UICC (Universal Integrated Circuit Card). A mobile payment protocol (SMWP) is proposed which ensures end to end security. Our proposed mobile payment protocol (SMWP) originating from Mobile wallet (which is in the UICC) to the Issuing bank Server ensures Confidentiality, Authentication, Integrity and Non Repudiation, prevents double spending, over spending and money laundering, and withstands replay, Man in the Middle (MITM), Impersonation attacks and Multi Protocol attacks. Proposed mobile payment protocol is formally verified using BAN Logic, AVISPA and Scyther Tool. We observe that there should be collaboration (i.e. understanding) among all the entities involved in the protocol (i.e. among UICC manufacturers, CA, MNO and Banks). There should be a regulatory authority to look into the functioning of each of these entities involved in the protocol.

Chapter 4: Mobile Payment Protocol based on Traveler's Check

4.1. Introduction

In this chapter we propose a new payment instrument (Mobile Traveler's Check (MTC)) in the realm of mobile commerce. It provides the merits of both e-cash and e-check i.e. like e-cash we can spend MTC anywhere, like e-check if MTC is lost then Issuer will issue new MTC (Hsien et al., 2001).

4.2. Related Work

Clients purchase travelers checks for the purpose of obtaining local or foreign currency and, to a lesser extent, making purchases of goods and services directly with the checks, while traveling abroad. Hsien et al. (2001) proposed an electronic traveler's check system for wired networks and Horng-Twu et al.(2007) proposed, "A new electronic traveler's Check Scheme based on one-way hash function". This scheme is based on utilization of one-way hash function with the properties of security and efficiency.

Gaps found in the Literature

a) Hsien et al. (2001) and Horng-Twu et al. (2007) proposed Traveler's Check for wired Environment.

b) Traveler's Check was not proposed in the realm of Mobile Commerce.

c) Traveler's Check proposed by Hsien et al., (2001) and Horng-Twu et al. (2007) does not ensure non repudiation property which is very important for financial transactions.Traveler's Check proposed by Hsien et al. (2001) and Horng-Twu et al. (2007) does not ensure Anonymity, Dispute Resolution, Order & Payment Secrecy properties which are very important for Mobile Payments.

d) Formal verification was not done for the protocols proposed by Hsien et al., (2001) and Horng-Twu et al. (2007).

Contributions made

a) Proposed a new payment instrument (MTC) in the realm of Mobile Commerce which achieves Resistance to forgery, avoids double spending, Overspending and Money laundering.

b) Proposed payment instrument (MTC) is used by Specific user, Re-issued if MTC is lost, achieves Privacy of the User, and ensures Order Secrecy, Payment secrecy, Order instruction Authorization, Payment instruction Authorization & Dispute Resolution.

c) Proposed protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway.

d) Proposed protocol can withstand Replay, Man in the Middle, Impersonation and Multi Protocol attacks.

e) Proposed protocol is formally verified using BAN logic, AVISPA & Scyther tools.

4.3. Proposed Mobile Payment Protocol based on Traveler's Check

Security requirements of Mobile Traveler's Check (MTC)

Resistance to forgery: MTC should be issued to the Client (C) by the Issuer (I) MTC should prevent malicious users or merchants from forging it.

Specific user: MTC must be signed by both issuer (**I**) and the Client (C). It should include the identification information of the issuer and user and only the specific owner of the MTC should be able to use it.

Re-issuing: When MTC is lost the Client (C) should use serial number (SNO) and endorsement of Client (C) to report the loss of MTC. Then the issuer (I) will issue a new MTC after checking MTC in its database.

There are four phases in our proposed protocol they are (i) Registration phase (ii) Issuance phase (iii) Payment phase and (iv) Deposit phase

Registration Phase

We assume that this phase is a onetime process this chapter also follows the same procedure as given in chapter 2. In order to achieve anonymity Client (C) sends $\{PNO, anonid, PI\}_{K_{CI}}$ to the Issuer (I) online i.e. $Anonid_C = PNO + h(PI, NRP)$

After receiving the message Issuer (I) checks the PNO with PI in its database if it matches then it adds $anonid$ as identity for the future transactions. Since $Anonid_C$ is only known to Client (C) and issuer (I), merchant (M) cannot map $Anonid_C$ with Client's (C) real identity.

Issuance Phase

In this phase, Client (C) requests Issuer (I) to issue MTC. The main steps involved in the Issuance Phase of our proposed protocol are as follows:

Step 1: $C \rightarrow UC : NRP$

Client (C) unlocks the payment application software using (NRP) Non-repudiation PIN, which is stored in the tamper resistant UICC. The client uses Non-repudiation PIN to convince the UICC (UC) of his identity.

Step 2: $C \rightarrow I : \{MS1, DS_I^C\}_{K_{CI}}, Cert_C$

The Client (C) uses his anonymous identity $Anonid_C$ to buy MTC from the Issuer (I) for amount (Amt). The Client (C) would choose a random number R_{C1}, then it performs exclusive OR operation with $Anonid_C$ and operates one-way hash function h (.) to generate a client endorsement $ENDOR_C$. It sends only $ENDOR_C$ to merchant (M) not $Anonid_C$.

$$\begin{aligned}
&\{MS1, DS_I^C\}_{K_{CI}} \\
&MS1 = (ENDOR_C, T_C^{'}, M, Amt, N_C, Anonid_C, R_{C1}), \\
&ENDOR_C = h(Anonid_C \oplus R_{C1}), \\
&DS_I^C = (MS1)_{K_C^{-1}}
\end{aligned}$$

$ENDOR_C$ is used as an identity of the Client (C) only for this transaction because R_{C1} will be different for every new transaction so merchant cannot link or trace the real identity of Client (C). Client (C) requests Issuer (I) to issue MTC using MS1. It sends R_{C1} to Issuer (I) to check $ENDOR_C$ and to inform $ENDOR_C$ is valid only for the current transaction.

Step 3: $I \rightarrow C : (MS2, DS_C^I)_{K_{CI}}, Cert_I$

Issuer (I) uses **Check ( )** algorithm to validate the message $\{MS1, DS_I^C\}_{K_{CI}}$ received from client if the validation is successful it accepts MS1.Then Issuer (I) chooses another Random Number R_{I1} and computes an exclusive-OR operation on the Private Key of the Issuer (I) and One- way hash function h (.) is applied on the outcome ($Iden_I = h(R_{I1} \oplus K_I^{-1})$) to generate the identity of the Issuer (I). Issuer (I) sends $\{MS2, DS_C^I\}_{K_{CI}}, cert_I$ to Client (C). $MS2 = (ENDOR_C, Iden_I, N_I, N_C, T_I^{'})$ & $DS_C^I = (MS2)_{K_I^{-1}}$

Step 4: $C \rightarrow I : \{MS3, DS_I^C\}_{K_{CI}}, Cert_C$

Device uses **Check ( )** algorithm to check the security properties of the message $MS2$ if the checking is successful UC accepts MS2. Client then calculates Payment Requirement (PR) $PR = h(ENDOR_C \oplus Amt \oplus Iden_I \oplus FV)$ $MS3$ is constructed by the UC where $MS3 = \{PR, Amt, FV, N_I, N_C, T_C\}$ then $\{MS3, DS_I^C)_{K_{CI}}, Cert_C$ is sent to the Issuer (I) where $DS_I^C = (MS3)_{K_C^{-1}}$.

Step 5: $I \rightarrow PG : (MTC, SNO)$

$I \rightarrow C : (MS4, DS_C^I)_{K_{CI}}, cert_I$

Issuer (I) uses **Check ( )** algorithm to check the security properties of the message $\{MS3, DS_I^C)_{K_{CI}}$ if the checking is successful Issuer (I) accepts $MS3$. Issuer (I) recalculates

Payment Requirement $PR^{'} = h(ENDOR_C \oplus Amt \oplus Iden_I \oplus FV)$ if $(PR^{'} = PR)$ then MTC is constructed which contains $MTC = (FV \oplus ENDOR_C \oplus Iden_I \oplus Amt \,||\, SNO)$. MTC is sent to the Payment Gateway (PG) for recording and safekeeping using Private Banking Network, which is secure and $(MS4, DS_C^I)_{K_{CI}}, Cert_I$, is sent to the Client (C).

$$MS4 = (MTC, SNO, N_C, N_I, T_I^{'}),$$
$$DS_C^I = (MS4)_{K_I^{-1}}$$

Step 6: $UC \to C : (MTC, SNO)$

Client (C) uses **Check ( )** algorithm to check the security properties of the message $(MS4, DS_C^I)_{SYYKEY_{IC}}$ if the checking is successful Client accepts $MS4$. Device recalculates $MTC^{'} = (FV \oplus ENDOR_C \oplus Iden_I \oplus Amt \,||\, SNO)$ if $(MTC = MTC^{'})$ then Client (C) accepts MTC and displays it to the Client (C) through secure channel i.e. on the output display of the Client (C). MTC is stored in the tamper resistant UICC of the mobile Client (C).

Payment Phase and Deposit Phase

Before starting the payment phase Client (C) has to receive the order instruction (OI) from merchant. A prologue, whose execution precedes the payment and deposit phase, is given below.

Prologue

The prologue consists of the following steps

i) Merchant (M) generates a random number R_{M1} and then asks the acquirer (A) for Unique Transaction Identifier (UTI_A).

ii) After receiving the request acquirer (A) generates UTI_A, inserts it in its database and returns UTI_A to the merchant (M).

iii) After receiving UTI_A the merchant (M) sends $(UTI_A, R_{M1}, OI, Cert_A, amt)$ to Client (C).

iv) After receiving $(UTI_A, R_{M1}, OI, Cert_A, Amt)$, Client (C) verifies and ascertains that they describe the agreed purchase.

Payment Phase:

Step 7: $C \rightarrow M : \{MS5, DS_M^C\}_{K_M}, cert_C$

$MS5 = (PI, Endor_C, n_C, HOI_C, cert_I, UTI_A, T_C^{'})\}$

Payment Instruction (PI) $PI = (Endor_C, (MTC, SNO), HOI_C, Amt, UTI_A)_{K_{CI}}$

$$DS_A^C = (MS5)_{K_C^{-1}} \quad (MS5, DS_M^C)_{K_M}$$

Client (C) constructs Payment Instruction (PI), Order Instruction (OI) i.e. (HOI_C) and sends it to the Acquirer (A) via Merchant (M).

Deposit Phase

Step 8: $M \rightarrow A : \{(MS5, MS6), DS_A^M\}_{K_{MA}}, cert_C, cert_M$

Merchant (M) uses **Check ( )** algorithm to check the authenticity of the message $(MS5, DS_M^C)_{K_M}$ if the checking is successful then Merchant (M) accepts $MS5$. Merchant (M) constructs another message MS6 containing merchants identity (M), hashed order instructions generated by merchant (HOI_M), nonce generated by merchant (N_M), Amount (Amt), and Unique Transaction Identifier (UTI_A) generated by Acquirer (A) and Timestamp of merchant (T_M) and sends $\{MS5, MS6\}$ to the Acquirer (A).

$MS6 = \{M, HOI_M, N_M, Amt, UTI_A, T_M^{'}\}$,

$DS_A^M = (MS5, MS6)_{K_M^{-1}}$, $\{(MS5, MS6), DS_A^M\}_{K_{MA}}$

Step 9: Under Private Banking Network (Secure Channel)

(a) $A \to PG$: $(MS5, MS6)$

After receiving $\{(MS5, MS6), DS_A^M\}_{K_{MA}}$ from merchant (M), Acquirer (A) uses **Check ( )** algorithm to validate the network security properties, timeliness and freshness of the received message. If the validation is successful then Acquirer (A) accepts $(MS5, MS6)$. Acquirer (A) verifies the hashed order instructions (OI) generated by Client (C) (HOI_C) and merchant (M) (HOI_M) in order to ascertain that merchant (M) and Client (C) agree on the same purchase. If ($HOI_C = HOI_M$) verification of OI is successful. Then Acquirer (A) authorizes order instructions (OI) and sends $(MS5, MS6)$ to PG through secure private banking network.

b) $PG \to I$: $(MS5, MS6)$

After receiving the message $(MS5, MS6)$ from Acquirer (A) PG forwards the message $(MS5, MS6)$ to Issuer (I) keeping a copy of HOI_C, HOI_M, UTI_A attributes in its database.

(c) $I \to PG : UTI_A$, $Depositoffunds$

After receiving $(MS5, MS6)$ message from 'PG', Issuer (I) performs the following checks

(i) Decrypts the PI with the symmetric key shared between 'I' and 'C'.

(ii) Verifies the amount (Amt) on (MTC) to the amount sent by 'M' if both matches then issuer (I) concludes 'M' & 'C' agree on the same price.

(iii) Verifies whether (MTC, SNO) was spent before from the 'SNO' of the 'MTC' in its database. If all the checks are successful then Issuer (I) authorizes Payment Instructions (PI). Thus payment secrecy and payment authorization are achieved. Issuer (I) then deposits funds into M's account and informs 'PG' of the transfer of funds and sends (MTC, SNO), UTI_A, HOI_M and HOI_C through secure private banking network for recording and safekeeping because 'PG' acts as an Arbitrator. In case of disputes the connection between (MTC, SNO), HOI_M, HOI_C and UTI_A is established.

The combination of attributes (i.e. UTI_A, (MTC, SNO), HOI_M and HOI_C) are unique for each payment transaction. Thus dispute resolution is also achieved.

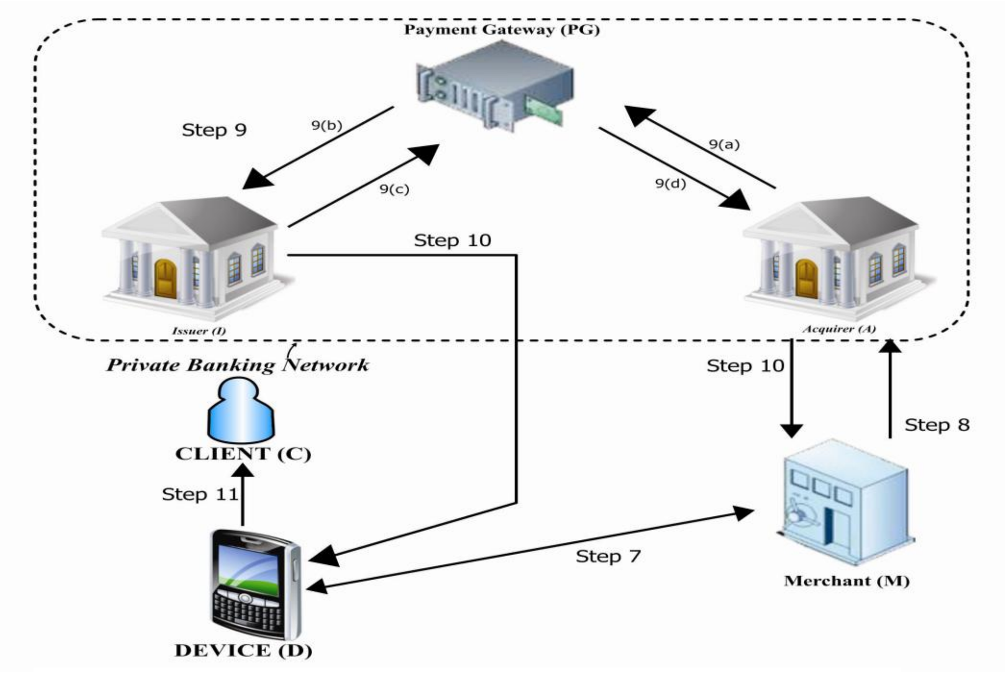


Figure 4.1: Payment and Deposit Phase of MTC

(d) $PG \rightarrow A : UTI_A, Depositoffunds, (MTC, SNO), HOI_M$ & HOI_C

After receiving response from **I, PG** informs **A** of the transfer funds into **M's** account

Step 10: $A \rightarrow M : (Intimation, UTI_A)_{K_{AM}}$, $I \rightarrow C : (Intimation, UTI_A)_{K_{CI}}$

Acquirer (A) sends $(Intimation, UTI_A)_{K_{AM}}$ to Merchant (M)

$I \rightarrow C : (Intimation, UTI_A)_{K_{CI}}$

Issuer (I) sends $I \rightarrow C : (Intimation, UTI_A)_{K_{CI}}$ to UC

Step 11: $UC \rightarrow C$: Intimation, UTI_A

The UC displays "Intimation, UTI_A" message on its screen to the Client (C).

Algorithm Check ( )

Step (a):

$$X \rightarrow Y : \{MS, DS_Y^X\}_{K_{XY}}$$

/*the receiver of the message first decrypts the message using DES decryption algorithm then verifies digital signature using ECDSA verification algorithm, here **verification ( )** function is used to verify DS_Y^X using ECDSA.*/

If **verification (DS_Y^X) = TRUE**

then integrity of the message 'M' is intact and go to **Step (b)**

else integrity of the message 'M' was compromised so reject the message.

Step (b):

Checks timestamps and nonce (which are in the message) if the timestamps and nonce are within the range then timeliness and freshness of the message is ensured so **Check ( )** is successful and receiver accepts the message.

Data Structures of the messages in ASN.1

The size or length of the digital signature output using SHA-1 is equal to the size of the signature key. Since ECDSA is used for generating and verifying the digital signatures so the size or the length of the digital signature is 163 bits i.e. 21 bytes.

1) **Digital Signature using ECDSA (DS_Y^X)**

```
ECDSA-Sig-Value: = SEQUENCE {

        r                           INTEGER,

        s                           INTEGER

   }
```

ECDSA-Sig-Value is a Structure (SEQUENCE) which has two components 'r' and's' whose data types are INTEGER. The digital signature generated (i.e. ECDSA-Sig-Value) has "Bit String" data type and size of the digital signature is 163 bits.

Endorsement ($ENDOR_C$)

```
Endorsement: = SEQUENCE {

        phone number                Numeric String (SIZE(13)),

        account number              Alphanumeric (SIZE(20)),

        non repudiation PIN         Alphanumeric(SIZE(10)),

        random number               Numeric String (SIZE(12))

   }
```

Identity of Issuer ($Iden_X$)

```
Identity of Issuer: = SEQUENCE {

        private key of issuer              Bit String (SIZE(20)),

        random number                  Numeric String (SIZE(12))      }
```

Anonymous Identity (anonid)

```
Anonymous Identity: = SEQUENCE {

        phone number                       Numeric String (SIZE(13)),

        account number                     Alphanumeric (SIZE(20)),

        non repudiation PIN (NRP)          Alphanumeric(SIZE(10))          }
```

Mobile Traveler's Check (MTC)

```
Mobile Traveler Check: = SEQUENCE {

        face value                  Numeric String (SIZE(15)),

        endorsement                 Endorsement ,

        identity of issuer          Identity of issuer,

        amount                      Numeric String (SIZE(20)),
     serial number                  Numeric String (SIZE(20))     }
```

Payment Requirement (PR)

```
Payment Requirement: = SEQUENCE {

     Facevalue                          Numeric String (SIZE (15)),

     endorsement                        Endorsement ,

    identity of issuer                  Identity of issuer,

     amount                             Numeric String (SIZE(20))   }
```

4.4. Security Analysis

We consider the goals defined in section 1.6 of Chapter 1 and perform the security analysis as below.

4.4.1. Payment Instrument (MTC) Security:

Resistance to Forgery: If an intruder wants to forge Issuer's identity he should know the private key of the Issuer (I) and the random number generated by the Issuer (I) for this transaction i.e. issuer's identity is $Iden_I = h(R_{I1} \oplus K_I^{-1})$. It is impossible to get the private key and random number of the Issuer (I) so intruder cannot forge MTC. Therefore "Resistance to Forgery" which is a security requirement of MTC is achieved.

Specific User: When a Client (C) requests Issuer (I) to issue MTC he should prove his credentials (i.e. User has to send message containing his $Anonid_C$ and DS_I^C the message should be encrypted using symmetric key shared between C and I). After ascertaining the authenticity and integrity of the message Issuer (I) issues a new MTC. So MTC is issued for a specific user only. Therefore "Specific User" which is a security requirement of MTC is achieved.

Re-issuing: If MTC is lost, then Client (C) will use SNO of MTC and C's endorsement $ENDOR_C$ to report the loss of his MTC to the issuer (I). Then **I** checks SNO and $ENDOR_C$ in its database. If it is not spent then **I** will reissue a new MTC. Therefore "Reissuing" which a security requirement is of MTC is achieved.

4.4.2. Data Security

Goal 1 (Third Party):

All the entities involved in the MTC protocol store their credentials (Private Keys, NRP and Certificates) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the MTC protocol will exchange messages using encryption and digital signature so any third entity will not be able to gain access to participant's transactional data thereby achieving Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. The Mobile Signature of data provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore if the mobile signature was generated by means of SSCD it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (MTC) withstands the following attacks

Replay Attacks: Replay attack is avoided by including Timestamps, Nonce, TID and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included in the messages exchanged ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C's private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating digital signature is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Our proposed protocol MTC withstands this attack because the intruder (In) does not have receiver's private key.

Multiprotocol Attack: Our proposed protocols in this chapter withstand this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2 (Secrecy):

In our proposed Mobile Payment protocol MTC Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)). PG will not know about OI and PI thereby achieving Transaction privacy from PG. Eavesdropper cannot get OI and PI because the messages are hashed and encrypted thereby achieving Transaction privacy from Eavesdropper.

Goal 3 (Uniqueness)

In our proposed Mobile Payment protocol MTC every transaction is unique. The uniqueness is achieved from TID, SNO of MTC, Timestamps and Nonce.

4.4.3. Client Security

Goal 4 (Authentication):

Our proposed Mobile Payment Protocol MTC uses certifying authority to certify the authenticity of public keys held by the Client, Merchant and Issuer, Acquirer and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5 (Authorization):

In our proposed Mobile Payment Protocol MTC the client obtains un forgeable proof of transaction authorization from the Issuer at the end of protocol (i.e. whether the transaction is success or failure).

Goal 6 (Identity Protection from Merchant and Eavesdropper):

Client will be issued an anonymous identity $Anonid_C$ by the CA after successful verification of client's credentials. Client's certificate will have anonymous identity instead of his/her real identity thereby achieving identity Protection from Merchant and Eavesdropper.

4.4.4. Bank Security (Issuer, Acquirer and PG)

Goal 7(Authentication):

Our proposed Mobile Payment Protocol MTC has certifying authority to certify the authenticity of the public keys held by the Client, Merchant and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer share a symmetric key between them.

Goal 8 (Authorization):

The Issuer obtains an authorization proof for transaction from the client in the form of PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer which contains payment details, merchant identity and hashed OI. The Merchant obtains OI and PI (Payment Information) encrypted with the shared symmetric key and forwards this information to Acquirer after successful verification of digital signature on the message, OI and amount. Merchant sends the hashed value of OI, which is verified by the Issuer, so Issuer authorizes the transaction after obtaining unforgeable proof from the Client and Merchant.

Goal 9(Prevents double spending, overspending and money laundering):

Issuer (I) keeps the message it has received from PG in its archives. If the client or merchant tries to double spend the PI, I can detect this from the TID, timestamp and nonce. So double spending is avoided in MTC by Issuer (I). If the client or merchant tries to overspend, I avoids them in doing so since it checks Client's balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Banks (I, A, PG) are always involved in every payment transaction thereby preventing money laundering.

Goal 10(Issuer, Acquirer and PG turning malicious):

If any one or all of the entities in the private banking network (Issuer, Acquirer and PG) turns malicious then also they will not succeed in performing transaction on behalf of Client (C) because they have no knowledge about the private key of the Client (C).

4.4.5. Merchant Security

Goal 11(Authentication):

Our proposed Payment Protocol MTC uses certification authorities, to certify the authenticity of public keys held by the Client, Acquirer, Issuer and PG.

Goal 12 (Authorization):

Merchant checks the authenticity and integrity of the messages received from C, validates digital signatures, checks timestamps and nonce. If the Merchant is convinced about the TID and OI then only M authorizes the OI thereby achieving order secrecy OI is not known to any of the engaging entities other than C and M. So Merchant authorizes the transaction after obtaining un forgeable proof from the Client.

4.5. Performance Analysis

Time taken for Symmetric Encryption/Decryption (T_{SYMM}) using DES is 0.0005 seconds; Time taken for performing one-way hash function operation (T_h) is 0.00005 seconds and Time taken for performing XOR operation (T_{XOR}) is 0.00005 seconds approximately from table 4.2. Digital Signature Generation using ECDSA ($T_{sign(gen)}$) is 46 milli seconds and Digital Signature Verification using ECDSA ($T_{sign(verf)}$) is 94 milli seconds in 80 MHZ 32 bit ARM7TDMI Processor (Aydos et al., 2001):

So in our payment scheme the time taken for Issuance Phase is

$4T_h + 8T_{SYM} + 4T_{sign(gen)} + 4T_{sign(verf)}$ = 4 (0.00005) +8(0.0005) +4(0.046) +4(0.094) = 0.5642 seconds. (We excluded $I \rightarrow PG : (MTC, SNO)$ in step 5 of Issuance Phase as it takes place in private banking network). The time taken for Payment and Deposit Phase is $0T_h + 7T_{SYM} + 2$ $T_{sign(gen)} + 2T_{sign(verf)}$ = 0 + 7(0.00005) + 2(0.046) + 2(0.094) = 0.28035 seconds. (We excluded step 9 in Payment and Deposit Phase as the data exchange takes place in private banking network)

Operation	Number of operations per second
Public key signature (1024 bits RSA)	2
Symmetric key encryption (DES)	2000
One way hash function (MD5/SHA)	20000

Table 4.1: summaries of the computation speed of cryptographic functions from (Hwang et al., 2001)

The total time taken to complete entire Payment Scheme (i.e. Issuance, Payment and Deposit Phase) is 0.84455 seconds.

4.6. Comparative Analysis of MTC with Related Works

We have compared our proposed protocol with related work; the report of the analysis based on multiple features is presented in table 4.2. The last column results are of our proposed protocol.

Protocols Key Features	Hsien et al. (2001)	Horng-Twu et al.(2007)	Our Proposed Protocol (MTC)
Authentication	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes
Integrity	Yes	Yes	Yes
Non Repudiation	Yes	Yes	Yes
Identity Protection from Eavesdropper	No	No	Yes
Transaction Privacy Protection from Eavesdropper	No	No	Yes
Transaction Privacy Protection from PG	No	No	Yes
Prevents Double Spending & Over spending	Yes	Yes	Yes
Prevents Money Laundering	No	No	Yes
Withstands Replay Attack	No	No	Yes
Withstands Impersonation Attack	No	No	Yes
Withstands MITM Attack	No	No	Yes
Experimental Verification using BAN Logic, SVO Logic	No	No	Yes
Experimental Verification using AVISPA andSCYTHER TOOL	No	No	Yes
OI Authorization	No	No	Yes
PI Authorization	No	No	Yes

Table 4.2: Comparative Analysis of MTC with Related Work

4.7. Formal Verification of the proposed mobile payment protocol

4.7.1. Formal Verification of the proposed mobile payment protocol using extended BAN Logic

Analysis and verification of Issuance Phase using 'extended BAN logic'

Step 1: $C \to UC: \langle Application \rangle_{NRP}$

The Client (C) unlocks the payment application software using his NRP, which is stored in the tamper resistant UICC of the device. The Client (C) uses Non-repudiation PIN to convince UICC (UC) of his identity as given in **AS3**. The idealized form of step 1 is

$$C \to UC: \langle Application \rangle_{NRP}$$

Step 2: $C \to I: (MS1, DS_I^C)_{K_{CI}}, Cert_C$

Client (C) forwards the request to the issuer (I) to issue MTC by sending his $anonid$ and $ENDOR_C$ in order to prove his identity with device certificate. The idealized form of the message is $C \to I: \{C$ **believes** $\{MS1, DS_I^C\}_{K_{CI}}\}$

Step 3: $I \to C: \{MS2, DS_C^I\}_{K_{CI}}, Cert_I$

Verification of step2:

Issuer receives the message $\{MS1, DS_I^C\}_{K_{CI}}\}$ $Cert_C$ from the device. The idealized form of the message received is $\{MS1, DS_I^C\}_{K_{CI}}\}$, the issuer (I) decrypts the message using shared key between I and C as given in **AS4** so we conclude that

"I **believes** C **said** $\{MS1, DS_I^C\}_{K_{CI}}\}$ …. statement (1).

From statement (1) {MS1, DS_I^C } we get

"I **believes** C **said** {MS1}"…statement (2), "I **believes** C **said** { DS_I^C }"statement (3)

and we know from Statement (1) that certificate belongs to Client … Statement (4)

From **AS6 and AS7** and statement (2) we conclude

"I **believes fresh** (T_C)"…statement (5).

From statement (4), **AS1** and **AS2** issuer (I) ascertains K_C is the public key of the Client (C) so we conclude "I **believes** $(\overset{K_{ca}}{\mapsto} CA)$ **"…**statement (6), and

From Statement (3) and using K_C from **AS1** issuer (I) verifies the digital signature on M1 if it is successful the issuer believes that the signature was generated by **C** using his private key, then we can conclude {I **believes** DS_I^C }... Statement (7).

So from statement (1) to statement (7) we conclude

I **believes C believes** $\{(MS1, DS_I^C)_{K_{CI}}\ ,\ cert_C\}$

Then we can conclude I **believes** $\{(MS1, DS_I^C)_{K_{CI}}\ Cert_C$

After ascertaining that MS1 was from Client (C), I construct the following message $\{MS2, DS_C^I\}_{K_{CI}}, Cert_I$ and send it to Client (C). The idealized form of this message is

$I \rightarrow C : \{MS2, DS_C^I\}_{K_{CI}}, Cert_I$

Proof for the steps 4, 5 and 6 is done in the same way.

Verification of step 7: Merchant receives {C **believes** $\{(MS5, DS_m^c)\}_{K_m}$

Merchant (M) decrypts the message using K_m^{-1} from **AS6** so we can conclude

“M **believes** C **said**$\{(MS5, DS_m^c)\}_{K_m}$ ”……..statement (1).

Now we apply the belief conjuncatenation rule (Syyerson and Cervesato 2001) on $\{(MS5, DS_m^c)\}$ of statement (35) then we get “M **believes** C **said** MS5”…. statement (36),

“M **believes** C **said** DS_m^c”…. statement (2),

and we know from statement (3)

$\{\overset{K_c}{\mapsto} C, T_c\}_{K_{ca}^{-1}}$ … Statement (4).

From **AS7**, **AS8** and statement (36) we conclude that “M believes **fresh** (n_m) and **fresh** ($T_c^{'}$)”…statement (5) From **AS1, AS2** and statement (6) merchant (M) ascertains K_c is the public key of the Client and concludes “M **believes** $\overset{K_{ca}}{\mapsto} CA$**”….** statement (40).

From **AS11 “M believes CA controls** $\overset{K_{ca}}{\mapsto} CA$ **” ….** Statement (41).

Using K_c from **AS1** and statement (37) merchant (M) verifies the digital signature on MS5 if it is successful then “M believes that the signature was generated by client (C) using his private key then we can conclude “M **believes** DS_m^c **”….** statement (7). From statements 1 to 7 we can conclude “M **believes** D **believes**$\{(MS5, DS_m^c)\}_{K_m}$. Therefore we can conclude that “M **believes** $\{(MS5, DS_m^c)\}_{K_m}, cert_c$ ”

Step 9 is sent through secure private banking network, Issuer successfully checks PI using the assumptions **(AS4, AS10, AS11 & AS12)** given in Appendix B.

4.7.2. Formal Verification of the proposed mobile payment protocol using AVISPA Tool

The validation of our proposed protocol MTC using the OFMC back-end of AVISPA Tool is given in Appendix C as figure 4.2

The validation of our proposed protocol MTC using the CL-AtSe back-end of AVISPA Tool is given in Appendix C as figure 4.3

4.7.3. Formal Verification of the proposed mobile payment protocol using Scyther Tool

Result of MTC using 'Verification Claim' Procedure of Scyther Tool is given in Appendix C as figure 4.4
Result of MTC using 'Automatic Claim' Procedure of Scyther Tool is given in Appendix C as figure 4.5

4.8. Conclusions

We have proposed a novel payment instrument namely Mobile traveler's check (MTC) for Mobile Commerce which is as secure as e-check and can be spent freely anywhere like e-cash. ECDSA is used for signing and verifying the digital signatures and DES for encrypting and decrypting the messages which are suitable for resource constrained devices like mobile phones. The total time taken for the completion of Issuance, Payment and Deposit phase is 0.84455 seconds. This proves our payment instrument protocol is secure and efficient for resource constrained devices. Our proposed MTC is formally verified using BAN Logic, AVISPA and Scyther tool. We observe that MTC cannot work in offline environment and MTC is indivisible.

Chapter 5: Secure and Optimized Mobile based Merchant Payment Protocol

5.1. Introduction

In this chapter we consider a scenario in which a client tries to buy goods/services from merchant through a communication network (i.e. internet) and the client's platform is mobile phone equipped with UICC as secure element which is tamper resistant. Client cannot tamper the inner working of UICC because of its tamper resistant nature of the UICC. In this chapter we propose a Secure and Optimized Mobile based Merchant Payment (SOMMP) Protocol using Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve (Hwang et al., 2005) which consumes less computational and communication cost. The communication channel between UICC and mobile phone is secure and reliable and the communication channel among the engaged entities in our proposed protocol SOMMP is unreliable which is prone to attacks. In our proposed SOMMP protocol client sends message in the form of TransCertC (Transaction Certificate) which is a X.509 SLC (X.509 Short Lived Certificate) thereby reducing the number of client interactions among the engaging parties (i.e. reducing the consumption of resources from Client's perspective which are very scarce in Resource Constrained Devices like Mobile Phones). In SOMMP protocol WSLC (WPKI Short Lived Certificate) eliminates the need of certificates validation and removes the hurdle of PKI thereby reducing the storage space, communication cost and computational cost. SOMMP ensures all the security properties and is free from attacks.

5.2. Related Works

Mobile Payment Protocols proposed by Tellez et.al (Téllez, J., & Sierra, J. (2007a); Téllez, J., & Sierra, J. (2007b); Téllez, J., & Sierra, J. (2007c); Téllez, J., et al., 2006a; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) are suitable for scenarios with communication restrictions, (Téllez, J., & Sierra, J. (2007c)., Téllez, J., et al., 2006a) employs symmetric-key operations and (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) protocols employs Digital Signature with Message Recovery using Self-Certified public keys schemes based on RSA. Our proposed SOMMP protocol is suitable for scenarios with/without communication restrictions. Following are the gaps found in the literature.

a) Protocols proposed by Tellez et.al (Téllez, J., & Sierra, J. 2007c; Téllez, J., et al., 2006a) employs symmetric-key operations and (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) protocols are based on RSA (which consumes more computational and communication cost compared with ECC).

b) The number of Client interactions with other engaged parties are more.

c) Protocols proposed by Tellez et.al (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., & Sierra, J. 2007c; Téllez, J., et al., 2006a; Téllez, J., et al., 2006b; Téllez, J., et al., 2008) do not ensure forward secrecy and Public Verification.

d) In (Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., & Sierra, J. (2007c); Téllez, J., et al., 2006a; Téllez, J., et al., 2006b) protocols every Client C needs to register itself with merchant in merchant registration protocol thereby consuming lot of resources.

e) Téllez, J., & Sierra, J.2007c; Téllez, J., et al., 2006a protocols does not ensure non repudiation.

f) Téllez, J., & Sierra, J. 2007a; Téllez, J., & Sierra, J. 2007b; Téllez, J., et al., 2006b protocols cannot withstand Replay attack, Impersonation attack, and MITM attack. Téllez, J., et al., 2008 Protocol cannot withstand MITM attack and Multiprotocol attack.

g) Security protocols are error prone and are not easy to identify errors and prove their correctness. Mobile Payment Protocols proposed by Tellez et.al. were not verified using Manual Formal verification methods (like BAN Logic) or using Automated Formal verification Tools like AVISPA, Scyther and CryptoVerif.

The contributions made by us in this chapter are

a) We have proposed SOMMP Protocol for mobile networks using signcryption scheme with forward secrecy (SFS) proposed by Hwang et al., 2005 based on elliptic curve which reduces the computational and communication cost. SFS ensures forward secrecy and Public Verification.

b) TransCertC (Transaction Certificate) is a self-signed (X.509 Short Lived Certificate) certificate generated by the client containing attributes related to the transaction. TransCertC is sent as a message reducing the interactions among the engaging parties thereby reducing consumption of resources.

c) WSLC eliminates the need of certificates validation and removes the hurdle of PKI thereby reducing storage space, communication cost and computational cost.

d) In our proposed SOMMP protocol client need not register itself with merchant in merchant registration protocol thereby reducing the consumption of resources.

e) Merchant communicates with Acquirer but not with Payment Gateway (PG)

f) Our proposed SOMMP ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering.

h) In addition to these SOMMP withstands Replay, Man in the Middle, Impersonation and Multiprotocol attacks.

g) Our proposed SOMMP Protocol was successfully verified using BAN Logic, the code and the results of SOMMP using BAN Logic.

h) Our proposed SOMMP Protocol is modeled using the High Level Protocol Specification Language (HLPSL) and was verified successfully using AVISPA (Automated Validation of Internet Security Protocols and Applications) tool.

i) Our proposed SOMMP Protocol is modeled using the high-level formal language SPDL (Security Protocol Description Language) and were verified successfully using Scyther tool, the code and the results of SOMMP using this tool.

5.3. Secure and Optimized Mobile based Merchant Payment Protocol using Signcryption (SOMMP) protocol

Assumptions specific to this chapter:

1. Before the start of the protocol Client (C) visits Merchant (M's) site and gets his WSLC, list of items with their price, checks the credentials of merchant and validates merchants WSLC.

2. Single CA is used in our model which issues WSLC certificates and is trusted by all the engaged parties. All the engaged parties/actors/entities have their own WSLC digital certificates.

3. We have modified system initialization phase from Hwang et al. (2005) i.e. when a client C wants to join the system, he has to prove his credentials and requests for anonymous identity $Anonid_C$ then CA allocates anonymous identity to the client so instead of C, CA issues $Anonid_C$ as client's identity thereby achieving anonymity.

4. In our Proposed Mobile Payment Protocol (SOMMP) Client (C) sends TransCertC which is a short lived certificates as message, i.e. Client (C) creates short lived certificate which is similar to X.509 Public Key Certificate (PKC) whose validity period is less. Client (C) issues (builds and signs) TransCertC using his pseudonym in order to achieve anonymity (i.e. issuer's field will have pseudonym instead of real identity of Client (C)). The TransCertC includes extension fields such as timestamps, $H(MQ)$ $H(PI)$ and TID which are marked as critical. Client (C), Merchant (M), Issuer (I), Acquirer (A) and Payment Gateway (PG) are the entities involved in SOMMP protocol.

There are four phases in SOMMP protocol they are

a) Registration Phase

b) Negotiation Phase

c) Payment Phase and

d) Deposit Phase

a) *Registration Phase*

Registration phase is the same as given in chapter 2 but this protocol uses WSLC (WPKI Short Lived Certificate).

b) *Negotiation Phase*

Client accesses a merchant's web site from his GPRS enabled Mobile Phone, selects items to be purchased and makes a Purchase Order containing the following attributes. Upon receiving this request, the merchant generates a unique transaction identifier (TID) and generates a Merchant Quote (MQ) containing merchant's identity, merchant's Public key, client's Anonymous Identity, client's Public key, merchandise item code, quantity, unit price, amount, order identity.

Note: The size of all the attributes in data structures are given in bits

Once merchant creates a quote and sends it to a client the merchant cannot repudiate it because no one except the merchant can generate a signcryption using his/her private key on the message.

$Step1: C \rightarrow M : SIG_{C_M}(MS1), WSLC_C$ (Remote Mobile Payments)

$$MS1 = \{PO, N_C, T_C\}$$

$Step2: M \rightarrow C : SIG_{M_A}(MS2), WSLC_M$

$$MS2 = \{MQ, TID, N_C, T_C, N_M, T_M, Amt\}$$

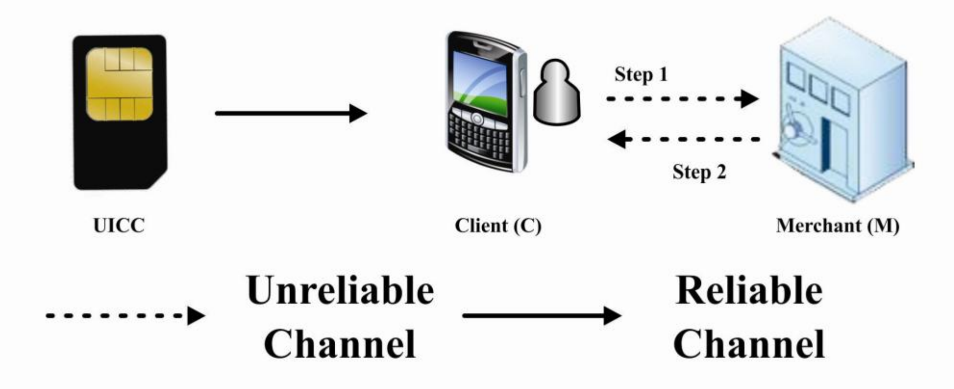


Figure 5.1: Negotiation Phase in SOMMP Protocol

C) Payment and Deposit Phase

$Step3: C \rightarrow M : SIG_{C_M}(MS3)$

$$MS3 = \{TransCertC, (PI)K_{ci}, MQ, TID, N_C, T_C, N_M, T_M, Amt\}$$

$$PI = AI, Amt, H(MQ), TID$$

In this phase Client (C) sends $SIG_{C_M}(MS3)$ to merchant (M) containing $\{TransCertC, (PI)_{K_{ci}}, MQ, TID, N_C, N_M, T_M, T_C, Amt\}$. $TransCertC$ is a short lived certificate generated by Client (C) with his $Anonid_C$. $TransCertC$ is sent as a message whose life is short lived i.e. it contains the validity period of the certificate and additional attributes are sent as extensions in the certificate. Four extensions are added in the $TransCertC$ certificate they are timestamps, $H(MQ)$ $H(PI)$ and TID. The main purpose of sending the data in the form of certificate is that is enables the Issuer (I) to get Client (C) authentication and assurances of the integrity and to lessen the communication cost, computation cost for the Client (C) which is very vital for resource constrained devices like mobile phones. $(PI)_{K_{CI}}$ is Payment Information containing $AI, Amt, H(MQ), TID$ encrypted with the shared secret key between C and I. TID, N_C, T_C, N_M, T_M are Transaction Identifier, nonce generated by M and C and timestamps of M and C.

Extension Name	Purpose of extension
Timestamps (TS)	This extension contains the initiation of the transaction by the Client (C)
$H(MQ)$	This extension contains the hashed product of MQ
HPI_C	This extension contains the hashed product of payment information (i.e. PI) sent by the Client (C) to Issuer (I).
TID	Transaction Identity generated by M

Table 5.1: List of Extensions used in TransCertC of SOMMP Protocol

$Step4: M \rightarrow A: SIG_{M_A}(MS4), WSLC_A$

$$MS4 = \{TransCertC, (PI)K_{ci}, TID, N_C, T_C, N_M, T_M, Amt\}$$

Merchant checks the authenticity and integrity of the message $MS3$ received from C, validates $TransCertC$, checks $MQ, TID, N_C, T_C, N_M, T_M$ if the validation and verification are successful then M generates and sends $MS4 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$ to Acquirer (A). If the Merchant is convinced about the TID and MQ then only M authorizes the MQ thereby achieving order secrecy MQ is not known to any of the engaging entities other than C and M, forwards $MS4$ containing only $TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M$ but not MQ.

$Step5: A \rightarrow PG: SIG_{A_{PG}}(MS5)$

$$MS5 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$$

Acquirer (A) checks the authenticity and integrity of the message $MS4$ it has received from M, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M, if the validation and verification are successful then it keeps a copy of $SIG_{M_A}(MS4)$ received from M, generates and forwards $MS5 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$ to Payment Gateway (PG).

$Step6: PG \rightarrow I: SIG_{PG_I}(MS6)$

$$MS6 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$$

Payment Gateway (PG) checks the authenticity and integrity of the message $MS5$ received from A, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M, if the validation and verification are successful then it keeps a copy of $SIG_{A_{PG}}(MS5)$ received from A, generates and forwards $MS6 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$ to Issuer (I).

$Step7: I \rightarrow PG: SIG_{I_{PG}}(MS7)$

$$MS7 = \{TID, Success, Amt\}$$

Issuer (I) checks the authenticity and integrity of the message $MS6$ received from PG, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M if the validation and verification are successful then it keeps a copy of $SIG_{PG_I}(MS6)$ received from PG, and authorizes the transaction generates and forwards $MS7 = \{TID, Success, Amt\}$ to Payment Gateway (PG). Payment Information (PI) contains $AI, Amt, H(MQ), TID$ which is encrypted using shared symmetric key between Client (C) and Issuer (I). Payment Information (PI) cannot be decrypted by the M, A, PG and I thereby achieving payment secrecy.

$Step8: PG \rightarrow A: SIG_{PG_A}(MS8)$

$$MS8 = \{TID, Success, Amt\}$$

Payment Gateway (PG) checks the authenticity and integrity of the message $MS7$ received from I, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M, if the validation and verification are successful then it keeps a copy of $SIG_{I_{PG}}(MS7)$ received from I, generates and forwards $MS8 = \{TID, Success, Amt\}$ to Acquirer (A).

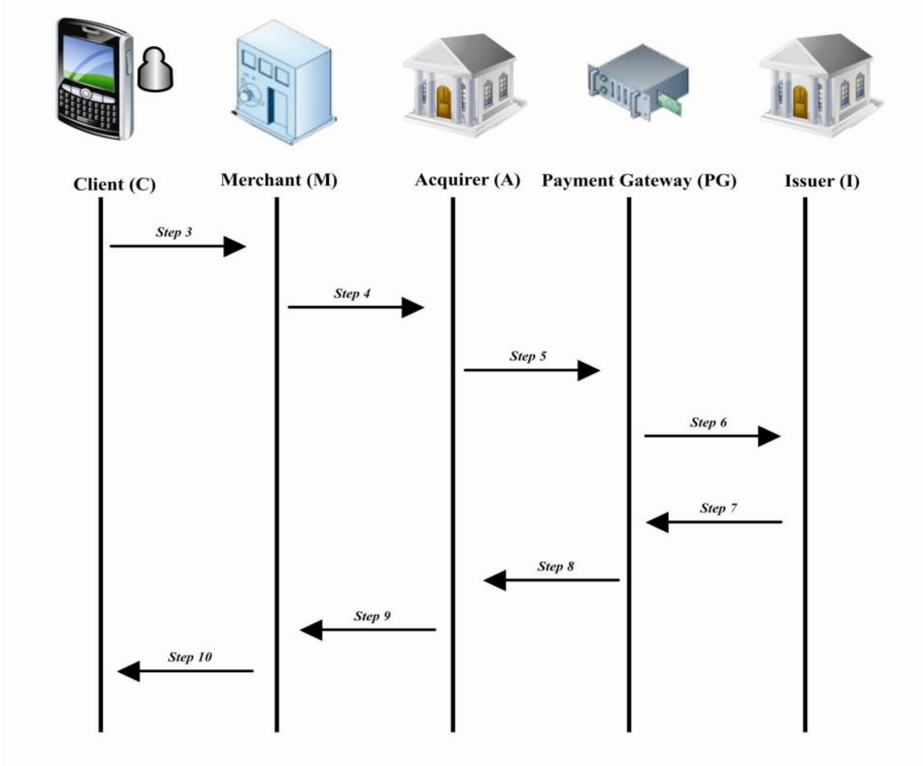


Figure 5.2: Payment and Deposit Phase in SOMMP Protocol

$Step9 : A \to M : SIG_{A_M}(MS9) \qquad MS9 = \{TID, Success, Amt\}$

Acquirer (A) checks the authenticity and integrity of the message $MS8$ received from PG, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M, if the validation and verification are successful then it keeps a copy of $SIG_{PG_A}(MS8)$ received from PG, generates and forwards $MS9 = \{TID, Success, Amt\}$ to Merchant (M).

$Step10 : M \to C : SIG_{M_C}(MS10) \qquad MS10 = \{TID, Success, Amt\}$

Merchant (M) checks the authenticity and integrity of the message $MS9$ received from A, validates $TransCertC$, checks TID, N_C, T_C, N_M, T_M, if the validation and verification are successful then it keeps a copy of $SIG_{A_M}(MS9)$ received from A, generates and forwards $MS10 = \{TID, Success, Amt\}$ to Client (C). So the computational, communication cost and storage space used for the execution of our proposed protocol is less from the Clients perspective which is vital for the survival of resource constrained devices like mobile device. So our proposed protocol is secure and optimized protocol for mobile based merchant payments.

Data Structures

```
Client's Purchase Order: =SEQUENCE {
        client's Anonymous Identity     Client's Anonymous Identity
        client's Public key             Alphanumeric (SIZE(163))
        merchant's Public key           Alphanumeric (SIZE(163))
        merchandise item code           Numeric String (SIZE(16))
        quantity                        Numeric String (SIZE(12))
        unit price                      Numeric String (SIZE(16))
       }

Merchant's Quote (MQ): =SEQUENCE {
            merchant's identity             Alphanumeric (SIZE(163))
            merchant's Public key           Alphanumeric (SIZE(163))
            client's Anonymous Identity     Client's Anonymous Identity
            client's Public key             Alphanumeric (SIZE(163))
            merchandise item code           Numeric String (SIZE(16))
            quantity                        Numeric String (SIZE(13))
            unit price                      Numeric String (SIZE(16))
            amount                          Numeric String (SIZE(24))
            order identity                  Alphanumeric (SIZE(163))
           }
```

Note: The size of all the attributes in data structures are given in bits

TransCertC

```
TransCertC{
    version             V3,
    serial Number       Certificate Serial Number,
    signature           AlgorithmIdentifier,
```

issuer $Anonid_C$,

validity { NOTbefore, NOTafter: valid time of authorization },

subject Merchant,

subjectpublickeyInfo Subject Public KeyInfo,

extensions Extensions }

SubjectpublickeyInfo ::= SEQUENCE {

algorithm AlgorithmIdentifier,

subjectpublickey BIT STRING }

Extensions: = SEQUENCE OF Extension

Extension of timestamps: = SEQUENCE {

extenID Extension.TS,

critical TRUE,

extnValue OCTET STRING }

Extension of $H(MQ)$: = SEQUENCE {

extenID Extension. $H(MQ)$,

critical TRUE,

extnValue OCTET STRING **}**

Extension of $H(PI)$: = SEQUENCE {

extenID Extension. $H(PI)$,

critical TRUE,

extnValue OCTET STRING **}**

Extension of TID : = SEQUENCE {

extenID Extension. TID,

critical TRUE,

extnValue OCTET STRING **}**

5.4. Security Analysis of SOMMP

We consider the goals defined in section 1.6 of Chapter 1 and perform the security analysis as below.

5.4.1. Data Security

Goal 1 (Third party):

All the entities involved in the SOMMP protocol store their credentials (Private Keys, NRP and WSLC) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the SOMMP protocol will exchange messages using signcryption mechanism which is a combination of encryption and digital signature so intruder will not be able to gain access to transactional data thereby achieving Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. Signcryption provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore signcryption was generated by means of SSCD so it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (SOMMP) withstands the following attacks

Replay Attacks: Replay attack is avoided by Timestamps, TID included in the $TransCertC$, Nonce (N_C) and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included in $TransCertC$ ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C’s private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating signcryption is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Our proposed protocol SOMMP withstands this attack because the intruder (In) does not have receiver's private key.

Multiprotocol Attack: Our proposed protocol SOMMP withstands this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2 (Secrecy):

In our proposed Mobile Payment Protocol (SOMMP) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing MQ (done by both the Client (C) and Merchant (M)). PG will not know about MQ and PI thereby achieving Transaction privacy from PG. Eavesdropper cannot get MQ and PI because the messages are encrypted using signcryption thereby achieving Transaction privacy from Eavesdropper.

Goal 3 (Uniqueness):

In our proposed Mobile Payment Protocol (SOMMP) every transaction is unique. The uniqueness is achieved due to the generation of fresh Transaction id by the merchant, and its verification by I, A and PG. Every the transaction is also linked to TID, Timestamps and Nonce.

5.4.2. Client Security

Goal 4 (Authentication):

Our proposed Mobile Payment Protocol (SOMMP) uses certifying authorities, to certify the authenticity of public keys held by the Merchant, Issuer, Acquirer and PG. The client's conversations are only with the Merchant and the Issuer and their authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5 (Authorization):

In our proposed Mobile Payment Protocol (SOMMP) the client obtains un forgeable proof of transaction authorization from the Merchant at the end of protocol (i.e. whether the transaction is success or failure).

Goal 6 (Identity Protection from Merchant and Eavesdropper):

In order to achieve Identity Protection from merchant and eavesdropper we have modified the initialization phase of Hwang et al., 2005. The client will be issued an anonymous identity $Anonid_C$ by the CA after successful verification of client's credentials.

5.4.3. Bank Security (Issuer, Acquirer and PG)

Goal 7(Authentication):

Our proposed Mobile Payment Protocol (SOMMP) has certifying authority to certify the authenticity of the public key held by the Client, Merchant and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their WSLC's. In addition to this client and Issuer share a symmetric key between them.

Goal 8 (Authorization):

The Issuer obtains an authorization proof for transaction from the client in the form of $TransCertC$ (Transaction Certificate signed by client using his private key) and PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer which contains payment details, merchant identity and hashed MQ. The Merchant obtains $TransCertC$ and PI (Payment Information) encrypted with the shared symmetric key and forwards this information to Acquirer after successful verification of digital signature on $TransCertC$, MQ and amount, merchant sends the hashed value of OI, which is verified by the Issuer. So Issuer authorizes the transaction after obtaining unforgeable proof from the Client and Merchant.

Goal 9(Prevents double spending, overspending and money laundering):

Issuer (I) keeps $TransCertC,(PI)_{K_{CI}},TID,N_C,N_M,T_M,T_C$ in its archives. If the client or merchant tries to double spend the PI, I can detect this from the validity period on the $TransCertC$, timestamp and serial number of the certificate. So double spending is avoided in SOMMP by Issuer (I). If the client or merchant tries to overspend, I avoids them in doing so since it checks Client's balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Banks (I, A and PG) are always involved in every payment transaction thereby preventing money laundering.

Goal 10(Issuer, Acquirer and PG turning malicious):

If any one or all of the entities in the private banking network (Issuer, Acquirer and PG) turns malicious then also they will not succeed in performing transaction on behalf of Client (C) because they have no knowledge about the private key of the Client (C).

5.4.4. Merchant Security

Goal 11(Authentication):

Our proposed Payment Protocol (SOMMP) uses certification authorities, to certify the authenticity of public keys held by the Client, Acquirer, Issuer and PG.

Goal 12 (Authorization):

Merchant checks the authenticity and integrity of the message $MS3$ received from C, validates $TransCertC$, checks MQ,TID,N_C,N_M,T_M,T_C. If the Merchant is convinced about the TID and MQ then only M authorizes the MQ thereby achieving order secrecy MQ is not known to any of the engaging entities other than C and M. So Merchant authorizes the transaction after obtaining un forgeable proof from the Client.

5.5. Performance Analysis of SOMMP

5.5.1. Computational Complexity from Client's Side (i.e. on the Mobile Device):

Operation	Number of operations per second
Public key signature (1024 bits RSA)	2
Symmetric key encryption (DES)	2000
One way hash function (MD5/SHA)	20000

Table 5.2: Summaries of the computation speed of cryptographic functions (Hwang, M.S et al., 2001)

So the time taken for Symmetric Encryption/Decryption (T_{SYMM}) using DES is 0.0005 seconds; Time taken for performing one-way hash function operation (T_h) is 0.00005 seconds approximately from table 5.3. Digital Signature Generation using ECDSA ($T_{sign(gen)}$) is 46 milli seconds and Digital Signature Verification using ECDSA ($T_{sign(verf)}$) is 94 milli seconds in 80 MHZ 32 bit ARM7TDMI Processor (Aydos, M et al., 2001) so the time taken to generate digital signature on X.509SLC using ECDSA is 46 milli seconds. SFS needs 166 milliseconds (Hwang et al., 2005). For generating SFS by the signer the signer needs $2T_{ECPM} + T_{MUL} + T_{ADD} + T_{HASH}$ and the recipient needs $3T_{ECPM} + T_{ECPA}$ (Hwang et al., 2005) where ECPM= the no of Elliptic Curve point Multiplication Operation, ECPA= the no of Elliptic Curve point Addition Operation, MUL= the no of Modular Multiplication Operation, ADD= the no of Modular Addition Operation, HASH= the no of One-way (or) Keyed one-way hash function Operation. ECPM needs only 83 milliseconds and Modular Exponentiation Operation needs 220 ms for average computational time in the Infineon's SLE66CUX640P security controller (Batina,L et al.,2003). So for generating SFS by the signer he/she needs 2*83=166ms. Recipient needs 3*83=249ms. Our proposed protocol (SOMMP) is light weight from the Client's side (i.e. from the mobile device). The Client is involved only in step 1 and 3 of the SOMMP protocol. The Client (C) generates one signature on TransCertC, one

Symmetric key encryption (DES), one way hash function (MD5/SHA) and one SFS (Hwang et al., 2005). So the total time taken for step 1 and 3 is 398.6 milliseconds.

Operation	No of operations	Time taken for each operation	Total time
Digital Signature on X.509 SLC using ECDSA	1	46 milliseconds	46 milliseconds
Symmetric key encryption (DES)	1	0.5 milliseconds	0. 5 milliseconds
One way hash function (MD5/SHA)	2	0.05 milliseconds	0.1 milliseconds
Signcryption with forward secrecy based on elliptic curve (Hwang et al., 2005)	2	166 milliseconds	2*166 =332 milliseconds

Table 5.3: Summaries the list and number of cryptographic functions performed by client in step 3 of SOMMP protocol

5.6. Comparative Analysis of SOMMP Protocol

Protocols Features	Téllez, J., & Sierra, J. 2007a	Téllez, J., & Sierra, J. 2007b	Téllez, J., & Sierra, J. 2007c	Téllez, J., et al., 2006a	Téllez, J., et al., 2006b	Téllez, J., et al., 2008	SOMMP
Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Non-Repudiation	Yes	Yes	No	No	Yes	Yes	Yes
Forward Secrecy	No	No	No	No	No	No	Yes
Order Secrecy	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Payment Secrecy	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Identity Protection from Merchant	Yes	Yes	Yes	No	Yes	Yes	Yes
Identity Protection from Eavesdropper	Yes	Yes	Yes	No	Yes	Yes	Yes
Transaction Privacy Protection from Eavesdropper	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Transaction Privacy Protection from PG	Yes	No	Yes	Yes	Yes	Yes	Yes
Certificate Validation is needed	No	No	No	No	No	No	No
Prevents Double Spending, Over spending & Money Laundering	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Withstands Replay Attack	No	No	Yes	No	No	Yes	Yes
Withstands Impersonation Attack	No	No	No	Yes	No	Yes	Yes
Withstands MITM Attack	No	No	No	Yes	No	No	Yes
Withstands Multi Protocol Attack	No	No	No	No	No	No	Yes
Number of Client interactions with engaging entities	2+4=6	2 + 4=6	2 + 4=6	3	2 + 2=4	2	2

Formal Analysis using BAN Logic	No	No	No	No	No	No	Yes
Formal Analysis using AVISPA and SCYTHER TOOL	No	No	No	No	No	No	Yes

Table 5.4: Comparative analysis of SOMMP Protocol with Related Works

We have compared our proposed protocol (SOMMP) with other related works; the report of the analysis based on multiple features is presented in table 5.4. The last column results are of our proposed protocol.

5.7. Formal Verification of SOMMP

5.7.1. Formal Verification of SOMMP using BAN Logic

Step 0: $C \rightarrow UC: \langle Application \rangle_{NRP}$

The Client (C) unlocks UICC using NRP. The Client (C) uses Non-repudiation PIN (NRP) to convince the UICC (UC) of his identity as given in **AS3**. The idealized form of step 0 is

$$C \rightarrow UC: \langle Application \rangle_{NRP}$$

Since UC receives NRP from C as given in **AS3**.

$$Step1: C \rightarrow M: SIG_{C_M}(MS1), WSLC_C$$

$$MS1 = \{PO, N_C, T_C\}$$

$$Step2: M \rightarrow C: SIG_{M_A}(MS2), WSLC_M$$

$$MS2 = \{MQ, TID, N_C, T_C, N_M, T_M\}$$

Proof of Step 1: After receiving the message $SIG_{C_M}(MS1)$ merchant (M) decrypts the signature and recovers MS1 using his private key

M **believes** C **said** {MS1})………..Statement (1)

Where $MS1 = \{PO, N_C, T_C\}$

From **AS5, AS6 and AS7** and statement (1) we conclude

“M **believes fresh** $(T_C^{'})$”…statement (2).

“M **believes fresh** (T_C)”………..Statement (3)

“M **believes fresh** (N_C)”…statement (4).

From **AS1** and **AS2** Merchant ascertains K_C is the public key of the Client (C) so we conclude

“M **believes** $(\overset{K_{ca}}{\mapsto} CA)$ **”…**Statement (5),

From Statement (1) to Statement (5)

M **believes** $\{SIG_{C_M}(MS1)\}$, $WSLC_C$

$$Step3 : C \rightarrow M : SIG_{C_M}(MS3)$$
$$MS3 = \{TransCertC, (PI)_{K_{CI}}, MQ, TID, N_C, T_C, N_M, T_M, Amt\}$$

Proof of Step 2: The idealized form of the message received is { $SIG_{M_A}(MS2)$ }, {$\overset{Y_m}{\mapsto} M$ **controls** C **believes** X}

After receiving the message $SIG_{M_A}(MS2)$ Client (C) decrypts the signature and recovers MS2 using his private key C **believes** M **said** {MS2})………..Statement (6)

Where $MS2 = \{MQ, TID, N_C, N_M, T_M, T_C\}$

From **AS6** and statement (5) we conclude

“C **believes fresh** $(T_M^{'})$”…statement (7).

From **AS6** C **believes fresh** (T_M)………..Statement (8)

M **believes fresh** (N_C)"…statement (9).

From Statement (6) to Statement (9)

C **believes M said** { $SIG_{M_A}(MS2), WSLC_M$ }

$Step4: M \rightarrow A: SIG_{M_A}(MS4), WSLC_A$

$$MS4 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, T_C, N_M, T_M, Amt\}$$

Proof of Step 3:

The idealized form is { $SIG_{C_M}(MS3)$ }. After receiving the message $SIG_{C_M}(MS3)$ Merchant (M) decrypts the signature and recovers MS3 using his private key

M **believes** C **said** {MS3})………..Statement (10)

Where $MS3 = \{TransCertC, (PI)_{K_{CI}}, MQ, TID, N_C, N_M, T_M, T_C, Amt\}$

From **AS6** we conclude "M **believes fresh** ($T_C^{'}$)"…statement (11)

From **AS6** M **believes fresh** (T_C)………..Statement (12)

M **believes fresh** (N_C)…statement (13).

From Statement (10) to Statement (13) M **believes C said** { $SIG_{C_M}(MS3)$ }

$Step5: A \rightarrow PG: SIG_{A_{PG}}(MS5)$

$$MS5 = \{TransCertC, (PI)_{K_{CI}}, TID, N_C, N_M, T_M, T_C, Amt\}$$

Proof of Step 4:

The idealized form is { $SIG_{M_A}(MS4), WSLC_A$ }

After receiving the message $SIG_{M_A}(MS4), WSLC_A$ Acquirer (A) decrypts the signature and recovers MS4 using his private key

A **believes** M **said** {MS4})………..Statement (14)

From **AS6** we conclude "A **believes fresh** ($T_M^{'}$)"…statement (15)

From **AS5, AS6 & AS7 we conclude**

A **believes fresh** (T_M)………..Statement (16)

A **believes fresh** (N_M)…statement (17).

From Statement (14) to Statement (17) A **believes** {$SIG_{M_A}(MS4), WSLC_A$}

Acquirer (A) successfully checks OI and PI from AS8, AS9 & AS12 given in Appendix B

$Step6: PG \rightarrow I: SIG_{PG_I}(MS6)$

Proof of Step 6

Step 6 is sent through secure private banking network Issuer successfully checks PI using the assumptions **(AS4, AS10, AS11 & AS12)** given in Appendix B.

$Step7: I \rightarrow PG: SIG_{I_{PG}}(MS7)$ $MS7 = \{TID, Success, Amt\}$

$Step8: PG \rightarrow A: SIG_{PG_A}(MS8)$ $MS8 = \{TID, Success, Amt\}$

/* Messages exchanged among I, A and PG are exchanged through Secure Private banking Network. So we are not verifying these exchanged messages */

$Step9: A \rightarrow M: SIG_{A_M}(MS9)$ $MS9 = \{TID, Success, Amt, T_A, T_A^{'}, N_A\}$

$Step10: M \rightarrow C: SIG_{M_C}(MS10)$ $MS10 = \{TID, Success, Amt, T_M, T_M^{'}, N_M\}$

Proof for Step 9:

After receiving the message $SIG_{A_M}(MS9)$ Merchant (M) decrypts the signature and recovers MS9 using his private key

M **believes** A **said** {MS9})………..Statement (18)

Where $MS9 = \{TID, Success, Amt\}$

From **AS6** we conclude "M **believes fresh** ($T_A^{'}$)"…statement (19)

From **AS6** we conclude M **believes fresh** (T_A)………..Statement (20)

From **AS6 & AS7** A **believes fresh** (N_A)…statement (21).

From Statement (18) to Statement (21) M **believes** A **said** { $SIG_{A_M}(MS9)$ }

Proof for Step 10:

After receiving the message $SIG_{M_C}(MS10)$ Client (C) decrypts the signature and recovers MS10 using his private key

C **believes** M **said** {MS10})………..Statement (22)

Where $MS10 = \{TID, Success, Amt, T_M, T_M^{'}, N_M\}$

From **AS6** we conclude "C **believes fresh** ($T_M^{'}$)"…statement (23)

From **AS6** we conclude C **believes fresh** (T_M)………..Statement (24)

From **AS6 & AS7** **C believes fresh** (N_M)…statement (25).

From Statement (22) to Statement (25) C **believes** M **said** { $SIG_{M_C}(MS10)$ }

5.7.2. Formal Verification of SOMMP using AVISPA Tool

The validation of our proposed protocol SOMMP using the OFMC back-end of AVISPA Tool is given in Appendix C as figure 5.3
The validation of our proposed protocol SOMMP using the CL-AtSe back-end of AVISPA Tool is given in Appendix C as figure 5.4.

5.7.3. Formal Verification of SOMMP using SCYTHER Tool

Result of SOMMP using 'Verification Claim' Procedure of Scyther Tool is given in Appendix C as figure 5.5.
Result of SOMMP using 'Automatic Claim' Procedure of Scyther Tool is given in Appendix C as figure 5.6.

5.8. Conclusions

This chapter proposes a Secure and Optimized Mobile based Merchant Payment (SOMMP) Protocol using Signcryption scheme with Forward Secrecy (SFS). SOMMP makes use of Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve and messages are sent in the form of Certificates X.509 SLC (X.509 Short Lived Certificates). SFS and X.509 SLC reduce the consumption of resources which are very scarce in resource constrained devices like mobile phones. SOMMP ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and PG, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these SOMMP withstands Replay, Man in the Middle and Impersonation attacks. Finally we have successfully verified the security properties of the proposed Mobile Payment protocol SOMMP using BAN Logic, AVISPA Tool and Scyther Tool.

Chapter 6: Mobile Payment Protocol based on Secure Electronic Transaction (SET)

6.1. Introduction

MasterCard and VISA have introduced SET protocol (MasterCard and Visa) which is a very popular credit-card payment protocol. SET was successfully implemented on fixed networks but it is not easy to implement it on wireless networks because of the nature of the SET itself and the problems in wireless networks. SET is a complex protocol which is implemented using public-key infrastructure (PKI). In this chapter we propose an Enhanced Mobile SET (EMSET) Payment Protocol in wireless environment using Mobile Agents and Digital Signature with Message Recovery (DSMR) based on ECDSA. Our proposed EMSET Protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. Proposed EMSET Protocol withstands Replay, Man in the Middle, Impersonation attack and Multi Protocol Attack. Our proposed EMSET Protocol has been verified successfully using BAN Logic, AVISPA and Scyther tools.

6.2. Related Works

To overcome these limitations Romao A. and da Silva M. M., 1998 proposed an agent-based SET payment system (SET/A). With SET/A, client is not required to stay connected to the Internet during the whole period of the transaction. An agent containing SET wallet plays the client's role in SET payment session. Thus, the client needs to connect to the Internet for short periods during the entire transaction. However, SET/A is vulnerable to attacks because the agent is required to bring SET wallet with it to perform cryptographic operations at the merchant environment which is considered to be hostile so Non repudiation property is not ensured. Wang X. F. et al, 1999 proposed SET/A+ which is a modified version of SET/A in order to solve the problems and limitations of SET/A. SET/A+ is operated in the larger scenario than that of SET/A, in that, it includes the brokering and negotiation phase which naturally requires the capability of agent in SET protocol. Client's Purchase Request is completely generated on the client's mobile device before it is brought with an agent to merchant. SET/A+ solves the problem of key compromise at the merchants site but performing all the cryptographic operations at clients site results in the problem of high computational load for the client. Moreover signature can be abused easily in malicious merchant environment, so Non repudiation property is not ensured. In order to overcome the limitations SET, SET/A and SET/A+ Supakorn Kungpisdan et al. (2003) have proposed to employ the combination of proxy –based solution and the agent technology to secure transactions and solve the problems of implementing SET payment in wireless environments. But the solution provided in (Supakorn Kungpisdan et al. (2003)) has to trust a proxy server, the client need to validate certificate of the issuer every time it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client, if digital signatures are generated in the memory of the mobile phone these signatures should not be considered as signatures because they are not generated in a tamper resistant device. SETNR/A protocol was proposed to improve the weakness of lacking non-repudiation mechanism from SET and SET/A for credit card-based transactions; on the other hand, agent-based protocol is ideal for complicated payment system. Broker generates a mobile agent for Buyer (i.e. client) which carries encrypted purchase order to Seller (i.e. merchant). A trusted third party (TTP) acts as a lightweight notary for evidence generations. But the solution provided by Chung-Ming Ou, C.R.Ou (2010) has to trust a proxy server, the client need to validate certificate of the Broker, merchant every time it wants to do a transaction and the authors did not elaborate where the digital signatures are generated by the client, if digital signatures are generated in the memory of the mobile phone these signatures should not be considered as signatures because they are not generated in a tamper resistant device. Authors of (Xiaolin Pang et al. (2002)) propose LITESET/A+ which does not ensure non repudiation property. In this chapter, we propose a new mobile payment protocol called EMSET based on SET which overcomes the flaws of existing solutions. SET Protocols proposed in the literature are not formally verified. So these protocols cannot be considered as secure protocols. We employ the combination of Mobile Agent technology and Digital Signature with Message Recovery (DSMR) mechanism in order to solve the problems of implementing SET payment in wireless environments. UICC is used a secure element for securely keeping client's credentials and for generating digital signatures. UICC is considered as SSCD (Secure Signature Creation Device) so digital signatures generated can be considered as legal signatures. We have verified our proposed EMSET protocol using BAN logic, AVISPA and Scyther Tools. Scyther tool is an automatic push-button tool for the verification and falsification of security protocols.

Contributions made

a) We have proposed an Enhanced Mobile SET (EMSET) Payment Protocol in mobile networks using Mobile Agents and Digital Signature with Message Recovery (DSMR) based on ECDSA.

b) Our protocol is proposed in the UICC of Mobile Device which is considered to be a tamper resistant device so UICC is a Secure Signature Creation Device (SSCD) because the signature processes are performed in the UICC and the private key never leaves the WIM. So non repudiation is ensured in devices where private key is stored in WIM.

c) The transaction flow in our proposed mobile payment protocol (EMSET) is from client to Issuer decreasing the risk of reusing client's information (PI) for the later transactions and issuer is a trusted entity of the client. So client can trust the TPE (Trusted Processing Environment) of the Issuer.

d) In our proposed EMSET client need not register itself with merchant in merchant registration protocol thereby reducing the consumption of resources.

e) Our proposed EMSET Protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering.

f) In addition to this EMSET Protocol withstands Replay, Man in the Middle, Impersonation and Multi Protocol attacks.

g) Our proposed EMSET Protocol has been verified successfully using BAN logic, AVISPA and Scyther tools.

6.3. Proposed Mobile Payment Protocol based on SET

Proxy Certificates: The proxy certificate basically follows standard certificate format (X.509) with minor change. The major difference is the subject identifier (SID), which is the certificate field, recorded the owner of this certificate. In proxy certificate, its subject identifier is equal to the certificate issuer. A proxy certificate (PC) of a mobile agent is issued and digitally signed by its owner. Beside standard certificate fields, this certificate contains a set of constraints which specifies valid operations that the agent is allowed to perform while using this certificate. $Cert\_Ver(PC\{C, K_{MA}[D]\}_{K_C^{-1}})$ is successful if and only if $Sign\_Ver(PC\{C, K_{MA}[D]\}_{K_C^{-1}})$ is successful namely $K_C(PC\{C, K_{MA}[D]\}_{K_C^{-1}}) = H(PC\{C, K_{MA}[D]\}_{K_C^{-1}})$. We use the notation $(PC\{C, K_{MA}[D]\}_{K_C^{-1}})$ to represent the proxy certificate of the Mobile Agent (MA) belonging to its owner Client (C) with data 'D'.

Trusted Processing Environment (TPE): A TPE in a mobile agent system provides a safe environment for the execution of any alien program; these include Software-based fault isolation and safe-code interpretation. We have modified system initialization phase from (Zuhua Shao et al. (2004)) i.e. when a client C wants to join the system, he has to prove his credentials and requests for anonymous identity $Anonid_C$ then CA allocates anonymous identity to the client so instead of C, CA issues $anonid_c$ as client's identity thereby achieving anonymity. Entities involved in EMSET protocol: Client (C), Merchant (M), Issuer (I), Acquirer (A) and Payment Gateway (PG). C & M trust their banks. Payment Gateway (PG) acts as an arbitrator; Issuer (I) authorizes Payment Information (PI) and Merchant (M) and Acquirer authorizes Order Information (OI). Client (C) initiates the payment transaction for purchasing goods from the merchant (M).

There are four phases in EMSET protocol they are 1) Registration Phase 2) Negotiation Phase 3) Payment Phase and 4) Deposit Phase.

Registration Phase

Registration Phase is given in chapter 2 of this thesis.

Negotiation Phase

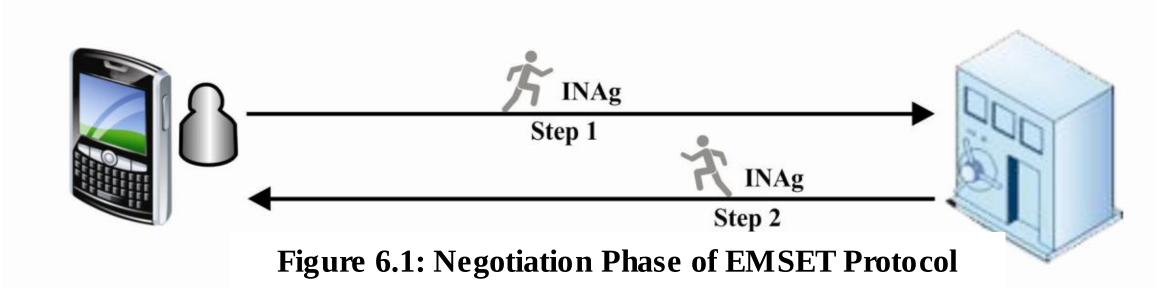


Figure 6.1: Negotiation Phase of EMSET Protocol

In our proposed protocol we employ three mobile agents performing three different tasks: (i) Information gathering and Negotiation Agent (INAg) for brokering and negotiating, which is used in Negotiation phase and (ii) Payment Agent (PAg1 and PAg2) for making payments used in Payment and Deposit phase. INAg is sent to collect the information about goods and the corresponding merchant information and return to client. PAg1 and PAg2 perform payment operations at the Issuer (I) and Payment Gateway (PG). Upon receiving this request, the merchant generates a unique transaction identifier (TID) and sends Order Information (OI), TID, HOI(Hashed OI), MID (Merchant's Identity) and Amt.

Payment Phase

$$Step1: C \rightarrow_{(PAg1)} I : DSMR_{C_I}(MS3), Pubkey_C$$

$$MS3 = PC_{PA1}, (PI)_{K_{CI}}, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$$

$$MS4 = OI, HOI_M, TID_M, MID, N_C, T_C, Amt_M$$

/* $C \rightarrow_{(PAg1)} I : MS$ means client (C) generates a Payment Agent 1 (PAg1) and sends message with the generated Payment Agent */

$$MS3 = PC_{PA1}, (PI)_{K_{CI}}, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$$

Client (C) generates a Payment Agent 1 (PAg1) for sending message MS to Issuer (I) PAg1carries this information to Issuer (I). Issuer (I) verifies the digital signature, validates the authenticity of the public key and recovers message MS3 from $DSMR_{C_I}(MS3)$

$$Step2: C \rightarrow_{(PAg2)} PG: DSMR_{C_{PG}}(MS5), Pubkey_C$$

$$MS5 = PC_{PA2}, HPI_C, HOI_C, TID_C, MID, N_C, T_C, Amt_C$$

Client (C) generates a Payment Agent 2 (PAg2) for sending message MS5 to Payment Gateway (PG) PAg2 carries this information to Payment Gateway (PG). Payment Gateway (PG) verifies the digital signature, validates the authenticity of the public key and recovers message MS5 from $DSMR_{C_{PG}}(MS5)$

Deposit Phase

$$Step3: I \rightarrow PG: HPI_I, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$$

The issuer (I) decrypts the message $DSMR_{C_I}(MS3)$ and gets MS3, decrypts PI using the shared symmetric key between the Client and the Bank, checks the PI if found successful it authorizes the PI (Payment Information) and proceeds with the protocol else it aborts the protocol. If the checks are successful it sends $HPI_I, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$ to Payment Gateway (PG).

$$Step4: PG \rightarrow A: TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$$

The Payment Gateway (PG) receives $DSMR_{C_{PG}}(MS5)$ from Client (C), decrypts the message $DSMR_{C_{PG}}(MS5)$ and gets $PC_{PA2}, HPI_C, HOI_C, TID_C, MID, N_C, T_C, Amt_C$. PG also receives $HPI_I, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$ from issuer (I) through Private Banking Network which is very secure. Payment Gateway (PG) will do the following verifications as given in using the Algorithm 2 from the data it received from Issuer (I) and Client (C)

Checks $HPI_C = HPI_I, HOI_C = HOI_I, N_C, T_C$

If all the verifications are found successful then it keeps a copy of the received messages from Issuer (I) and Payment Gateway (PG) and forwards $TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$ to the Acquirer (A) through Private Banking Network which is very secure.

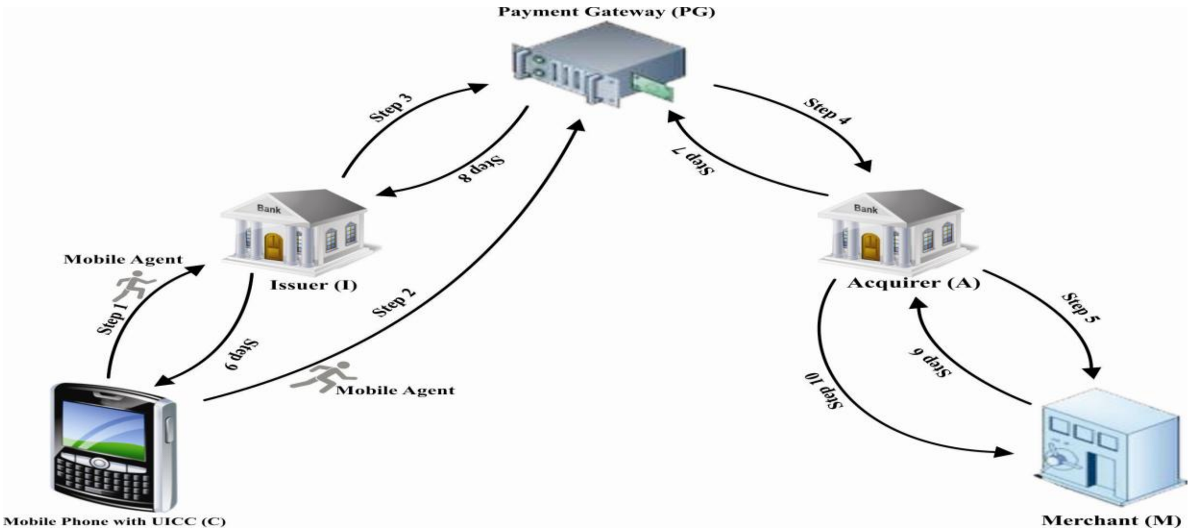


Figure 6.2: Payment and Deposit Phase of EMSET Protocol

$Step5: A \to M : DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$

A receives $TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4)$ from Payment Gateway (PG) and forwards it to M in the form of $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$

$Step6: M \to A: DSMR_{M_A}(TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M)$

M receives $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ from A and decrypts it using his private key and gets $(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$.Merchant (M) will do the following verifications from the data it received from Acquirer (A). Checks

$HOI_C = HOI_M, Amt_C = Amt_M$. If the verifications of the message are successful M authorizes Order Information and sends $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ to A.

$Step7: A \rightarrow PG: (TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M)$

Acquirer (A) receives $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ from M and decrypts it using his private key and gets $TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4)$. Acquirer (A) forwards $TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M$ message to PG.

$Step8: PG \rightarrow I: TID_C, MID, Amt_C, AuthOI$

PG receives $TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M$ from Acquirer (A) and performs all the verifications as given in Algorithm 2 if the verifications are successful it authorizes the transaction and forwards $TID_C, MID, Amt_C, AuthOI$ message to Issuer (I).

$Step9: I \rightarrow C: TID, Success / Failure, Amt$

Issuer (I) receives $TID_C, MID, Amt_C, AuthOI$ from PG and forwards $TID, Success / Failure, Amt$ to Client (C).

$Step10: A \rightarrow M: TID, Success / Failure, Amt$

Acquirer (A) sends $TID, Success / Failure, Amt$ to Merchant (M).

Authorization Algorithm of Transaction by PG

```
Algorithm AuthPG {     IF $HOI_C = HOI_I = HOI_M = TRUE$   {
So $HOI_C$ sent by C, $HOI_I$ sent by I and $HOI_M$ sent by M are same then PG Authorizes Order Information (OI)}
 ELSE  {    PG will not Authorize Order Information (OI)   }
IF       $TID_C = TID_I = TID_M = TRUE$     {
PG Authorizes TID }
ELSE  {
PG will not Authorize Transaction Identity (TID)    }
IF $HPI_C = HPI_I = HPI_M = TRUE$ {
 PG Authorizes Payment Information (PI) }
   ELSE  { PG will not Authorize Payment Information (PI) }{
   PG authorizes the transaction}
   ELSE {PG will not authorize the transaction it aborts the Transaction} }
```

6.4. Security Analysis

We consider the goals defined in section 1.6 of Chapter-1 and perform the security analysis as below.

6.4.1. Data Security

Goal 1 (Third party):

All the entities involved in the EMSET protocol store their credentials (Private Keys, NRP and Certificates) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the EMSET protocol will exchange messages using DSMR using self recovery using self certified public keys providing authenticated encryption scheme that integrates the mechanism of signature and encryption which enable only the specified receiver to verify and recover the original message. DSMR achieves Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. DSMR provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore DSMR was generated by means of SSCD so it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (EMSET) withstands the following attacks

Replay Attacks: Replay attack is avoided by Timestamps, TID, Nonce (N_C) and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C’s private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating DSMR is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Our proposed protocol EMSET withstands this attack because the intruder (In) does not have receiver’s private key.

Multiprotocol Attack: Our proposed protocol EMSET withstands this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2 (Secrecy):

In our proposed Mobile Payment Protocol (EMSET) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)). PG will not know about OI and PI thereby achieving Transaction privacy from PG. Eavesdropper cannot get OI and PI because the messages are sent using DSMR mechanism thereby achieving Transaction privacy from Eavesdropper.

Goal 3 (Uniqueness):

In our proposed Mobile Payment protocol (EMSET) every transaction is unique. The uniqueness is achieved due to the generation of fresh Transaction id by the merchant, and its verification by the bank. Every transaction is also linked to Timestamps and Nonce.

6.4.2. Client Security

Goal 4 (Authentication):

Our proposed Mobile Payment Protocol (EMSET) uses certifying authority to certify the authenticity of public keys held by the Client, Merchant and Payment Gateway. The client’s conversations are only with the Issuer, PG and Merchant. Client, Merchant, Issuer and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5 (Authorization):

In our proposed Mobile Payment Protocol (EMSET) the client obtains unforgeable proof of transaction authorization from the Issuer at the end of protocol (i.e. whether the transaction is success or failure). I, A and PG obtains an authorization proof for transaction from the client and merchant in the form of HOI_M, HOI_C and PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer containing payment details, merchant identity and hashed OI. So Acquirer authorizes OI of the transaction, I authorizes OI and PI of the transaction.

Goal 6 (Identity Protection from Merchant and Eavesdropper):

In order to prevent a merchant from knowing the identity of Client, an anonymous identity is enrolled by the client with CA and Issuer. CA and Issuer know the real identity of the client. Therefore merchant and eavesdropper cannot map the anonymous identity with C'S true identity, client's privacy is protected and untraceable.

6.4.3. Bank Security (Issuer, Acquirer and PG)

Goal 7(Authentication):

Our proposed Mobile Payment Protocol (EMSET) has certifying authority to certify the authenticity of the public keys held by the Client, Merchant and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer share a symmetric key between them.

Goal 8 (Authorization):

The Issuer obtains an authorization proof for transaction from the client in the form of PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer which contains payment details, merchant identity and hashed OI. The Merchant obtains OI and

PI (Payment Information) encrypted with the shared symmetric key and forwards this information to Acquirer after successful verification of digital signature on the message, OI and amount merchant sends hashed OI, which is verified by the Issuer. So Issuer authorizes the transaction after obtaining unforgeable proof from the Client and Merchant.

Goal 9(Prevents double spending, overspending and money laundering):

Issuer (I) keeps $N_C, N_M, T_C, T_M, TID, (PI)_{K_{CI}}$ in its archives. If the client or merchant tries to double spend the PI, I can detect this from TID, nc, nm, Tm, Tc so double spending is avoided in EMSET by Issuer. If the client or merchant tries to overspend, I avoids them in doing so since it checks Client's balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Issuer is always involved in every payment transaction thereby preventing money laundering.

Goal 10(Issuer, Acquirer and PG turning malicious):

In our proposed EMSET if any one or all the entities in the private banking network (Issuer, Acquirer and PG) turns malicious then also they will not succeed in performing transaction on behalf of Client (C) because they have no knowledge about the private key of the Client (C).

6.4.4. Merchant Security

Goal 11(Authentication):

Our proposed Payment Protocol (EMSET) uses certifying authority to certify the authenticity of public keys held by the Client, Acquirer, Issuer and PG.

Goal 12 (Authorization):

Merchant checks the authenticity and integrity of the messages received from C, validates DSMR, checks timestamps and nonce. If the Merchant is convinced about the TID and OI then only M authorizes the OI thereby achieving order secrecy, OI is not known to any of the engaging entities other than C and M. So Merchant authorizes the transaction after obtaining un forgeable proof from the Client.

6.5. Comparative Analysis

PROTOCOLS FEATURES	Master card and Visa (1997)	Romao A. and da Silva M. M., 1998	Wang X. F. et al, 1999	Supakorn Kungpisd an (2003)	Chung-Ming Ou, C.R.Ou (2010)	Xiaolin Pang et al (2002)	EMSET
Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Non- Repudiation	Yes	No	No	Yes	Yes	No	Yes
Forward Secrecy	NR	NR	NR	NR	NR	NR	Yes
Order Secrecy	NR	NR	NR	NR	YES	NR	Yes
Payment Secrecy	NR	NR	NR	NR	YES	NR	Yes
Key pairs are generated and stored in Tamper resistant device	No	No	No	No	No	No	Yes
Are the Signatures generated in SSCD	No	No	No	No	No	No	Yes
For every transaction client needs to agree for a shared symmetric key with PG	Yes	Yes	Yes	Yes	Yes	Yes	No
Execution of Mobile Agent at merchant's TPE	Yes	Yes	Yes	No	Yes	Yes	No
Identity Protection from Merchant	No	No	No	No	No	No	Yes
Identity Protection from Eavesdropper	Yes	No	No	No	No	No	Yes
Transaction Privacy Protection from PG & Eavesdropper	Yes	No	No	No	No	No	Yes
Certificate Validation is needed	Yes	Yes	Yes	Yes	Yes	Yes	No
Prevents Double Spending, Over Spending & Money Laundering	NR	NR	NR	NR	NR	NR	Yes
Withstands Replay, Impersonation, MITM & Multi Protocol Attacks	No	No	No	No	No	No	Yes
Formal Analysis using BAN Logic, AVISPS and SCYTHER TOOL	No	No	No	No	No	No	Yes

Table 6.1: Comparative Analysis of EMSET Protocol with Related Works

We have compared our proposed protocol (EMSET) with other related works; the report of the analysis based on multiple features is presented in table 6.1. The last column results are of our proposed protocol.

6.6. Formal verification

6.6.1. Formal verification of EMSET using BAN Logic

Security Proof of the Proposed Protocol

$Step1: C \rightarrow_{(PAg1)} I: DSMR_{C_I}(MS3), Pubkey_C$

$MS3 = PC_{PA1}, (PI)_{K_{CI}}, HOI_C, TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$

$MS4 = OI, HOI_M, TID_M, MID, N_C, T_C, Amt_M$

Issuer (I) receives $DSMR_{C_I}(MS3), Pubkey_C$ from Client (C) and decrypts the message using his private key and gets MS3 so from the assumptions **AS1, AS2, AS5 & AS6**

I **believes** $DSMR_{C_I}(MS3), Pubkey_C$ ………..statement (1)

I validate the public key parameters of Client (C) and from **AS4** it decrypts the symmetric key shared between C and I and validates PI using **AS10 & AS11**

If the public key parameters of Client (C) are successfully validated then

I **believes C said** $Pubkey_C$ ………..statement (2)

I **believes C said MS3** ………..statement (3)

From **AS1 & AS2** I **believes fresh** T_C ………..statement (4)

I **believes fresh** N_C ……..statement (5)

From statements 1 and 5

I **believes C said** $DSMR_{C_I}(MS3), Pubkey_C$ ……..statement (6)

$Step2: C \rightarrow_{(PAg2)} PG: DSMR_{C_{PG}}(MS5), Pubkey_C$

$MS5 = PC_{PA2}, HPI_C, HOI_C, TID_C, MID, N_C, T_C, Amt_C$

Payment Gateway (PG) receives $DSMR_{C_{PG}}(MS5), Pubkey_C$ from Client (C) and decrypts the message using his private key and gets $MS5$ so from the assumptions **AS1, AS2, AS6**

PG **believes** $DSMR_{C_{PG}}(MS5), Pubkey_C$ ………..statement (7)

PG validate the public key parameters of Client (C)

If the public key parameters of Client (C) are successfully validated then

PG **believes C said** $DSMR_{C_{PG}}(MS5), Pubkey_C$ ………..statement (8)

PG **believes C said MS5** ………..statement (9)

From **AS1 & AS2**

PG **believes fresh** T_C ………..statement (10)

PG **believes fresh** N_C …..statement (11)

From statements 7 and 11

PG **believes C said** $DSMR_{C_{PG}}(MS5), Pubkey_C$ ………..statement (12)

$Step4: PG \rightarrow A: TID_C, MID, N_C, T_C, Amt_C, DSMR_{C_M}(MS4)$

Step 3 and Step 4 are the messages exchanged in private banking network so they are sent as DSMR. So we are not formally verifying these steps.

$Step5: A \rightarrow M: DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$

Merchant (M) receives $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ from Acquirer (A) and decrypts the message using his private key and gets $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ so from the assumptions **AS1, AS2, AS6**

M **believes** $DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ ………..statement (13)

M validate the public key parameters of Acquirer (A)

If the public key parameters of Acquirer (A) are successfully validated then

M **believes A said** $Pubkey_a$………..statement (14)

From statements 13 and 14

M **believes A said**

$DSMR_{A_M}(TID_C, HOI_C, MID, Amt_C, DSMR_{C_M}(MS4))$ ………..statement (15)

$Step6: M \rightarrow A: DSMR_{M_A}(TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M)$

Acquirer (A) receives $DSMR_{M_A}(TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M)$ from Merchant (M) and decrypts the message using his private key and gets $TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M$ so from the assumptions **AS1, AS2, AS5 & AS6**

From the **AS8, AS9, AS12** of Appendix B, **A checks** PI which is encrypted using shared symmetric key, OI (i.e. HOI)

A **believes** $TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M$ …..Statement (16)

Acquirer (A) validate the public key parameters of Merchant (M)

If the public key parameters of Merchant (M) are successfully validated then

A **believes M said** $Pubkey_m$………..statement (17)

From statements 16 and 17 A **believes M said**

$DSMR_{M_A}(TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M), Pubkey_M$ ………..statement (18)

$Step7: A \rightarrow PG: TID_C, HOI_C, MID, Amt_C, AuthOI, HOI_M$

$Step8: PG \rightarrow I: TID_C, MID, Amt_C, AuthOI$

Step 7 and Step 8 are the messages exchanged in private banking network so they are not sent as DSMR. So we have not formally verified these steps.

$Step9: I \rightarrow C: TID, Success/Failure, Amt$

$Step10: A \rightarrow M: TID, Success/Failure, Amt$

Messages exchanged in Step 9 and Step 10 are not encrypted using DSMR as these messages are sent as acknowledgements to Client and Merchant.

6.6.2. Formal verification of EMSET using AVISPA tool

The validation of our proposed protocol EMSET using the OFMC back-end of AVISPA Tool is given in Appendix C as figure 6.4
The validation of our proposed protocol EMSET using the CL-AtSe back-end of AVISPA Tool is given in Appendix C as figure 6.5

6.6.3. Formal verification of EMSET using Scyther tool

Result of EMSET using 'Verification Claim' Procedure of Scyther Tool is given in Appendix C as figure 6.6

Result of EMSET using 'Automatic Claim' Procedure of Scyther Tool is given in Appendix C as figure 6.7

6.7. Conclusions

In this chapter we have proposed an Enhanced Mobile SET (EMSET) protocol with formal verification using Mobile Agent technology and Digital Signature with Message Recovery based on ECDSA mechanism. Mobile Agent technology and Digital Signature with Message Recovery (DSMR) based on ECDSA mechanism are used in EMSET protocol for Mobile Networks. Mobile Agent technology has many benefits such as bandwidth conservation, reduction of latency, reduction of completion time, Asynchronous (disconnected) communications. Digital Signature with Message Recovery based on ECDSA eliminates the need of adopting PKI cryptosystems. Our proposed protocol EMSET ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these our proposed protocol withstands Replay, Man in the Middle, Impersonation and Multi Protocol attacks. The security properties of the proposed protocol have been verified using BAN logic, AVISPA and Scyther Tools and presented with results.

Chapter 7: Secure and Optimized Proximity Mobile Payment Protocol

7.1. Introduction

We consider a scenario in which client has a NFC enabled Mobile Phone with UICC as secure element, merchant has NFC enabled POS (Point Of Sale) and all the items (in the store) chosen by client are all tagged with NFC tags. Client chooses some tagged items at a department store, and approaches one of the several NFC enabled POS in the store. We consider 6 entities: (i) Client with NFC-enabled Smartphone with UICC, (ii) Merchant with NFC enabled POS (iii) Issuer (client's Bank) (iv) Acquirer (merchant's Bank) (v) Payment Gateway (PG) and (vi) Certifying Authority (CA) in our ecosystem. The communication channel between UICC and mobile phone is secure and reliable; Issuer, Acquirer and Payment Gateway exchange messages in Secure Private Banking Network. The data exchanged among the entities involved in this chapter are in the form of DSMR (Zuhua Shao (2004)). The NFC enabled POS scans the tagged items in the shopping cart and generates a signed (merchant signs using his private key) Digital Invoice Certificate (DIC). Our proposed protocol ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these our proposed protocol withstands Replay, Man in the Middle and Impersonation attacks. The security properties of the proposed protocol have been verified using AVISPA and Scyther Tools and presented with results. Existing mobile payment solutions based on proximity does not ensure all the security properties, proposed protocols are prone to attacks, proposed in the memory of mobile phone and the proposed protocols were not verified with formal methods and tools. So in this chapter we propose a Secure and Optimized Proximity Mobile Payment (SOPMP) Protocol using NFC (Near Field Communication) technology, WPKI (Wireless Public Key Infrastructure), UICC (Universal Integrated Circuit Card) and Digital Signature Message Recovery (DSMR).

7.2. Related Work

Existing literature on NFC based Mobile Payments does not ensure end-to-end security, consumes more resources and there is no formal proof to prove that their proposed Mobile Payment protocol is free from attacks. So there is a need of a protocol which ensures end-to-end security, consumes optimal resources and is free from attacks (verified using formal tools). Following are the gaps found in the related work.

GAPS FOUND IN RELATED WORK

a) Huda Ubaya et al. (2012) solution is a prototype application designed for the implementation of security in e-commerce transactions by using Tag-to-Tag protocol so that the user needs security and comfort during the financial transaction. But this solution has the following flaws

   i) It does not ensure all the security properties (especially non repudiation)
   ii) It works in the memory of mobile phone (i.e. Android)
   iii) It is not clear where the session key (shared session key) generated for encryption and decryption during the payment process between merchant and user is stored
   iv) It is prone to attacks
   v) It was not verified with formal methods and tools

b) Emir Husni et al. (2012) solution proposes a secure and efficient protocol that will reduces physical activity of the device owners and reduce transaction time. The data exchanged between merchant and payer is executed without waiting for each other and one transaction requires two data transmissions which are performed by the merchant and payer. Transactions are secured by the use of encryption on each data sent by the merchant and payer. This solution has the following drawbacks

   i) Does not ensure non repudiation
   ii) It is prone to attacks
   iii) It was not verified with formal methods and tools

c) Massaoth and Bingel's et al. (2009) discusses about different mobile payment services compared with a NFC based solution. They showed that NFC is a trend for mobile payment solution. But it's not clear how NFC technology improves the security of mobile payments.

d) Tan Soo Fun et al. (2008) proposed a symmetric key centric mobile payment system that constructs on the mobile operator protocol. Pointing out the symmetric mobile payment system has performance advantages on the limited computation platform than the PKI based system, e.g. the handset. Their symmetric system reduces the communications steps between engaging parties meanwhile the security is still not compromised. Their solution does not ensure non repudiation and moreover their solution is proposed in the memory of mobile phone.

e) Authors of Chen, W et al. (2010) & Weidar Chen et al. (2010) proposes mobile payment system for merchants, which can be built on existing GSM and NFC architecture. Their proposal leverages the SIM's authentication and identification capabilities and uses GSM and 3G cryptographic primitives, which simplifies integration into the current mobile infrastructure but their solution does not ensure non repudiation, and their solution is proposed in the memory of mobile phone.

f) Mateja Jovanovic and Mario Muñoz Organero et al. (2011) proposes new mobile commerce proximity payment architecture, based on the analysis of existing solutions and current and future market needs. But they did not propose any protocol.

g) In the existing proximity mobile payments solutions client's credentials are generated and stored in the memory of mobile phone and could be infected by viruses or can be aliciously replaced, it does not ensure secure and reliable communication security and does not ensure end to end security in the application layer.

h) Protocols proposed in the realm of Proximity Mobile Payments were formally verified using manual verification methods like BAN logic and SVO Logic but not with Automated Tools like Scyther ((Cremers, C. J. F.(2006)) & (Cremers, C. and Lafourcade 2007))) and AVISPA (Armando.A *et al.,* 2005).

Contributions made

a) We have proposed SOPMP protocol for mobile networks using DSMR based on elliptic curve which eliminates the need of certificates validation and removes the hurdle of PKI thereby reducing storage space, communication cost and computational cost.

b) Our protocol is proposed in the UICC of Mobile Device which is considered to be a tamper resistant device as UICC is a Secure Signature Creation Device (SSCD) because the signature processes are performed in the UICC and the private key never leaves the WIM. So non repudiation is ensured in devices where private key is stored in WIM.

c) Digital Invoice Certificate (DIC) is a self-signed (X.509 Short Lived Certificate) certificate generated by the merchant containing transaction attributes related to the transaction. DIC is sent as a message reducing the number of interactions among the engaging parties thereby reducing consumption of resources.

d) In our proposed SOPMP protocol client need not register itself with merchant in merchant registration protocol thereby reducing the consumption of resources.

e) Our proposed SOPMP ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, and prevents Double Spending, Overspending and Money laundering.

f) In addition to these SOPMP withstands Replay, Man in the Middle, Impersonation and Multi Protocol attacks.

g) Our proposed SOPMP Protocol is modeled using the High Level Protocol Specification Language (HLPSL) and result was verified successfully using AVISPA (Automated Validation of Internet Security Protocols and Applications) tool.

h) Our proposed SOPMP Protocol is modeled using the high-level formal language SPDL (Security Protocol Description Language) the code and results were verified successfully using Scyther tool.

7.3. Our Proposed Mobile Payment Protocol (SOPMP)

Client chooses some tagged items at a department store, and approaches one of the several NFC enabled POS in the store. The NFC enabled POS scans the tagged items in the shopping cart and generates a signed (merchant signs using his private key) Digital Invoice Certificate (DIC). The data exchanged among the entities involved in this chapter are in the form of DSMR. DIC is a short lived certificate which is similar to X.509 Public Key Certificate (PKC) whose validity period is less. The DIC includes extension fields such as timestamps, $H(MQ)$ and TID which are marked as critical. All the steps involved in our proposed NFC based Mobile Payment Protocol is shown in figure 7.1.

$$Step1: C \rightarrow M: \{DSMR_{C_M}(MS4)\}\ MS4 = \{OI, ID_C, N_C, T_C\}$$

Client (C) approaches one of the several NFC enabled POS in the store with chosen tagged items at a department store and sends $\{DSMR_{C_M}(MS4)\}$ from his NFC enabled Mobile Phone (with UICC as secure element) to NFC enabled POS (Point Of Sale) containing $\{OI, ID_C, N_C, T_C\}$.

$$Step2: M \rightarrow C: \{DSMR_{M_C}(MS5)\}\ MS5 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M\}$$

NFC enabled POS scans the tagged items in the shopping cart and generates a signed (merchant signs using his private key) Digital Invoice Certificate (DIC).Merchant decrypts $\{DSMR_{C_M}(MS4)\}$ using his private key and gets $\{ID_C, N_C, T_C\}$ the data exchanged among the entities involved in this chapter are in the form of DSMR. DIC is a short lived certificate which is similar to X.509 Public Key Certificate (PKC) whose validity period is less. The DIC includes extension fields such as timestamps, $H(MQ)$ and TID which are marked as critical. Then merchant generates $\{DSMR_{M_C}(MS5)\}\ MS5 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M\}$ sends it to Client (C).

$$Step3: C \rightarrow M: \{DSMR_{C_M}(MS6)\}$$

$MS6 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$

Client (C) decrypts $\{DSMR_{M_C}(MS5)\}$ using his private key and gets $MS5 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M\}$ validates Certificate using the procedure given in D.R.Stinson (2006). After successful verification and validation client sends $\{DSMR_{C_M}(MS6)\}$

$MS6 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$ to merchant.

$Step4 : M \rightarrow A : \{DSMR_{M_A}(MS7)\}$
$MS7 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$

Merchant (M) decrypts $\{DSMR_{C_M}(MS6)\}$ using his private key and gets $MS6 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$

Merchant sends $\{DSMR_{M_A}(MS7)\}$ $MS7 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$ to Acquirer.

$Step5 : A \rightarrow PG : \{(MS8)\}$

$MS8 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$

Acquirer decrypts $\{DSMR_{M_A}(MS7)\}$ and gets $MS7 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$ Acquirer forwards $\{(MS8)\}$ message to the PG $MS8 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$ via Secure Private Banking Network.

$Step6 : PG \rightarrow A : \{(MS9)\}$
$MS9 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$
PG receives $MS8 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$ from the Acquirer through Private Banking Network which is very secure. Payment Gateway (PG) will perform the

following verifications from the MS8 it has received. Validates DIC, Checks Timestamps and Checks nonce. If all the verifications are found successful then it keeps a copy of the received message MS8 and forwards $MS9=\{DIC,Amt,ID_C,N_C,T_C,ID_M,N_M,T_M,(PI)_{K_{CI}}\}$ to the Issuer (I).

$Step7:I\rightarrow PG:\{(MS10)\}$
$MS10=AuthPI$

Issuer receives $MS9=\{DIC,Amt,ID_C,N_C,T_C,ID_M,N_M,T_M,(PI)_{K_{CI}}\}$ from the PG through Private Banking Network which is very secure. Issuer (I) will perform the following verifications from the MS9 it has received. Validates DIC, Checks Timestamps, and Checks nonce and Decrypts the PI using the shared symmetric key between Issuer and Client and checks the PI, if the client has sufficient funds it authorizes the transaction. If all the verifications are successful it authorizes the transaction and sends $\{(MS10)\}$ to PG.

$Step8:I\rightarrow C:\{DSMR_{I_C}(MS11)\}$

$MS11=\{TID,Amt,ID_M,Success/Failure\}$

Issuer (I) informs Client (C) about the success/failure of the transaction

$Step9:PG\rightarrow A:\{MS12)\}$ $MS12=\{TID,Amt,ID_M,Success/Failure\}$

PG informs Acquirer (A) about the success/failure of the transaction.

$Step10:A\rightarrow M:\{MS13)\}$ $MS13=\{TID,Amt,ID_M,Success/Failure\}$

A informs merchant (M) about the success/failure of the transaction

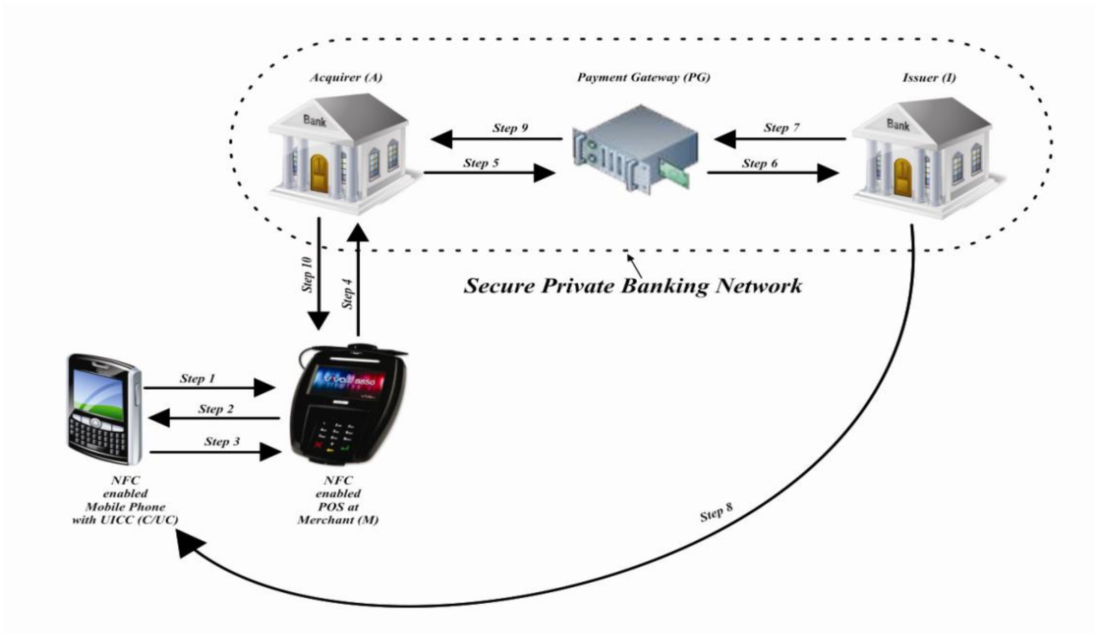


Figure 7.1. NFC based Mobile Payment Protocol

Digital Invoice Certificate (DIC)

```
DIC {
      version                    V3,
     serial Number              Certificate Serial Number,
     signature                  AlgorithmIdentifier,
      issuer                    merchantID,
    validity                    { NOTbefore, NOTafter: valid time of authorization },
     subject                     Client,
     subjectpublickeyInfo        Subject Public KeyInfo,
      extensions                 Extensions
 }
SubjectpublickeyInfo ::= SEQUENCE  {
      algorithm                 AlgorithmIdentifier,
       subjectpublickey          BIT STRING
  }
```

Extensions ::= SEQUENCE OF Extension

Extension of timestamps: = SEQUENCE {

extenID Extension.TS,

critical TRUE,

extnValue OCTET STRING }

Extension of $H(MQ)$: = SEQUENCE {

extenID Extension. $H(MQ)$,

critical TRUE,

extnValue OCTET STRING **}**

Extension of TID : = SEQUENCE {

extenID Extension. TID ,

critical TRUE,

extnValue OCTET STRING **}**

7.4. Security Analysis of SOPMP

We consider the goals defined in section 1.6 of Chapter 1 and perform the security analysis as below.

7.4.1. Data Security

Goal 1 (Third party): All the entities involved in the SOPMP protocol store their credentials (Private Keys, NRP and Certificates) in tamper resistant hardware tokens so that their credentials are not compromised. All the entities involved in the SOPMP protocol will exchange messages using DSMR using self recovery using self certified public keys providing authenticated encryption scheme that integrates the mechanism of signature and encryption which enable only the specified receiver to verify and recover the original message. DSMR achieves Data confidentiality, Entity Authentication, Data Integrity and Non-Repudiation. DSMR provides non repudiation evidence that the signer knew the information and that unless his/her private key was compromised (this attack is not possible if we use perfect cryptography and use SSCD), he/she was the only entity able to generate mobile signature. Furthermore DSMR was generated by means of SSCD so it is recognized automatically as legally equivalent to handwritten signature. Our proposed Mobile Payment protocol (SOPMP) withstands the following attacks

Replay Attacks: Replay attack is avoided by Timestamps included in the DIC as an extension, TID, Nonce (N_C) and by means of symmetric keys used (because for every protocol run the symmetric key is different). Timestamps included ensures timeliness and nonce (N_C) ensures freshness of the message thereby avoiding replay attacks. Thus, our proposed protocol is secure against Replay attacks.

Impersonating attack: Since intruder (In) does not have C’s private key it fails to do so. As a result, impersonating attacks fail in our protocol.

Man In The Middle Attack: In our system intruder cannot impersonate any other entity of the system unless (authentication and encryption) keys are compromised. If we assume perfect cryptography and nobody has revealed his/her keys this attack cannot happen. Private key used for generating DSMR is stored in the tamper resistant UICC (which is a SSCD) and Private Key never leaves the WIM of UICC and is securely protected by NRP. Our proposed protocol SOPMP withstands this attack because the intruder (In) does not have receiver’s private key.

Multiprotocol Attack: Our proposed protocols in this chapter withstand this attack because they are successfully verified using Scyther tool (results are given in Appendix C).

Goal 2 (Secrecy):

In our proposed Mobile Payment Protocol (SOPMP) Payment Secrecy is achieved by encrypting the Payment Information (PI) using secret symmetric key which is shared between Client (C) and Issuer (I). Merchant will not be able to decrypt Payment Information (PI) and Order Secrecy is achieved by hashing OI (done by both the Client (C) and Merchant (M)). PG will not know about OI and PI thereby achieving Transaction privacy from PG.

Eavesdropper cannot get MQ and PI because the messages are encrypted using DSMR thereby achieving Transaction privacy from Eavesdropper.

Goal 3 (Uniqueness):

In our proposed Mobile Payment Protocol (SOPMP) every transaction is unique. The uniqueness is achieved due to the generation of fresh Transaction id by the merchant, and its verification by the bank. Every the transaction is also linked to Timestamps and Nonce.

7.4.2. Client Security

Goal 4 (Authentication):

Our proposed Mobile Payment Protocol (SOPMP) uses certification authorities, to certify the authenticity of public keys held by the Merchant, Issuer, Acquirer and PG. The client's conversations are only with the Merchant and the Bank and their authenticity is proved by verifying their certificates. In addition to this client and Issuer shares a symmetric key between them.

Goal 5 (Authorization):

In our proposed Mobile Payment Protocol (SOPMP) the client obtains un forgeable proof of transaction authorization from the Merchant at the end of protocol (i.e. whether the transaction is success or failure).

Goal 6 (Identity Protection from Merchant and Eavesdropper):

In order to achieve Identity Protection from merchant and eavesdropper we have modified the initialization phase of Zuhua Shao et al., 2004. The client will be issued an anonymous identity $Anonid_C$ by the CA after successful verification of client's credentials.

7. 4.3. Bank Security (Issuer, Acquirer and PG)

Goal 7(Authentication):

Our proposed Payment Protocol (SOPMP) has certifying authority to certify the authenticity of the public keys held by the Client, Merchant and Payment Gateway. The client's conversations are only with the Merchant. Client, Merchant and Payment Gateway's authenticity is proved by verifying their certificates. In addition to this client and Issuer share a symmetric key between them.

Goal 8 (Authorization):

The Issuer obtains an authorization proof for transaction from the client in the form of $TransCertC$ (Transaction Certificate signed by client using his private key) and PI (Payment Information) encrypted with the shared symmetric key between Client and Issuer which contains payment details, merchant identity and hashed MQ. The Merchant obtains DIC and PI (Payment Information) encrypted with the shared symmetric key and forwards this information to Acquirer after successful verification of digital signature on DIC, MQ and amount merchant sends the hashed value of OI, which is verified by the Issuer. So Issuer authorizes the transaction after obtaining unforgeable proof from the Client and Merchant.

Goal 9(Prevents double spending, overspending and money laundering):

Issuer (I) keeps $DIC, (PI)K_{CI}, TID, N_C, N_M, T_M, T_C$ in its archives. If the client or merchant tries to double spend the PI, I can detect this from the validity period on the DIC, timestamp and serial number of the certificate. So double spending is avoided in SOPMP by I. If the client or merchant tries to overspend, I avoids them in doing so since it checks Client's balance funds for every transaction, if the check is successful it authorizes the payment else it aborts the transaction thereby preventing overspending. Banks are always involved in every payment transaction thereby preventing money laundering.

Goal 10(Issuer, Acquirer and PG turning malicious):

If any one or all of the entities in the private banking network (Issuer, Acquirer and PG) turns malicious then also they will not succeed in performing transaction on behalf of Client (C) because they have no knowledge about the private key of the Client (C).

7.4.4. Merchant Security

Goal 11(Authentication):

Our proposed Payment Protocol (SOPMP) uses certification authorities, to certify the authenticity of public keys held by the Client, Acquirer, Issuer and PG.

Goal 12 (Authorization):

Merchant checks the authenticity and integrity of the message $MS3$ received from C, validates DIC, checks $MQ, TID, N_C, N_M, T_M, T_C$. If the Merchant is convinced about the TID and MQ then only M authorizes the MQ thereby achieving order secrecy MQ is not known to any of the engaging entities other than C and M. So Merchant authorizes the transaction after obtaining un forgeable proof from the Client.

7.5. Comparative Analysis of SOPMP Protocol with Related Works

We have compared our proposed protocol with other ones; the report of the analysis based on multiple features is presented in table 7.1. The last column results are of our proposed protocol.

Protocols Features	Chen, W.et al. (2010)	Weidar Chen et al. (2010)	M. Massoth and T. Bingel (2009)	T. S. Fun et al. (2008)	Huy Hoang Ngo et al. (2011)	Gianluigi Me, Maurizio A. Strangio et al. (2005)	Mateja Jovanovic and Mario Muñoz Organero (2011)	Huda Ubaya (2012)	Emir Husni, et al. (2008)	SOPMP
Authentication	Yes	Yes	Yes	Yes	Yes	Yes	No	yes	Yes	Yes
Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	No	yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes	Yes	Yes	No	yes	Yes	Yes
Non- Repudiation	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes
Client's credentials are generated using OBKG procedure	No	No	No	No	No	No	No	No	No	Yes
WPKI is implemented in the memory of Mobile Phone	No	No	No	No	No	No	No	No	No	Yes
Ensures end to end Security at application level	No	No	No	No	No	No	No	No	No	Yes
Identity Protection from Eavesdropper	No	No	No	No	yes	No	No	No	No	Yes
Transaction Privacy Protection from Eavesdropper	No	No	No	No	yes	No	No	No	No	Yes
Transaction Privacy protection from PG	No	No	No	No	No	No	No	No	No	Yes
Prevents Double Spending	NR	NR	NR	NR	NR	NR	No	No	No	Yes

Prevents Over spending	NR	NR	NR	NR	NR	NR	No	No	No	Yes
Prevents Money Laundering	NR	NR	NR	NR	NR	NR	No	No	No	Yes
Withstands Replay Attack	Yes	No	No	No	No	Yes	No	No	No	Yes
Withstands Impersonation Attack	Yes	No	No	No	No	No	No	No	No	Yes
Withstands MITM Attack	Yes	No	No	No	No	No	No	No	No	Yes
Formal Verification using AVISPA & SCYTHER TOOL	No	No	No	No	No	No	No	No	No	Yes

7.1. Comparative Analysis of SOPMP with Related Work

7.6. Formal Verification of SOPMP

7.6.1. Formal Verification of SOPMP using BAN Logic

$$Step1: C \rightarrow M : \{DSMR_{C_M}(MS4)\} \; MS4 = \{OI, ID_C, N_C, T_C\}$$

Merchant (M) receives $\{DSMR_{C_M}(MS4), Pubkey_C\}$ from Client (C) and decrypts the message using his private key and gets $MS4$ so from the assumptions **AS1, AS2, AS6**

M **believes** $\{DSMR_{C_M}(MS4)\}$ ……..statement (1)

M validate the public key parameters of Client (C)

If the public key parameters of Client (C) are successfully validated then

M **believes C said** $Pubkey_C$ ………..statement (2)

From **AS6** we conclude M **believes fresh** $Pubkey_C$

From **AS6** and statement (1) we conclude

"M **believes fresh** (T_C)"…statement (3).

"M **believes fresh** (N_C)"………..Statement (4)

From statements 1 and 4

M **believes C said** $\{DSMR_{C_M}(MS4), Pubkey_C\}$ ………..statement (5)

$Step2: M \rightarrow C: \{DSMR_{M_C}(MS5)\}\ MS5 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M\}$

Client (C) receives $\{DSMR_{M_C}(MS5), Pubkey_M\}$ from Merchant (M) and decrypts the message using his private key and gets $MS5$ so from the assumptions **AS1, AS2, AS6**

C **believes** $\{DSMR_{M_C}(MS5)\}$ ………..statement (6)

C validate the public key parameters of Merchant (M)

If the public key parameters of Merchant (M) are successfully validated then

C **believes M said** $Pubkey_M$ ………..statement (7)

From **A6** and statement (6) we conclude

"C **believes fresh** (T_M)"…statement (8).

"C believes fresh (N_M)"………..Statement (9)

From statements 6 and 9

C **believes M said** $\{DSMR_{M_C}(MS5), Pubkey_M\}$ ………..statement (10)

$Step3: C \rightarrow M: \{DSMR_{C_M}(MS6)\}$

$MS6 = \{DIC, Amt, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{CI}}\}$

Merchant (M) receives $\{DSMR_{C_M}(MS6), Pubkey_C\}$ from Client (C) and decrypts the message using his private key and gets $MS6$ so from the assumptions **AS1, AS2, AS6**

M **believes** $\{DSMR_{C_M}(MS6)\}$ ………..statement (11)

M validate the public key parameters of Client (C)

If the public key parameters of Client (C) are successfully validated then

M **believes C said** $Pubkey_C$ ………..statement (12)

From **AS6** we conclude

“M **believes fresh** (T_C)”…Statement (13).

“M **believes fresh** (N_C)”…Statement (14)

From statements 11 and 14

M **believes C said** $\{DSMR_{C_M}(MS6), Pubkey_C\}$ ……..statement (15)

$Step4: M \rightarrow A: \{DSMR_{M_A}(MS7), Pubkey_M\}$

$MS7 = \{DIC, Amt, ID_C, N_C, T_C, N_M, T_M, DIC, (PI)_{K_{CI}}\}$

Acquirer (A) receives $\{DSMR_{M_A}(MS7), Pubkey_M\}$ from Merchant (M) and decrypts the message using his private key and gets $MS7$ so from the assumptions **AS1, AS2, AS6**

A **believes** $\{DSMR_{M_A}(MS7)\}$ ……….statement (16)

Acquirer (A) validate the public key parameters of Merchant (M)

If the public key parameters of Merchant (M) are successfully validated then

A **believes C said** $Pubkey_M$ ………..statement (17)

From **AS6** and statement (16) we conclude

"A **believes fresh** (T_M)"…statement (18).

"A believes fresh (N_M)"………..Statement (19)

Acquirer verifies OI and PI using **AS8, AS9 and AS12** given in Appendix C

From statements 16 and 19

A **believes M said** $\{DSMR_{M_A}(MS7), Pubkey_M\}$ ……..statement (20)

From Step 5, 6 and Step 9 messages are exchanged in private banking network so they are not sent in DSMR format. So we are not formally verifying these steps.

Proof of Step 7

Step 7 is sent through secure private banking network Issuer successfully checks PI using the assumptions **(AS4, AS5, AS10 & AS11)** given in Appendix B.

$Step8: I \rightarrow C: \{DSMR_{I_C}(MS11)\}$

$MS11 = \{TID, Amt, ID_M, Success / Failure\}$

Client (C) receives $\{DSMR_{I_C}(MS11)\}, Pubkey_I$ from Issuer (I) and decrypts the message using his private key and gets $MS11$ so from the assumptions **AS1, AS2, AS6**

C **believes** $\{DSMR_{I_C}(MS11)\}, Pubkey_I$ ………..statement (21)

C validate the public key parameters of Issuer (I)

If the public key parameters of Issuer (I) are successfully validated then

C **believes A said** $Pubkey_I$ ………..statement (22)

From **AS6** we conclude "C **believes fresh** (T_I)"…statement (23).

"C believes fresh (N_I)"……….Statement (24)

From statements 21 and 24

“C **believes I said** $\{DSMR_{I_C}(MS11)\}, Pubkey_I$”……….statement (25)

7.6.2. Formal Verification of SOPMP using AVISPA Tool

The validation of our proposed protocol SOPMP using the OFMC back-end of AVISPA Tool is given in Appendix C as figure 7.2.

The validation of our proposed protocol SOPMP using the CL-AtSe back-end of AVISPA Tool is given in Appendix C as figure 7.3

7.6.3. Formal Verification of SOPMP using Scyther Tool

Result of SOPMP using ‘Verification Claim’ Procedure of Scyther Tool is given in Appendix C as figure 3.6
Result of SOPMP using ‘Automatic Claim’ Procedure of Scyther Tool is given in Appendix C as figure 3.7

7.7. Conclusions

We have proposed a Secure and Optimized Proximity Mobile Payment (SOPMP) Protocol using NFC (Near Field Communication) technology, WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card). In our proposed protocol messages are exchanged in the form of Digital Signature with Message Recovery (DSMR) and merchant sends Invoice in the form of Digital Invoice Certificate (DIC) (which is digitally signed by the merchant). Our proposed SOPMP ensures end-to-end security, consumes fewer resources and is free from attacks (verified using formal tools). We observe that our proposed SOPMP is better than the existing solutions.

Chapter 8: Conclusions and Scope for Future Work

Conclusions

Mobile phones have the power to enable payments anywhere at any time, but the existing mobile payment solutions do not ensure end to end security. This thesis proposes mobile payment protocols suitable for wireless environments which ensure end to end security at application layer for financial transactions. Proposed mobile payment protocols use digital signature, Signcryption and DSMR (Digital Signature with Message Recovery) mechanisms based on ECC (Elliptic Curve Cryptography) and a Symmetric-key cryptography algorithm such as AES and DES. Mobile payment protocols are proposed in tamper-resistant security-sensitive device such as UICC. Proposed mobile payment protocols ensure all the security properties (Authentication, Integrity, Confidentiality and Non Repudiation), achieve Identity protection from merchant and Eavesdropper, achieve Transaction privacy from Eavesdropper and Payment Gateway, achieve Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering. In addition to these our proposed mobile payment protocols withstand Replay, Man in the Middle, Impersonation attacks and Multiprotocol 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 works and found to be performing better than the existing ones in terms of communication cost, computational cost and the number of security properties ensured. Thus this thesis work enhances security in mobile payment area.

Following are the unique characteristics of our proposed protocols in this thesis.

a) **End to End Security**

This thesis uses GPRS, Mobile Agents and NFC (Near Field Communication) technologies as communication channels and uses WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card) for achieving end-to-end security. NFC technology is better when compared with the existing proximity wireless technologies (such as Bluetooth, Wi-Fi and RFID) in terms of Security, Personalization, Flexibility and Power Consumption. One of the major properties of NFC technology is its implicit security because of the short communication distance involved. The close proximity of the two devices makes the signal interception probability very low. The other property is the implicit pairing capability of NFC. Mobile Agent technology has many benefits such as bandwidth conservation, reduction of latency, reduction of completion time, Asynchronous (disconnected) communications. Our proposed mobile payment protocols are proposed in tamper-resistant security-sensitive device such as UICC which is considered as SSCD so the digital signature generated in UICC is recognized automatically as legally equivalent to the handwritten signature. In an SSCD, which is a tamper-proof device the private key never leaves WIM of UICC. Our proposed protocols ensures Authentication, Integrity, Confidentiality and Non Repudiation, achieves Identity protection from merchant and Eavesdropper, achieves Transaction privacy from Eavesdropper and Payment Gateway, achieves Payment Secrecy, Order Secrecy, forward secrecy, and prevents Double Spending, Overspending and Money laundering.

b) Optimal Consumption of Resources

We have used Digital Signature, Signcryption with Forward Secrecy (SFS), DSMR mechanisms based on ECDSA. The advantages of ECC compared to other public-key cryptosystems are higher speeds, lower power consumption, bandwidth savings, storage efficiencies and smaller certificates, so ECC is suitable for resource constrained devices like mobile phones. Digital Signature Message Recovery (DSMR) eliminates the need of adopting PKI cryptosystems thereby reducing the consumption of resources i.e. it consumes less computational and communication cost. DSMR eliminates the need of certificates validation and removes the hurdle of PKI thereby reducing storage space, communication cost and computational cost. Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve is a combination of digital signature and encryption so it consumes fewer resources. Messages are sent in the form of Certificates X.509 SLC (X.509 Short Lived Certificates) and Digital Invoice Certificate (DIC) reducing the consumption of resources which are very scarce in resource constrained devices like mobile phones. So our proposed payment protocols consume fewer resources.

C) Formal Verification

We have carried out formal verification of all the proposed mobile payment protocols in this thesis using BAN logic, AVISPA and Scyther tool. Our proposed protocol withstands Replay, Man in the Middle, Impersonation and Multi Protocol attacks. The security properties of the proposed protocol have been successfully verified using BAN logic, AVISPA and Scyther Tools.

Following are the important findings of this research work

The work carried out during this research has 6 major deliverables whose salient features are summarized as follows

In chapter 2, we have proposed protocols for personalizing UICC by the client, Mobile Payments Application (which is on UICC) by the Bank (Issuer) and a mobile payment protocol between the personalized Mobile Payment Application on UICC and the Bank Server. Our proposed protocols ensure end to end security.

In chapter 3, we have proposed a protocol for the personalization of Mobile Payment Application of Mobile Wallet which is in UICC (Universal Integrated Circuit Card). A mobile payment protocol (SMWP) is proposed which ensures end to end security. Our proposed mobile payment protocol (SMWP) using NFC originating from Mobile wallet (which is in the UICC) to the Issuing bank Server via merchant POS ensures Confidentiality, Authentication, Integrity and Non Repudiation, prevents double spending, over spending and money laundering, and withstands replay, Man in the Middle (MITM), Impersonation attacks and Multi Protocol attacks.

In chapter 4, we have proposed a novel payment instrument namely Mobile traveler’s check (MTC) for Mobile Commerce which is as secure as e-check and can be spent freely anywhere like e-cash. The total time taken for the completion of Issuance, Payment and Deposit phase is 0.84455 seconds. This proves our payment instrument protocol is secure and efficient for resource constrained devices.

In chapter 5, we have proposed a Secure and Optimized Mobile based Merchant Payment (SOMMP) Protocol using Signcryption scheme with Forward Secrecy (SFS). SOMMP makes use of Signcryption scheme with Forward Secrecy (SFS) based on elliptic curve and messages are sent in the form of Certificates X.509 SLC (X.509 Short Lived Certificates). SFS and X.509 SLC reduce the consumption of resources which are very scarce in resource constrained devices like mobile phones. Our proposed protocol ensures all the security properties, consumes fewer resources and is free from attacks.

In chapter 6, we have proposed an Enhanced Mobile SET (EMSET) protocol using Mobile Agent technology and Digital Signature with Message Recovery based on ECDSA mechanism. Mobile Agent technology has many benefits such as bandwidth conservation, reduction of latency, reduction of completion time, Asynchronous (disconnected) communications. Digital Signature with Message Recovery based on ECDSA eliminates the need of adopting PKI cryptosystems. Our proposed protocol EMSET ensures all the security properties consumes fewer resources and is free from attacks.

In chapter 7, we have proposed a Secure and Optimized Proximity Mobile Payment (SOPMP) Protocol using NFC (Near Field Communication) technology, WPKI (Wireless Public Key Infrastructure) and UICC (Universal Integrated Circuit Card) which focuses on end-to-end security, optimal consumption of resources and free from attacks (verified using formal tools). In SOPMP messages are exchanged in the form of Digital Signature with Message Recovery (DSMR) and merchant sends Invoice in the form of Digital Invoice Certificate (DIC) (which is digitally signed by the merchant). Our proposed protocol SOPMP ensures all the security properties consumes fewer resources and is free from attacks.

Scope for Future work

Research and Development is a continuous process. Although the main objectives are covered in this thesis, there are still possibilities for the enhancement of this presented work and future lines of investigation.

a) In order to make the proposed protocols more secure biometric based authentication can be added.

b) Our proposed protocols can be adopted in MANET based payments using Mobile Agents.

c) Our proposed protocols can be adopted for Mobile Health Care.

List of Publications based on the Present Research Work

In Journals

P1. Shaik Shakeel Ahamad, Udgata, S.K. and Sastry, V.N. (2012). A new mobile payment system with formal verification. *International Journal of Internet Technology and Secured Transactions*, 4(1), 71–103. Inderscience Publications **(S.S.Ahamad et al., 2012 a)**

P2. Shaik Shakeel Ahamad., Sastry, V.N. and Udgata, S.K. (2012). Secure and Optimized Mobile based Merchant Payment Protocol using Signcryption. *International Journal of Information Security and Privacy*, 6(2), 64-94. IGI Global Publications **(S.S.Ahamad et al., 2012 b)**

P3. Shaik Shakeel Ahamad., Sastry, V.N. and Udgata, S.K. (2012). A Secure Mobile Wallet Framework with Formal Verification. *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, 4(2), 1-15. IGI Global Publications **(S.S.Ahamad et al., 2012 c)**

P4. Shaik Shakeel Ahamad., Sastry, V.N. and Udgata, S.K. (2013). Secure Mobile Payment Framework based on UICC with Formal Verification. *Int. J. Computational Science and Engineering (Special Issue on Future Trends in Security Issues in Internet and Web Applications)* *(***Accepted) (in Press) 2013)** Inderscience Publications http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijcse **(S.S.Ahamad et al., 2013 a)**

P5. Shaik Shakeel Ahamad., V.N.Sastry & Siba K.Udgata (2013). A Secure and Optimized Proximity Mobile Payment Framework with Formal Verification. *International Journal of E-Services and Mobile Applications (IJESMA)*, 5(4) **((Accepted) (in Press) 2013)** IGI Global Publications. **(S.S.Ahamad et al., 2013 b)**

In Conferences

P6. Shaik Shakeel Ahamad, Sastry, V.N.; Udgata, Siba K. (2012). Enhanced Mobile SET Protocol with Formal Verification. In *Procedings of Third International Conference on Computer and Communication Technology (ICCCT)* (pp.288-293). **(S.S.Ahamad et al., 2012 d)**

P7 Shaik Shakeel Ahamad., V.N.Sastry & Siba K.Udgata. A Secure and Optimized Mobile Payment Framework with Formal Verification. In *Proceedings of International Conference on Security of Internet of Things, Amrita Vishwa Vidyapeetham* ***(Accepted),*** August 16-19, 2012 **(S.S.Ahamad et al., 2012 e)**

P8 Shaik Shakeel Ahamad., and V.N.Sastry. Importance and Issues of Implementing Public Key Infrastructure for Mobile Payments. *Presented at Seventh Meeting of Mobile Payment Forum of India (MPFI) held on April 17, 2010 at IDRBT*, Hyderabad http://www.mpf.org.in/meetings.html

APPENDIX A

Assumptions using BAN Logic for the BAN Logic Proof given in Chapter 2

‘P’ is a set of engaging entities/principals consisting of {C/UC, I, and CA}. The Certification Authority (CA) knows the public keys of every participant and each participant knows the Certification Authority’s public key **(AS1, AS2)**.

AS1. CA **believes** $(\forall P \in \{C, I, CA\} \overset{K_p}{\mapsto} P)$ /* CA (Certifying Authority) believes that K_C, K_I, K_{CA} are the public keys to communicate with the Client C, Issuer I and Certification Authority CA */

AS2. P **believes** $\overset{K_{ca}}{\mapsto} CA$ Where $P \in \{C, I, CA\}$ /* Client C, Issuer I believes that K_{ca} is the public key of the certification authority CA */

AS3.C, UC **believes** $\overset{NRP}{C \rightleftharpoons UC}$ /* The Client C and UICC (UC) **believes** that NRP is a secret that is not known to anyone except C and UC */

AS4.C, I **believes** $\overset{K_{CI}}{C \rightleftharpoons I}$ /* The Client (C) and Issuer (I) believes that K_{CI} is a shared symmetric key between them and is not known to anyone except C and I */

AS5.C **believes fresh**(N_C) , I **believes fresh**(N_I). /* C and I believes that nonce it generated by it say N_C, N_I respectively are fresh, that is, the same nonce is never used in two different execution instances of the protocol */

AS6.C **believes fresh** (T_I') & I **believes fresh** (T_C') /* where T_C', T_I' are the validity periods of Client C & Issuer’s (I) certificates. So C believes I’s certificate is within the validity period & I believes C’s certificate is within the validity period */

AS7.C **believes fresh** (T_I) & I **believes fresh**(T_C) /* T_C & T_I are the timestamps generated by the Client (C) & Issuer (I) for ensuring **timeliness** of the messages */

AS8. $\forall P$, P **believes** CA **controls**$\overset{K_{ca}}{\mapsto Q}$. Where $Q \in \{C, I, CA\}$ **/*** every participant trusts the Certification Authority CA to correctly certify other participants */

AS9. I **believes** (C **controls** PI). /* The Issuer (I) trusts the Client (C) to control the Payment Information*/

APPENDIX B

Assumptions using BAN Logic for the BAN Logic Proofs given for the chapters 3 to 7

AS1. CA **believes** $(\forall P \in \{A, M, C, I, PG\} \overset{K_p}{\mapsto} P)$. /* Certification Authority CA believes that K_A, K_M, K_C, K_I and K_{PG} are the public keys to communicate with Acquirer A, the Merchant M, the Client C, Issuer I and Payment Gateway (PG) respectively */

AS2. P **believes** $(\overset{K_{ca}}{\mapsto} CA)$. Where $\forall P \in \{A, M, C, I, PG\}$ / * Acquirer A, merchant M, Client C, Issuer I and Payment gateway PG believe that K_{ca} is the public key of the certification authority CA */

AS3.C, UC believes $\overset{\boldsymbol{NRP}}{\boldsymbol{C \rightleftharpoons UC}}$ /* The Client C and UICC (UC) believes that NRP is a secret that is not known to anyone except C and UC */

AS4.C, I believes $\overset{\boldsymbol{K_{CI}}}{\boldsymbol{C \rightleftharpoons I}}$ /* The Client (C) and Issuer (I) believes that $\boldsymbol{K_{CI}}$ is a shared symmetric key between them and is not known to anyone except C and I */

AS5.C **believes fresh**(N_C) , I **believes fresh**(N_I), A **believes fresh**(N_A) , M **believes fresh**(N_M). /* C, I, A and M believes that nonce it generated by it say N_C, N_I ,N_A, N_M respectively are fresh, that is, the same nonce is never used in two different execution instances of the protocol */

AS6.C **believes fresh** (T_I'), I **believes fresh**(T_C'), M **believes fresh** (T_C') & C **believes fresh**(T_M') /* whereT_C', T_I', T_M' are the validity periods of Client C, Issuer's (I) and Merchant (M) certificates. So C believes I's and M's certificates are within the validity period & I & M believes C's certificate is within the validity period */

AS7.C **believes fresh** $(T_I\ )$, I **believes fresh**(T_C) , A **believes fresh**(T_M), M **believes fresh**(T_C) & A **believes fresh**(T_M) /* T_C , T_I , $T_M$are the timestamps generated by the Client (C), Issuer (I) and Merchant (M) for ensuring **timeliness** of the messages */

AS8. A **believes** (M **controls** OI_M). /* the Acquirer trusts the merchant to control the Order Information (OI) it issues */

AS9. A **believes** (C **controls** OI_C). /* the acquirer trusts the Client to control the Order Information */

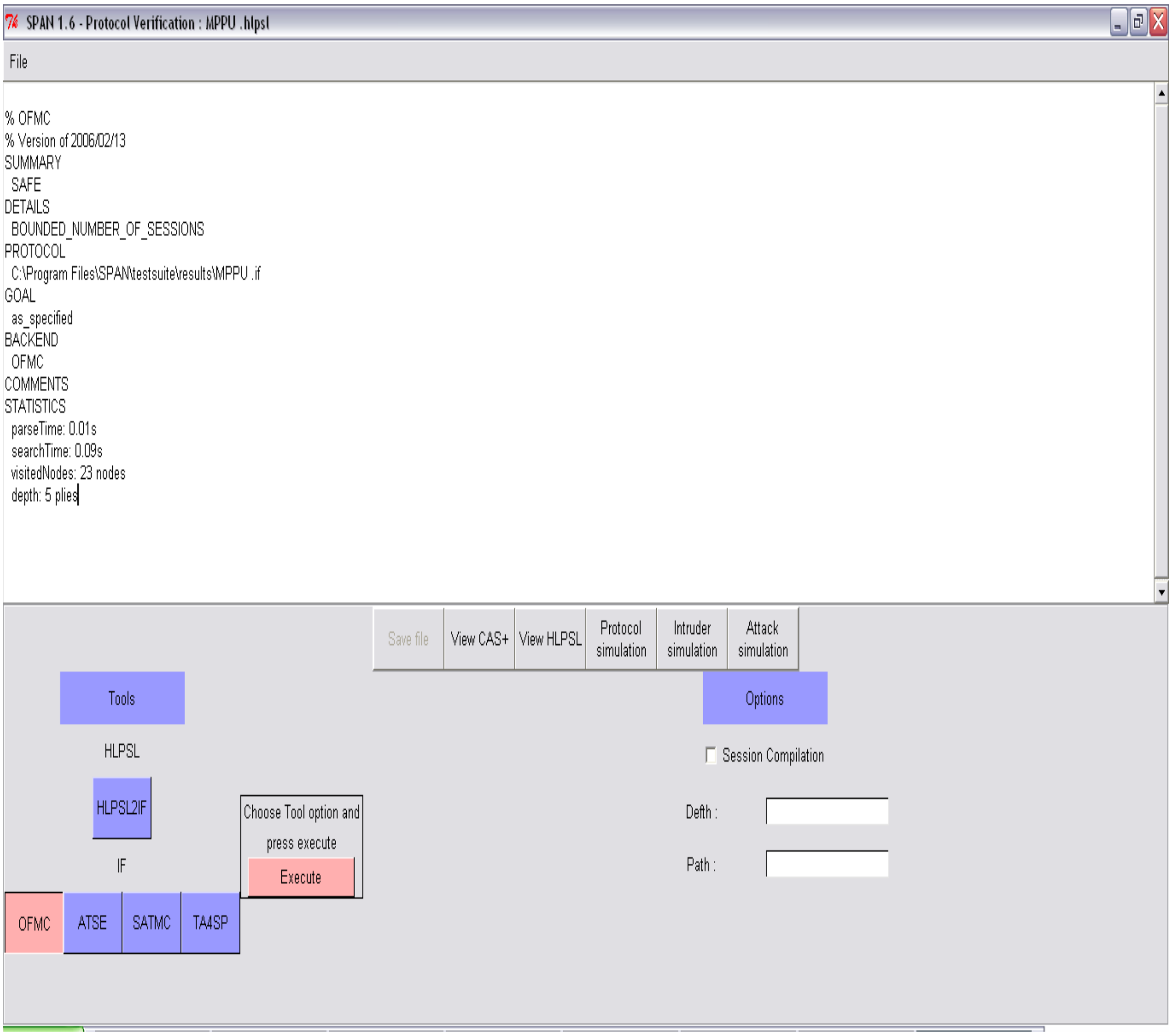
AS10. I **believes** (C **controls** OI_C). /* the Issuer trusts the Client to control the Order Information */

AS11. I **believes** (C **controls** PI). /* the Issuer trusts the Client to control the Payment Information */

AS12. A **believes** (C **controls** PI). /* the Acquirer trusts the Client to control the Payment Information.

APPENDIX C

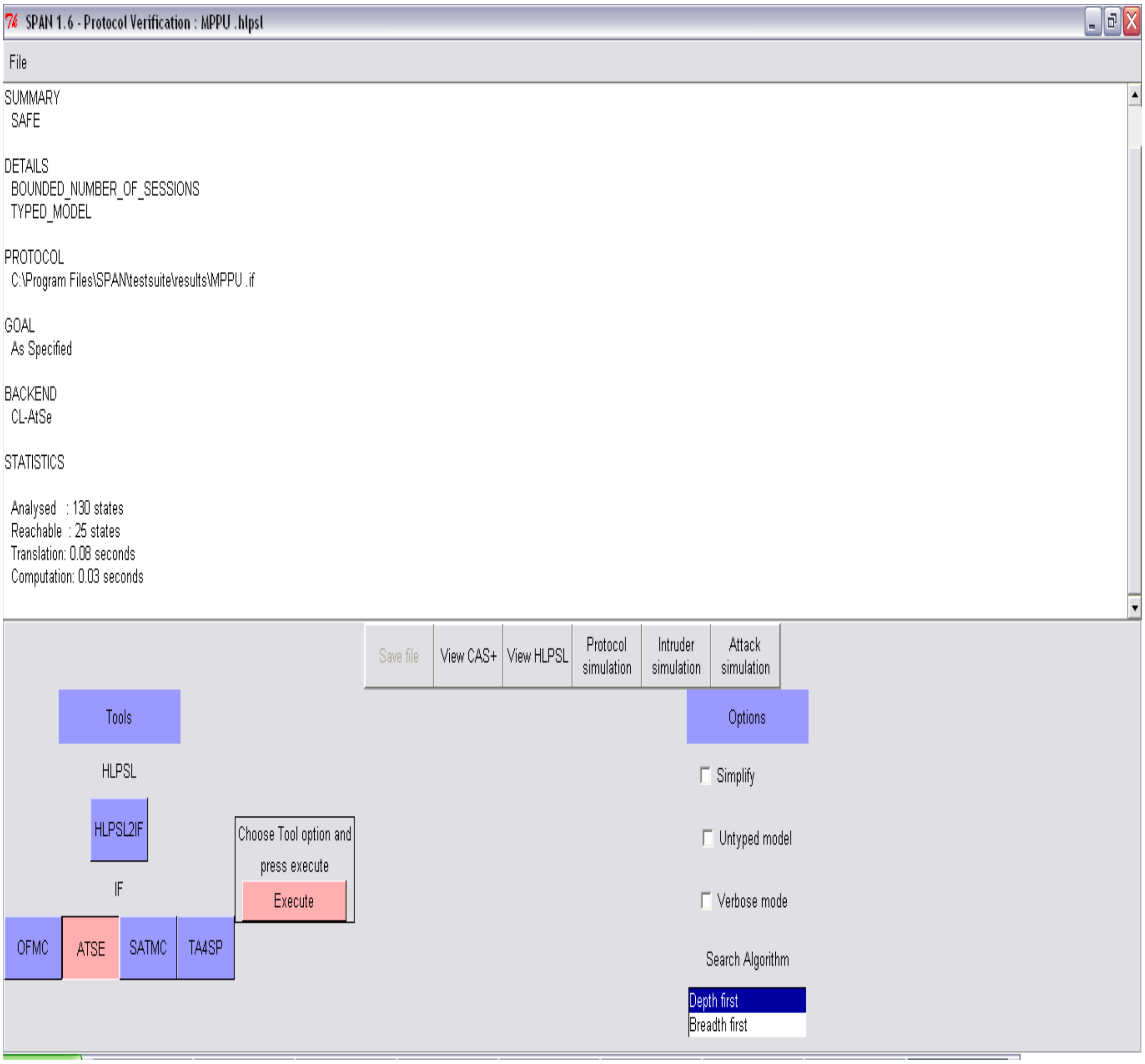
Formal Verification Results of Chapter 2



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\MPPU .if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.01s
  searchTime: 0.09s
  visitedNodes: 23 nodes
  depth: 5 plies
```

Figure 2.3: Result of MPPU using OFMC backend of AVISPA Tool

The validation of our proposed protocol (MPPU) specification, using the OFMC back-end in AVISPA Tool produces the following output shown in figure 2.3.



```
SPAN 1.6 - Protocol Verification : MPPU .hlpsl
File

SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 C:\Program Files\SPAN\testsuite\results\MPPU .if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 130 states
 Reachable  : 25 states
 Translation: 0.08 seconds
 Computation: 0.03 seconds
```

Options

☐ Simplify

☐ Untyped model

☐ Verbose mode

Search Algorithm

Depth first

Breadth first

Figure 2.4: Result of MPPU using CL-AtSe backend of AVISPA Tool

The validation of our proposed protocol (MPPU) specification, using the CL-Atse back-end in AVISPA Tool produces the following output shown in figure 2.4.

Scyther is a tool for the verification, falsification and the analysis of security protocols, where it is assumed that all cryptographic functions are perfect. Scyther provides a number of novel features that include the possibility of unbounded verification with guaranteed termination, analysis of infinite sets of traces in terms of patterns and support for multi-protocol analysis. Scyther is based on a pattern refinement algorithm, providing concise representations of (infinite) sets of traces. This allows the tool to assist in the analysis of classes of attacks and possible protocol behavior, or to prove correctness for an unbounded number of protocol sessions.

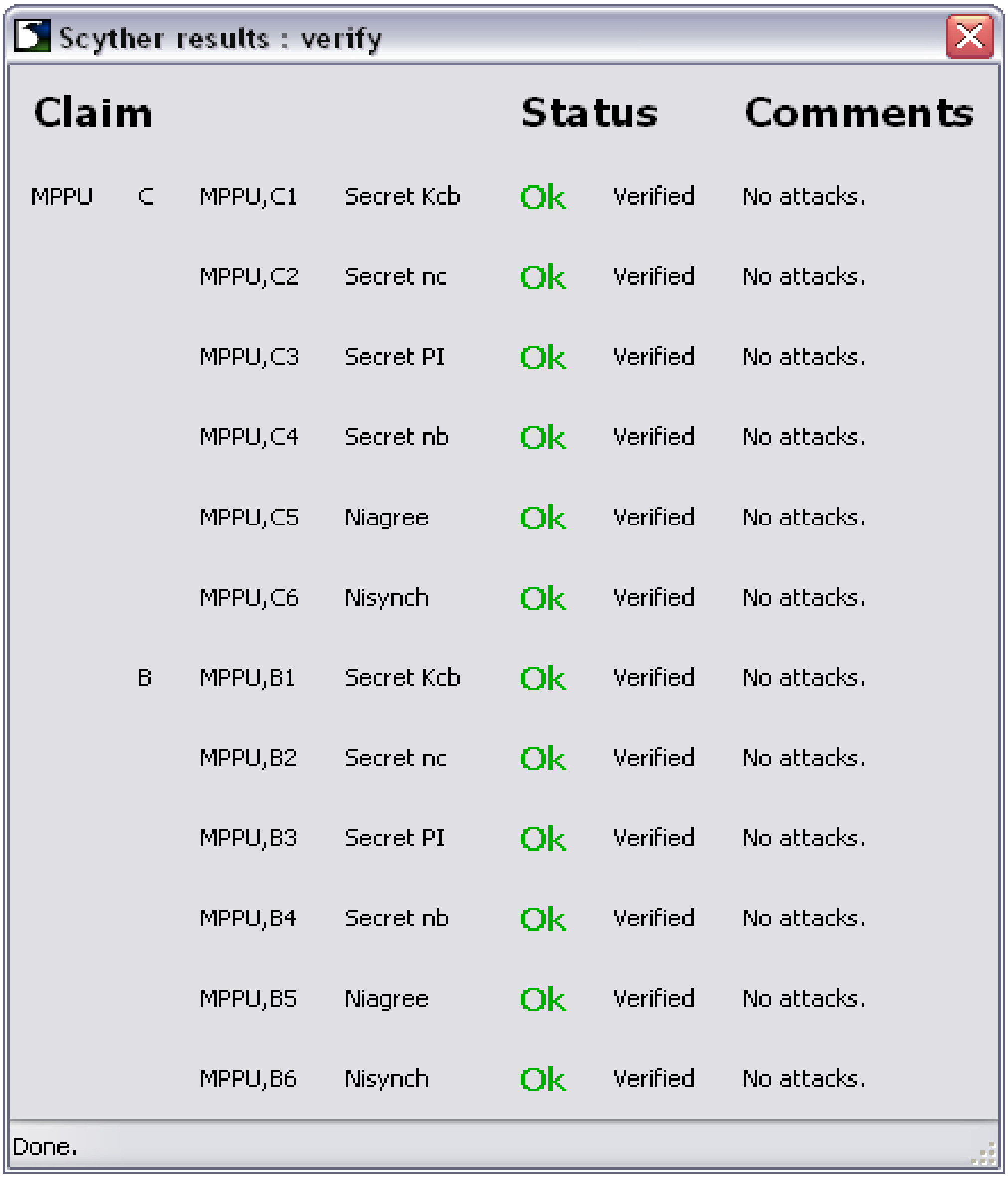
```
/* Mobile Payment Framework based on UICC */
/* Authentication and Key Agreement Protocol */
/* Mobile Payment Protocol */
// PKI
const pk: Function;
secret sk: Function;
inversekeys (pk,sk);
usertype Timestamp;
usertype success;
usertype PI,PayeeID,amt,Ack,Tc,Tb;
// Protocol description
protocol MPPU(C,B)
{
role C
{
const nc: Nonce;
var nb: Nonce;
const Kcb:SessionKey;
/* Authentication and Key Agreement Protocol */
send_1 (C,B, { nc,PI,Tc }pk(B) );
read_2 (B,C, { nc,nb,PI,Kcb,Tb,Tc}pk(C) );
```

```
send_3 (C,B, { Tc,Tb,nc,nb,Ack}Kcb);
/* Mobile Payment Protocol */
send_4 (C,B, { Tc,Tb,nc,nb,PI,PayeeID,amt}Kcb);
read_5 (B,C, { Tc,Tb,nc,nb,PayeeID,amt,success}Kcb);
claim_C1 (C, Secret, Kcb );
claim_C2 (C, Secret, nc);
claim_C3 (C, Secret, PI);
claim_C4 (C, Secret, nb);
claim_C5 (C, Niagree);
claim_C6 (C, Nisynch);
}
role B
{
const nb: Nonce;
var nc: Nonce;
const Kcb:SessionKey;
/* Authentication and Key Agreement Protocol */
read_1 (C,B, { nc,PI,Tc }pk(B) );
send_2 (B,C, { nc,nb,PI,Kcb,Tb,Tc}pk(C) );
read_3 (C,B, { Tc,Tb,nc,nb,Ack}Kcb);
/* Mobile Payment Protocol */
read_4 (C,B, { Tc,Tb,nc,nb,PI,PayeeID,amt}Kcb);
send_5 (B,C, { Tc,Tb,nc,nb,PayeeID,amt,success}Kcb);
claim_B3 (B, Secret, PI);
claim_B1 (B, Secret, Kcb );
claim_B2 (B, Secret, nc);
claim_B4 (B, Secret, nb);
```

claim_B5 (B, Niagree);

claim_B6 (B, Nisynch);

}}

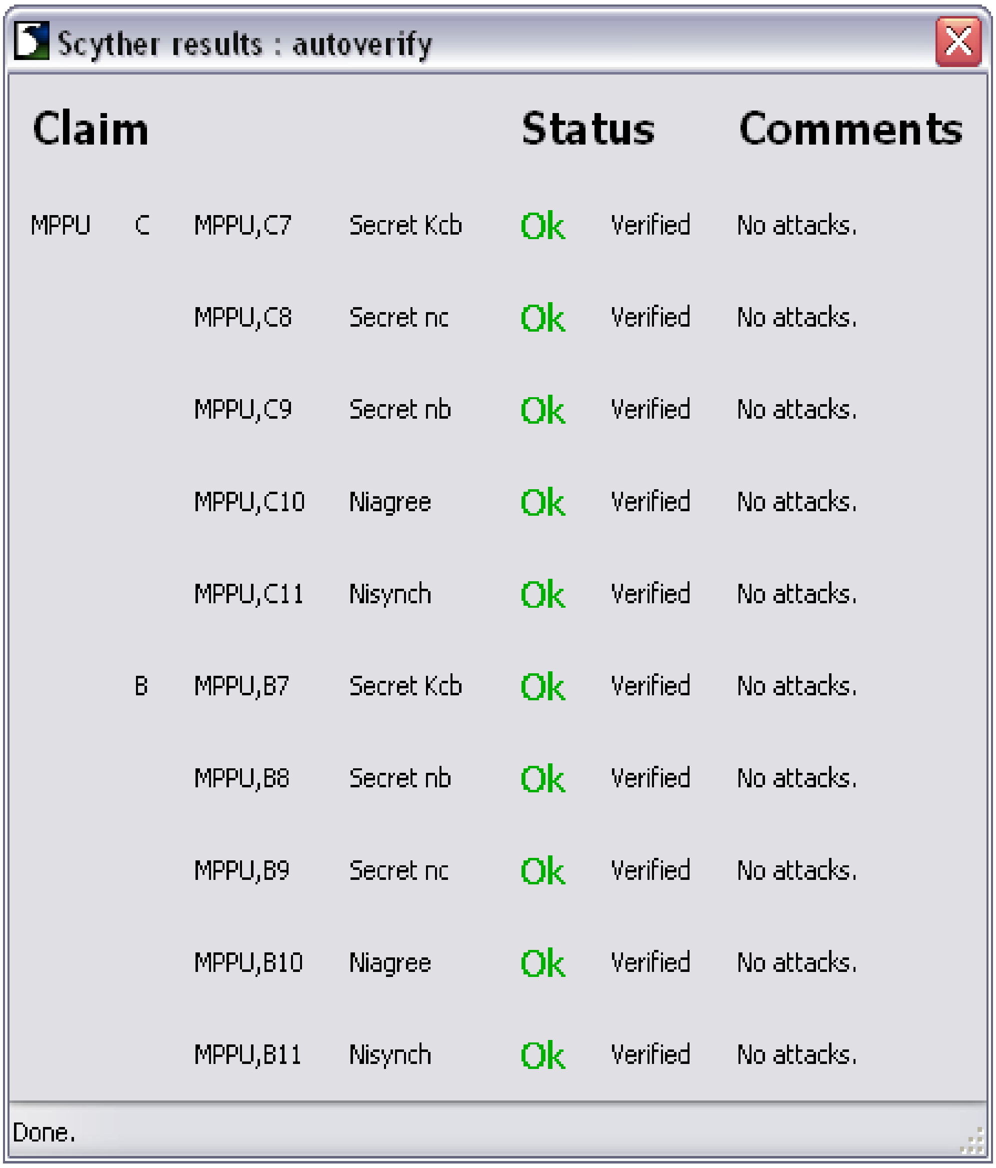


Claim				Status		Comments
MPPU	C	MPPU,C1	Secret Kcb	Ok	Verified	No attacks.
		MPPU,C2	Secret nc	Ok	Verified	No attacks.
		MPPU,C3	Secret PI	Ok	Verified	No attacks.
		MPPU,C4	Secret nb	Ok	Verified	No attacks.
		MPPU,C5	Niagree	Ok	Verified	No attacks.
		MPPU,C6	Nisynch	Ok	Verified	No attacks.
	B	MPPU,B1	Secret Kcb	Ok	Verified	No attacks.
		MPPU,B2	Secret nc	Ok	Verified	No attacks.
		MPPU,B3	Secret PI	Ok	Verified	No attacks.
		MPPU,B4	Secret nb	Ok	Verified	No attacks.
		MPPU,B5	Niagree	Ok	Verified	No attacks.
		MPPU,B6	Nisynch	Ok	Verified	No attacks.

Done.

Figure 2.5: Result of MPPU Protocol using "Verification Claim" Procedure of Scyther Tool

The result window shows that the secrecy of Kcb, nc, PI, nb, the claim for non-injective agreement and non-injective synchronization at both Client and Bank end are successfully verified



Scyther results : autoverify

Claim				Status		Comments
MPPU	C	MPPU,C7	Secret Kcb	Ok	Verified	No attacks.
		MPPU,C8	Secret nc	Ok	Verified	No attacks.
		MPPU,C9	Secret nb	Ok	Verified	No attacks.
		MPPU,C10	Niagree	Ok	Verified	No attacks.
		MPPU,C11	Nisynch	Ok	Verified	No attacks.
	B	MPPU,B7	Secret Kcb	Ok	Verified	No attacks.
		MPPU,B8	Secret nb	Ok	Verified	No attacks.
		MPPU,B9	Secret nc	Ok	Verified	No attacks.
		MPPU,B10	Niagree	Ok	Verified	No attacks.
		MPPU,B11	Nisynch	Ok	Verified	No attacks.

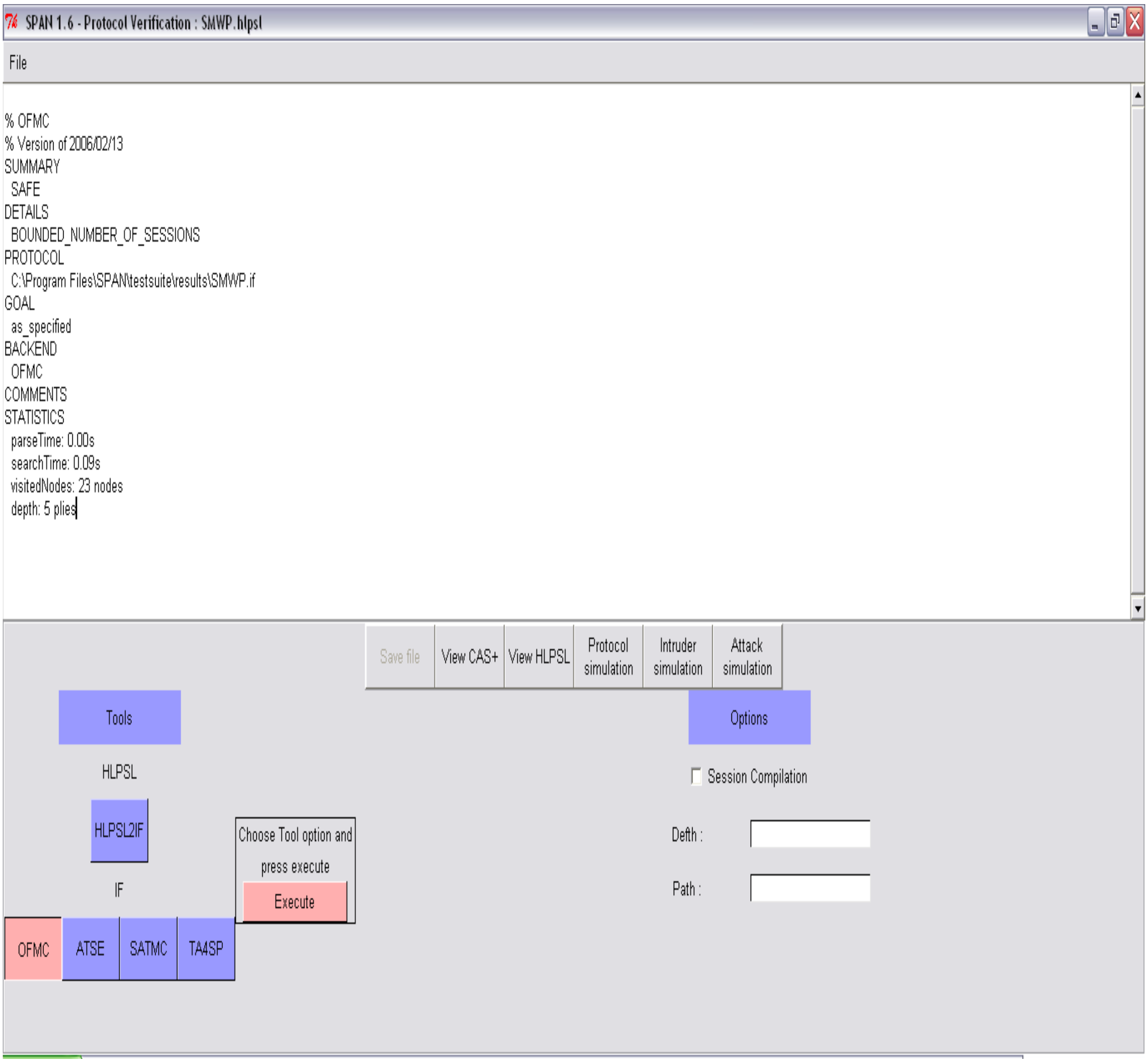
Done.

Figure 2.6: Result of MPPU Protocol using "Automatic Claim" Procedure of Scyther Tool

The result window shows that the secrecy of Kcb, secrecy of nc, nb, the claim for non-injective agreement and non-injective synchronization at both Client and Bank are successfully verified, where Kcb, nc, nb is shared symmetric key between Client and Bank, na and nb are nonce.

Formal Verification Results of Chapter 3 (SMWP)

The validation of our proposed protocol (SMWP) specification, using the OFMC back-end in AVISPA Tool produces the following output shown in figure 3.4

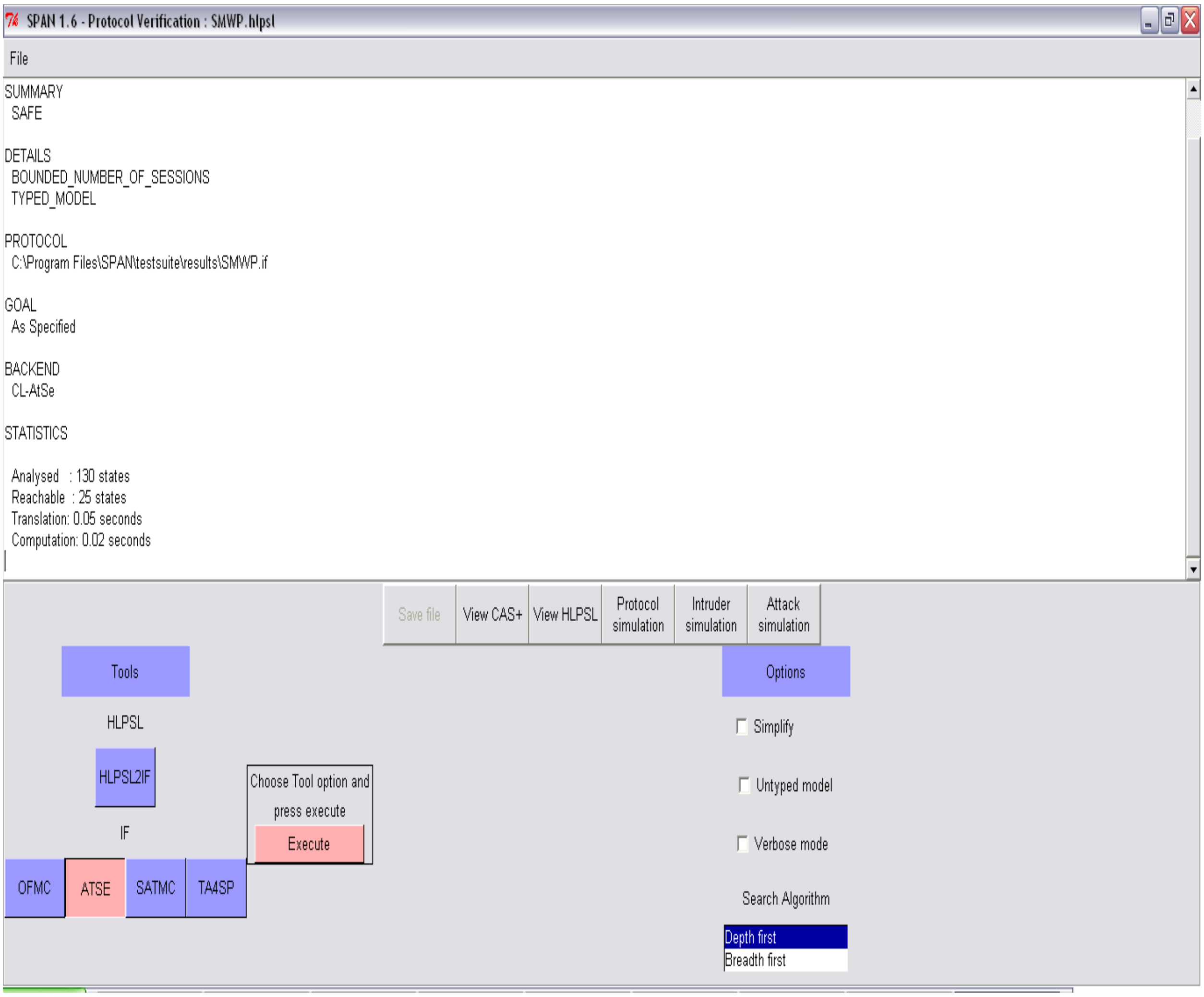


```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SMWP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.09s
  visitedNodes: 23 nodes
  depth: 5 plies
```

Choose Tool option and press execute

Figure 3.4: Result of SMWP Protocol using "OFMC" backend of AVISPA Tool

The validation of our proposed protocol SMWP using the CL-AtSe back-end in AVISPA Tool produces the following output shown in figure 3.5



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SMWP.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 130 states
  Reachable  : 25 states
  Translation: 0.05 seconds
  Computation: 0.02 seconds
```

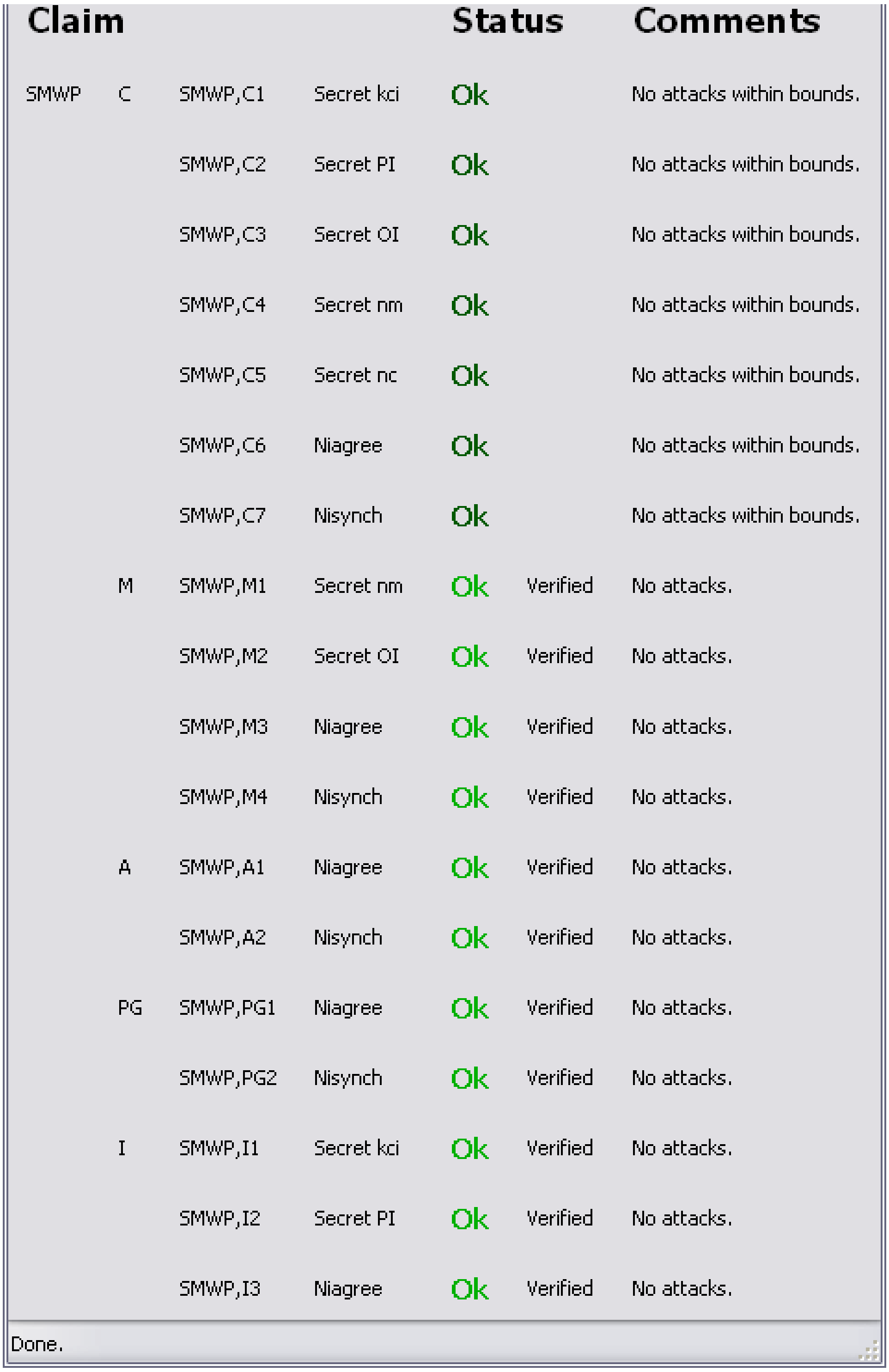
☐ Simplify

☐ Untyped model

☐ Verbose mode

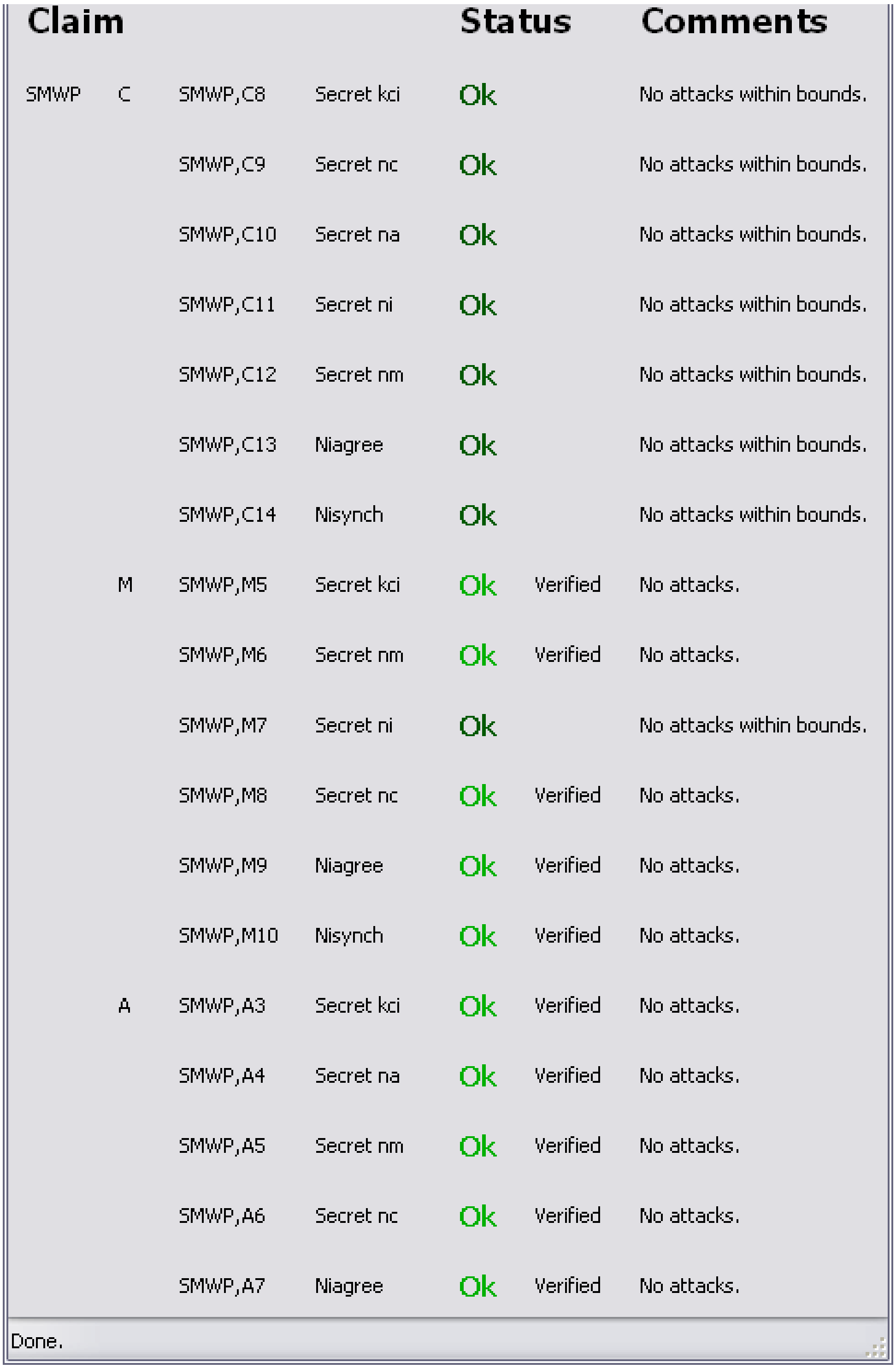
Search Algorithm
[✓ Depth first]
Breadth first

Figure 3.5: Result of SMWP Protocol using "CL-AtSe" backend of AVISPA Tool



```
Claim                                   Status       Comments

SMWP   C    SMWP,C1    Secret kci       Ok                    No attacks within bounds.
            SMWP,C2    Secret PI        Ok                    No attacks within bounds.
            SMWP,C3    Secret OI        Ok                    No attacks within bounds.
            SMWP,C4    Secret nm        Ok                    No attacks within bounds.
            SMWP,C5    Secret nc        Ok                    No attacks within bounds.
            SMWP,C6    Niagree          Ok                    No attacks within bounds.
            SMWP,C7    Nisynch          Ok                    No attacks within bounds.
       M    SMWP,M1    Secret nm        Ok     Verified       No attacks.
            SMWP,M2    Secret OI        Ok     Verified       No attacks.
            SMWP,M3    Niagree          Ok     Verified       No attacks.
            SMWP,M4    Nisynch          Ok     Verified       No attacks.
       A    SMWP,A1    Niagree          Ok     Verified       No attacks.
            SMWP,A2    Nisynch          Ok     Verified       No attacks.
       PG   SMWP,PG1   Niagree          Ok     Verified       No attacks.
            SMWP,PG2   Nisynch          Ok     Verified       No attacks.
       I    SMWP,I1    Secret kci       Ok     Verified       No attacks.
            SMWP,I2    Secret PI        Ok     Verified       No attacks.
            SMWP,I3    Niagree          Ok     Verified       No attacks.
Done.
```

Figure 3.6: Result of SMWP Protocol using "Verification Claim" procedure of Scyther Tool



Claim				Status		Comments
SMWP	C	SMWP,C8	Secret kci	Ok		No attacks within bounds.
		SMWP,C9	Secret nc	Ok		No attacks within bounds.
		SMWP,C10	Secret na	Ok		No attacks within bounds.
		SMWP,C11	Secret ni	Ok		No attacks within bounds.
		SMWP,C12	Secret nm	Ok		No attacks within bounds.
		SMWP,C13	Niagree	Ok		No attacks within bounds.
		SMWP,C14	Nisynch	Ok		No attacks within bounds.
	M	SMWP,M5	Secret kci	Ok	Verified	No attacks.
		SMWP,M6	Secret nm	Ok	Verified	No attacks.
		SMWP,M7	Secret ni	Ok		No attacks within bounds.
		SMWP,M8	Secret nc	Ok	Verified	No attacks.
		SMWP,M9	Niagree	Ok	Verified	No attacks.
		SMWP,M10	Nisynch	Ok	Verified	No attacks.
	A	SMWP,A3	Secret kci	Ok	Verified	No attacks.
		SMWP,A4	Secret na	Ok	Verified	No attacks.
		SMWP,A5	Secret nm	Ok	Verified	No attacks.
		SMWP,A6	Secret nc	Ok	Verified	No attacks.
		SMWP,A7	Niagree	Ok	Verified	No attacks.

Done.

Figure 3.7: Result of SMWP Protocol using "Automatic Claim" procedure of Scyther Tool

SMWP Code in SPDL of Scyther Tool

```
const pk: Function;
secret sk: Function;
inversekeys(pk,sk);
usertype Timestamp;
usertype OI,PI,HOI,TID,Amt,Tc,Tm,LI,AuthPI;
usertype success;
usertype Sessionkey;
protocol SMWP(C,M,A,PG,I)
{
role C
{
const nc: Nonce;
        var nm,ni,na: Nonce;
        const kci: SessionKey;
        send_1(C,M,{{nc,Tc}sk(C)}pk(M));
        read_2(M,C,{{OI,HOI,TID,Tc,nc,Amt,LI,nm,Tm}sk(M)}pk(C));
        send_3(C,M,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(C)}pk(M));
        read_8(I,C,{{AuthPI,TID}sk(I)}pk(C));
        read_11(M,C,{{success,Amt,TID}sk(M)}pk(C));
        claim_C1(C,Secret,kci);
        claim_C2(C,Secret,PI);
         claim_C3(C,Secret,OI);
        claim_C4(C,Secret,nm);
         claim_C5(C,Secret,nc);
        claim_C6 (C,Niagree);
        claim_C7(C,Nisynch);
   }
```

```
role M
{
        const  nm: Nonce;
        var  nc,ni: Nonce;
         const kci: SessionKey;
         read_1(C,M,{{nc,Tc}sk(C)}pk(M));
         send_2(M,C,{{OI,HOI,TID,Tc,nc,Amt,LI,nm,Tm}sk(M)}pk(C));
         read_3(C,M,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(C)}pk(M));
         send_4(M,A,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(M)}pk(A));
         read_10(A,M,{{success,Amt,TID}sk(A)}pk(M));
         send_11(M,C,{{success,Amt,TID}sk(M)}pk(C));
         claim_M1(M,Secret,nm);
          claim_M2(M,Secret,OI);
          claim_M3(M,Niagree);
          claim_M4(M,Nisynch);
       }
role A
      {
         const na: Nonce;
         var nc,nm: Nonce;
         const kci: SessionKey;
         read_4(M,A,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(M)}pk(A));
         send_5(A,PG,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(A)}pk(PG));
         read_9(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
         send_10(A,M,{{success,Amt,TID}sk(A)}pk(M));
         claim_A1(A,Niagree);
        claim_A2(A,Nisynch);
}
```

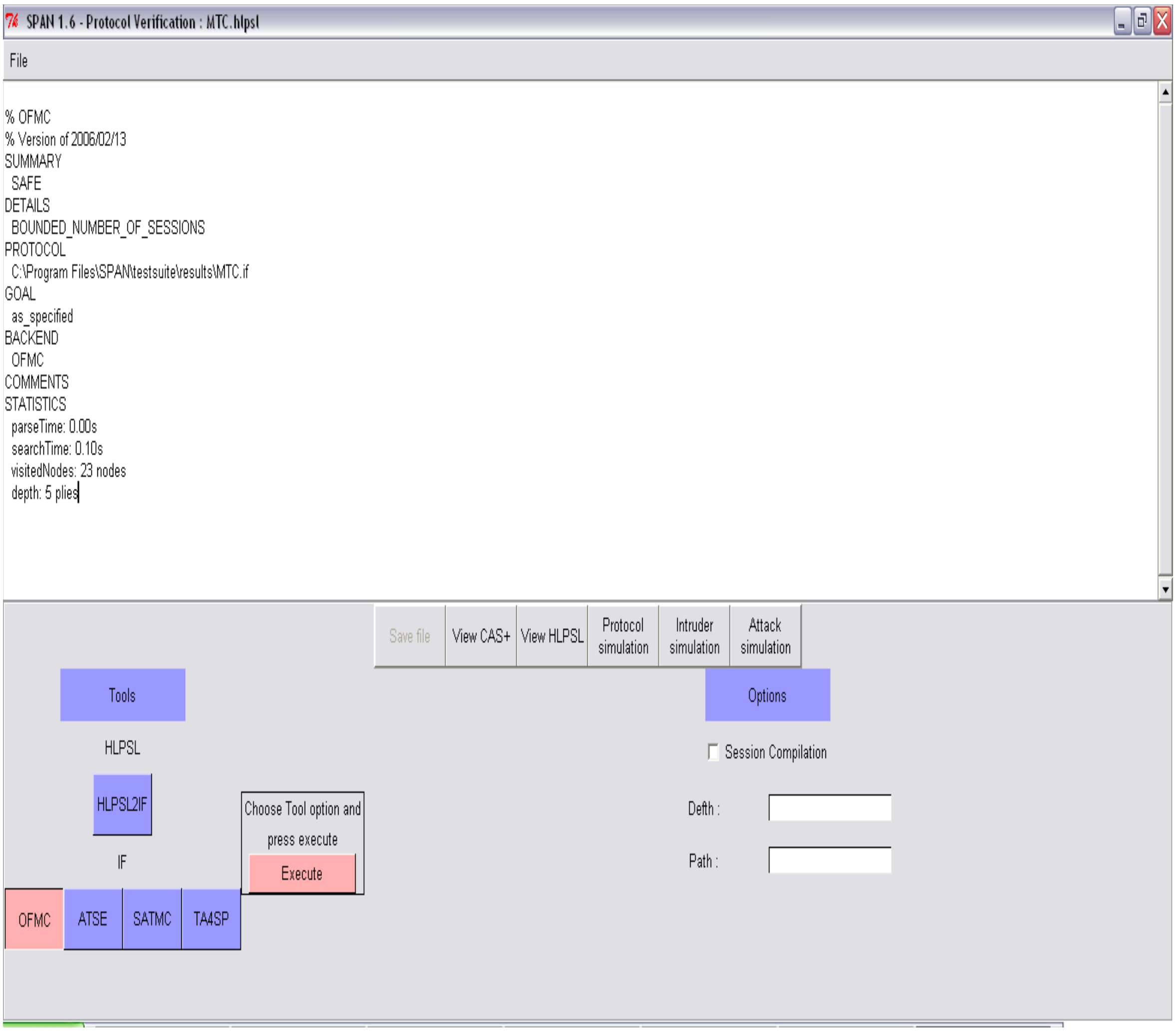
```
  role PG
{
        const npg: Nonce;
        var nc,nm: Nonce;
        const kci: SessionKey;
        read_5(A,PG,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(A)}pk(PG));
        send_6(PG,I,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(PG)}pk(I));
        read_7(I,PG,{{AuthPI,TID}sk(I)}pk(PG));
        send_9(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
        claim_PG1(PG,Niagree);
       claim_PG2(PG,Nisynch);
      }
 role I
{
        const ni: Nonce;
        var nc,nm: Nonce;
        const kci: SessionKey;
        read_6(PG,I,{{HOI,TID,Tc,Amt,LI,nc,nm,Tm,{PI}kci,TID}sk(PG)}pk(I));
        send_7(I,PG,{{AuthPI,TID}sk(I)}pk(PG));
        send_8(I,C,{{AuthPI,TID}sk(I)}pk(C));
        claim_I1 (I,Secret,kci);
         claim_I2(I,Secret,PI);
        claim_I3 (I,Niagree);
       claim_I4 (I,Nisynch);
      }
  }
const C,M,A,PG,I,E: Agent;
untrusted E;
```

const ne: Nonce;

const te: Timestamp;

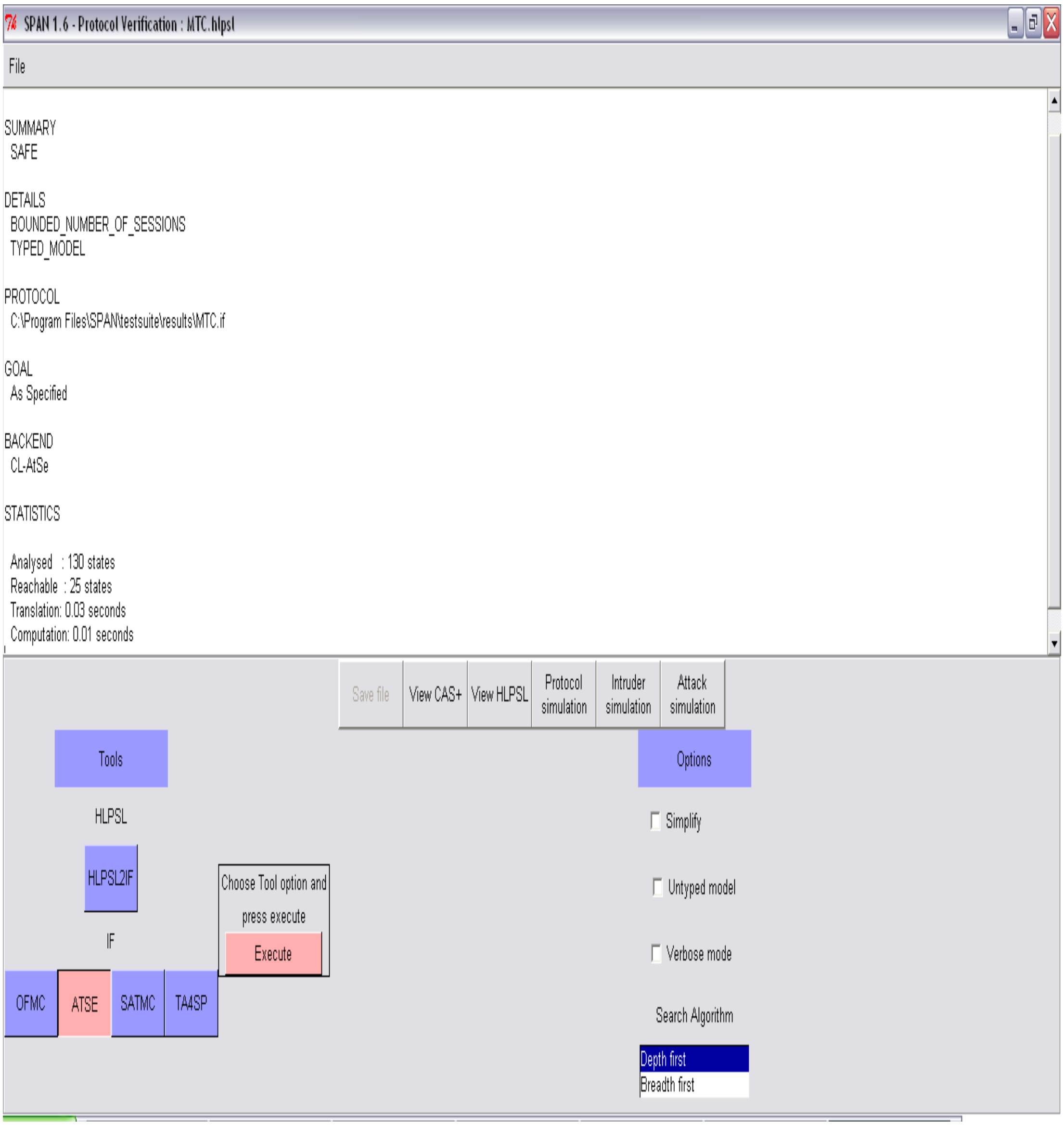
compromised sk(E);

Formal Verification Results of Chapter 4 (MTC)



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\MTC.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 23 nodes
  depth: 5 plies
```

Figure 4.2. Result of MTC using OFMC backend of AVISPA Tool



```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 C:\Program Files\SPAN\testsuite\results\MTC.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 130 states
 Reachable  : 25 states
 Translation: 0.03 seconds
 Computation: 0.01 seconds
```

□ Simplify

□ Untyped model

□ Verbose mode

Search Algorithm

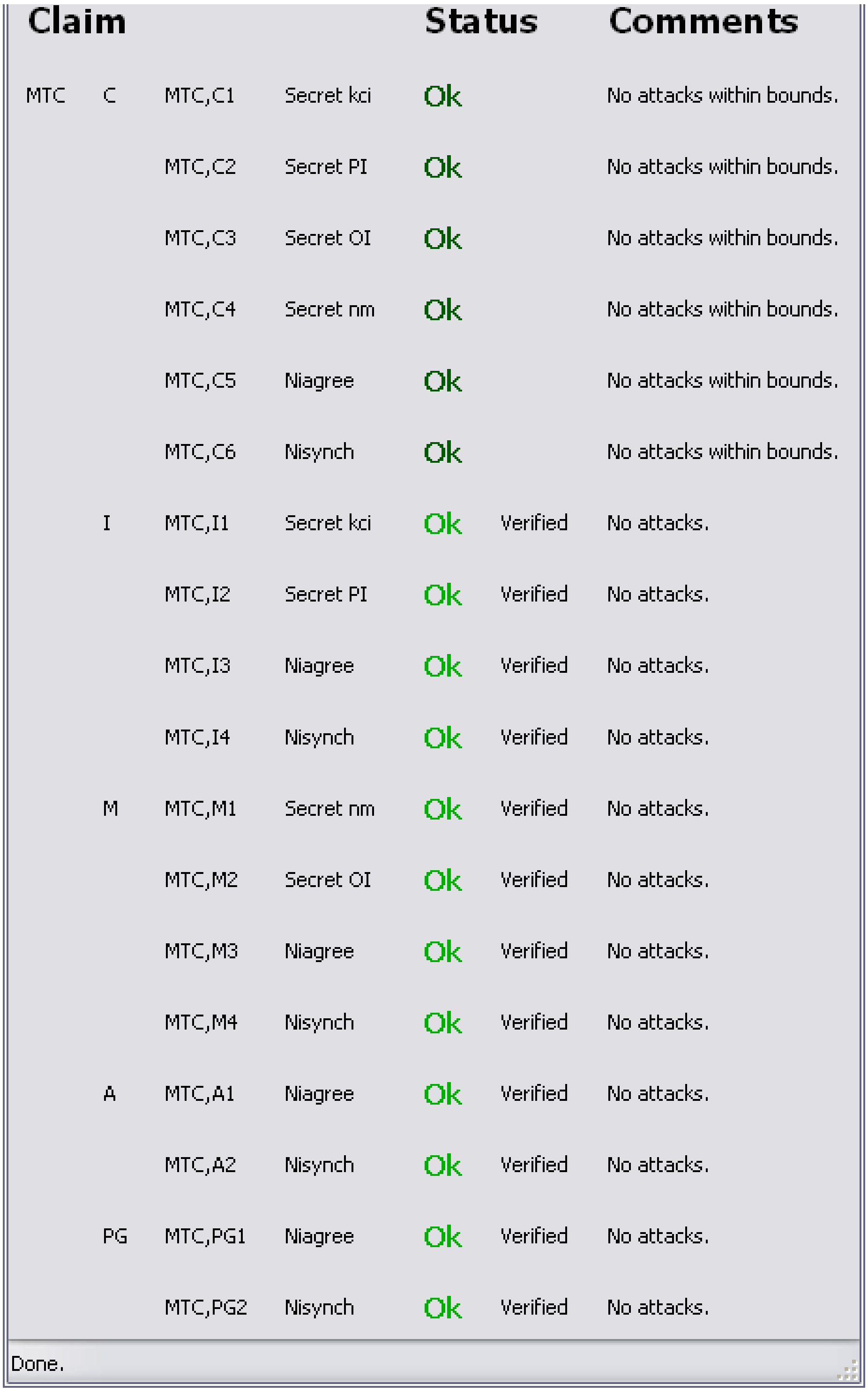
[✓ Depth first]
Breadth first

Figure 4.3: Result of MTC using ATSE backend of AVISPA Tool

```
/* Code of Mobile Traveler's Check MTC Protocol in SPDL of Scyther Tool */

const pk: Function;

secret sk: Function;

inversekeys(pk,sk);

usertype Timestamp;

usertype OI,PI,Amt,ENDORc,Tc,Ti,Tm,IDENi,PR,FV,MTC,SNO,UTIm,HOIc,HOIm;

usertype Success;

usertype Sessionkey;
```

```
protocol MTC(C,I,M,A,PG)
{
    role C
{
        const nc: Nonce;
        var nm,ni: Nonce;
        const kci: SessionKey;
        send_1(C,I,{{ENDORc,Tc,Amt,nc}sk(C)}pk(I));
        read_2(I,C,{{ENDORc,IDENi,Ti,Amt,nc,ni}sk(I)}pk(C));
       send_3(C,I,{{PR,Amt,FV,nc,ni,Tc}sk(C)}pk(I));
       read_4(I,C,{{MTC,SNO,nc,ni,Ti}sk(I)}pk(C));
       send_5(C,M,{{{PI}kci,ENDORc,HOIc,UTIm,Tc,Amt,nc}sk(C)}pk(M));
       read_10(I,C,{{Success,UTIm}sk(I)}pk(C));
        claim_C1(C,Secret,kci);
        claim_C2(C,Secret,PI);
         claim_C3(C,Secret,OI);
        claim_C4(C,Secret,nm);
        claim_C5 (C,Niagree);
       claim_C6(C,Nisynch);
    }
    role I
    {
       const  ni: Nonce;
       var  nc,nm: Nonce;
        const kci: SessionKey;
        read_1(C,I,{{ENDORc,Tc,Amt,nc}sk(C)}pk(I));
        send_2(I,C,{{ENDORc,IDENi,Ti,Amt,nc,ni}sk(I)}pk(C));
        read_3 (C,I,{{PR,Amt,FV,nc,ni,Tc}sk(C)}pk(I));
        send_4(I,C,{{MTC,SNO,nc,ni,Ti}sk(I)}pk(C));
        read_8(PG,I,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(PG)}pk(I));
        send_9(I,PG,{{Success,UTIm}sk(I)}pk(PG));
        send_10(I,C,{{Success,UTIm}sk(I)}pk(C));
        claim_I1 (I,Secret,kci);
         claim_I2(I,Secret,PI);
        claim_I3 (I,Niagree);
       claim_I4 (I,Nisynch);
      }
     role M
     {
        const nm: Nonce;
        var nc,nb: Nonce;
        const kci: SessionKey;
       read_5(C,M,{{{PI}kci,ENDORc,HOIc,UTIm,Tc,Amt,nc}sk(C)}pk(M));
       send_6(M,A,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(M)}pk(A));
       read_12(A,M,{{Success,UTIm}sk(A)}pk(M));
       claim_M1(M,Secret,nm);
       claim_M2(M,Secret,OI);
        claim_M3(M,Niagree);
        claim_M4(M,Nisynch);
         }
role A
     {
        const na: Nonce;
        var nc,nm: Nonce;
        const kci: SessionKey;
        read_6(M,A,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(M)}pk(A));
```

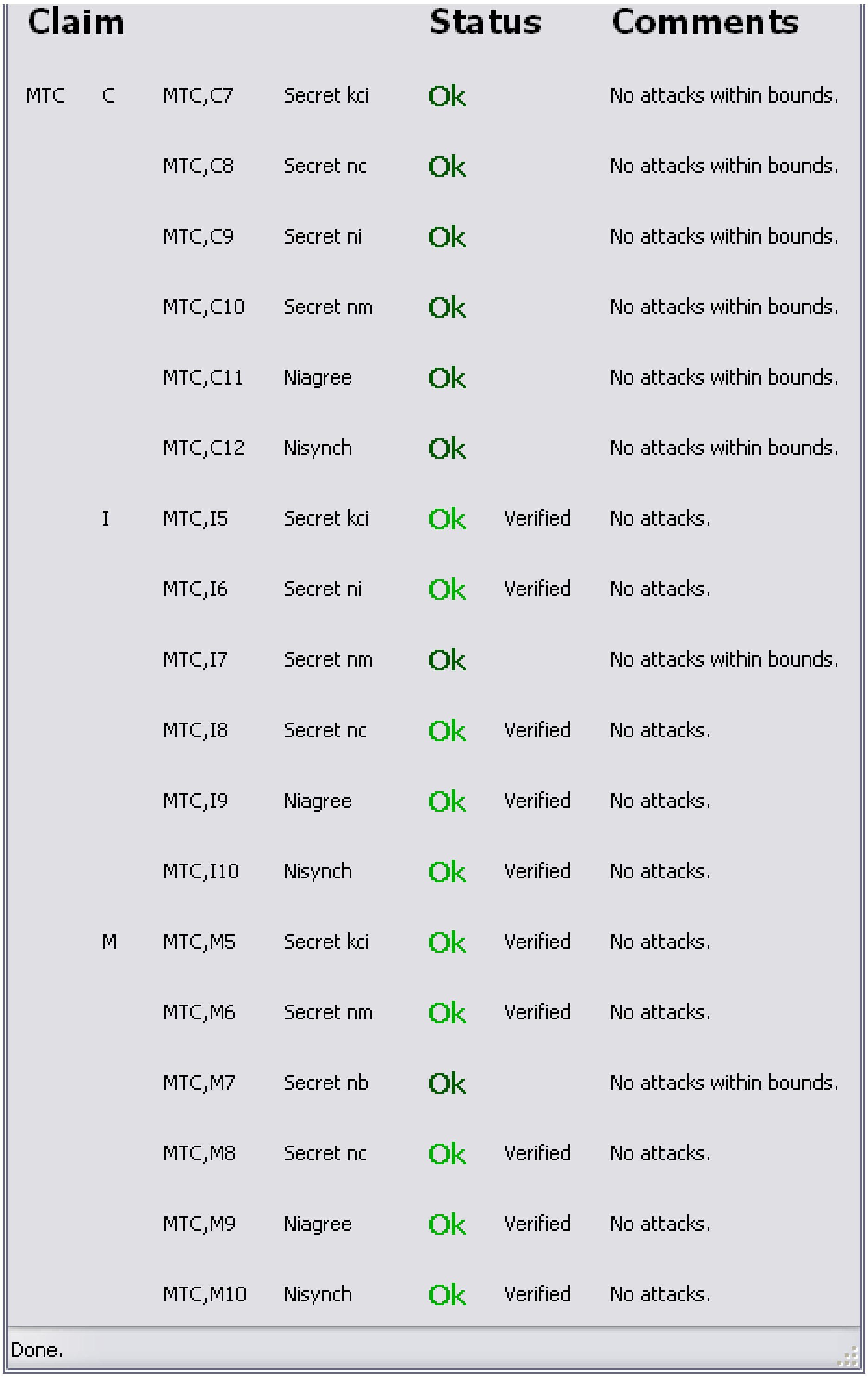
```
        send_7(A,PG,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(A)}pk(PG));
        read_11(PG,A,{{Success,UTIm}sk(PG)}pk(A));
        send_12(A,M,{{Success,UTIm}sk(A)}pk(M));
        claim_A1(A,Niagree);
       claim_A2(A,Nisynch);
      }
role PG
{        const npg: Nonce;
        var nc,nm: Nonce;
        const kci: SessionKey;
        read_7(A,PG,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(A)}pk(PG));
        send_8(PG,I,{{{PI}kci,ENDORc,HOIc,HOIm,UTIm,Tc,Amt,nc}sk(PG)}pk(I));
        read_9(I,PG,{{Success,UTIm}sk(I)}pk(PG));
        send_11(PG,A,{{Success,UTIm}sk(PG)}pk(A));
        claim_PG1(PG,Niagree);
       claim_PG2(PG,Nisynch);
      }
const C,I,M,A,PG,E: Agent;
untrusted E;
const ne: Nonce;
const te: Timestamp;
compromised sk(E);
```



Claim				Status		Comments
MTC	C	MTC,C1	Secret kci	Ok		No attacks within bounds.
		MTC,C2	Secret PI	Ok		No attacks within bounds.
		MTC,C3	Secret OI	Ok		No attacks within bounds.
		MTC,C4	Secret nm	Ok		No attacks within bounds.
		MTC,C5	Niagree	Ok		No attacks within bounds.
		MTC,C6	Nisynch	Ok		No attacks within bounds.
	I	MTC,I1	Secret kci	Ok	Verified	No attacks.
		MTC,I2	Secret PI	Ok	Verified	No attacks.
		MTC,I3	Niagree	Ok	Verified	No attacks.
		MTC,I4	Nisynch	Ok	Verified	No attacks.
	M	MTC,M1	Secret nm	Ok	Verified	No attacks.
		MTC,M2	Secret OI	Ok	Verified	No attacks.
		MTC,M3	Niagree	Ok	Verified	No attacks.
		MTC,M4	Nisynch	Ok	Verified	No attacks.
	A	MTC,A1	Niagree	Ok	Verified	No attacks.
		MTC,A2	Nisynch	Ok	Verified	No attacks.
	PG	MTC,PG1	Niagree	Ok	Verified	No attacks.
		MTC,PG2	Nisynch	Ok	Verified	No attacks.

Done.

Figure 4.4: Result of MTC using "Verification Claim" Procedure of Scyther Tool

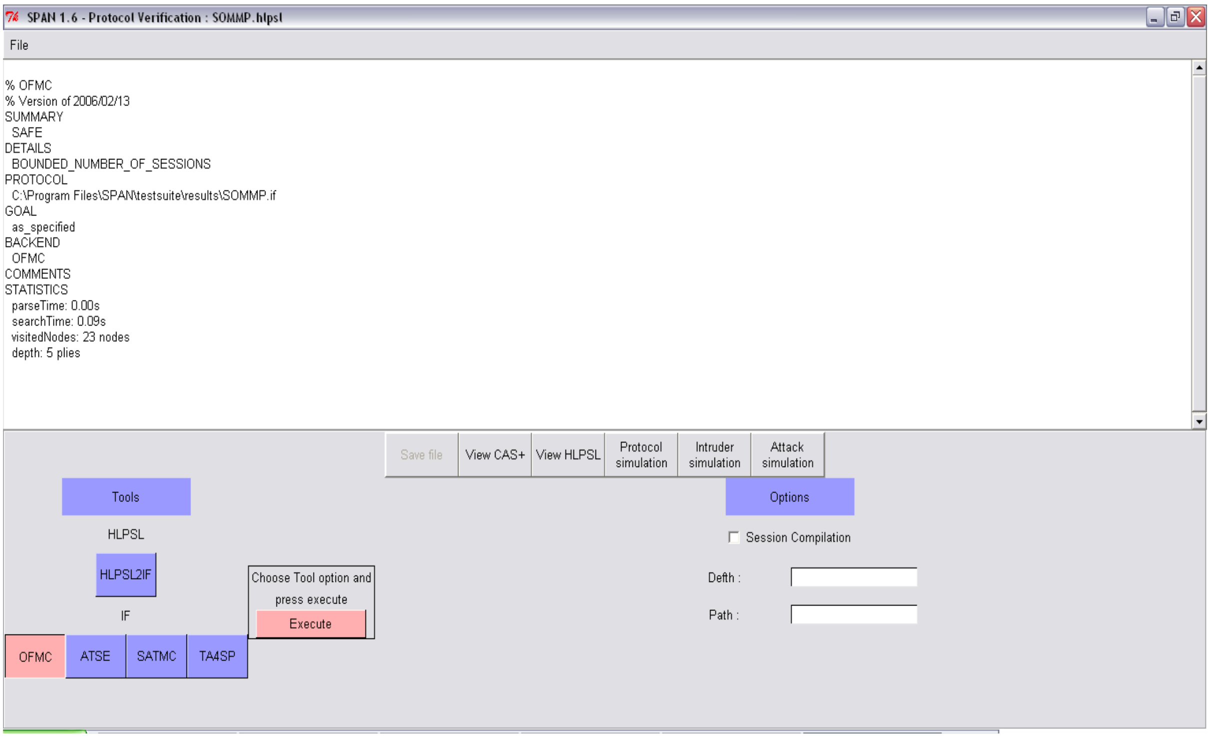


Claim				Status		Comments
MTC	C	MTC,C7	Secret kci	Ok		No attacks within bounds.
		MTC,C8	Secret nc	Ok		No attacks within bounds.
		MTC,C9	Secret ni	Ok		No attacks within bounds.
		MTC,C10	Secret nm	Ok		No attacks within bounds.
		MTC,C11	Niagree	Ok		No attacks within bounds.
		MTC,C12	Nisynch	Ok		No attacks within bounds.
	I	MTC,I5	Secret kci	Ok	Verified	No attacks.
		MTC,I6	Secret ni	Ok	Verified	No attacks.
		MTC,I7	Secret nm	Ok		No attacks within bounds.
		MTC,I8	Secret nc	Ok	Verified	No attacks.
		MTC,I9	Niagree	Ok	Verified	No attacks.
		MTC,I10	Nisynch	Ok	Verified	No attacks.
	M	MTC,M5	Secret kci	Ok	Verified	No attacks.
		MTC,M6	Secret nm	Ok	Verified	No attacks.
		MTC,M7	Secret nb	Ok		No attacks within bounds.
		MTC,M8	Secret nc	Ok	Verified	No attacks.
		MTC,M9	Niagree	Ok	Verified	No attacks.
		MTC,M10	Nisynch	Ok	Verified	No attacks.

Done.

Figure 4.5: Result of MTC using "Automatic Claim" Procedure of Scyther Tool

Formal Verification Results of Chapter 5



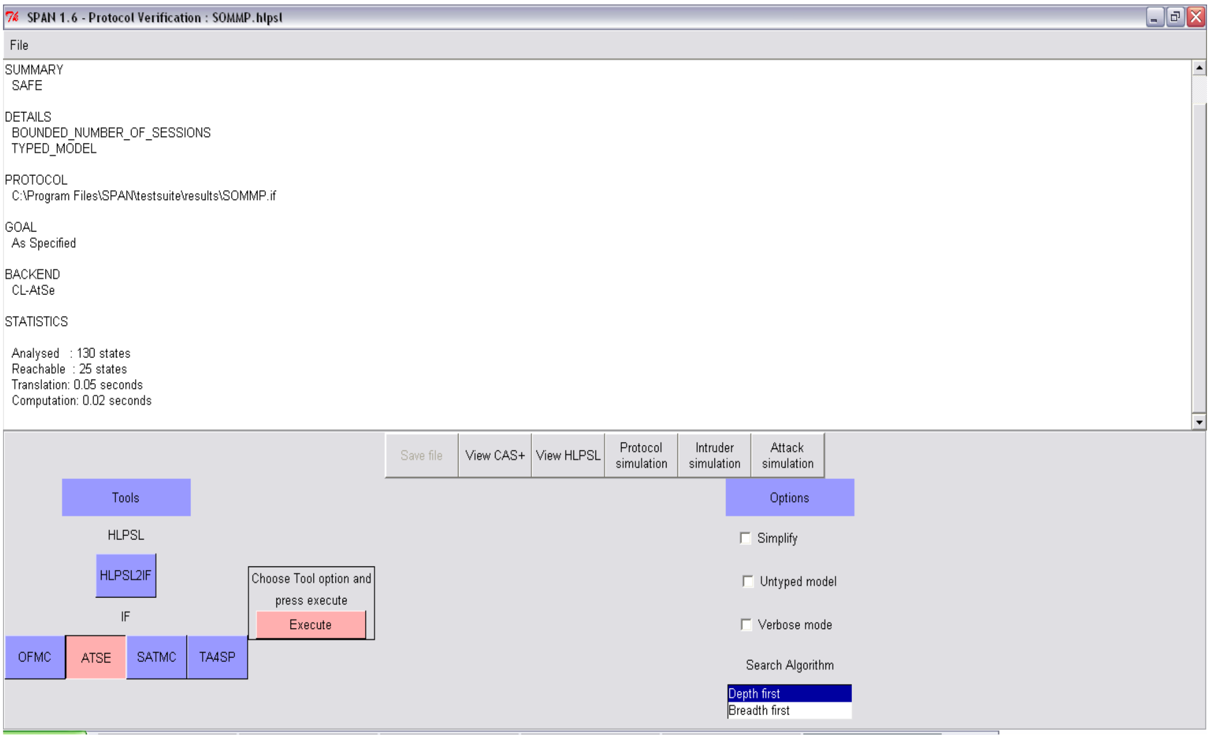
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SOMMP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.09s
  visitedNodes: 23 nodes
  depth: 5 plies
```

□ Session Compilation

Defth :

Path :

Figure 5.3: Result of SOMMP using 'OFMC' backend in AVISPA Tool



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SOMMP.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 130 states
  Reachable  : 25 states
  Translation: 0.05 seconds
  Computation: 0.02 seconds
```

□ Simplify

□ Untyped model

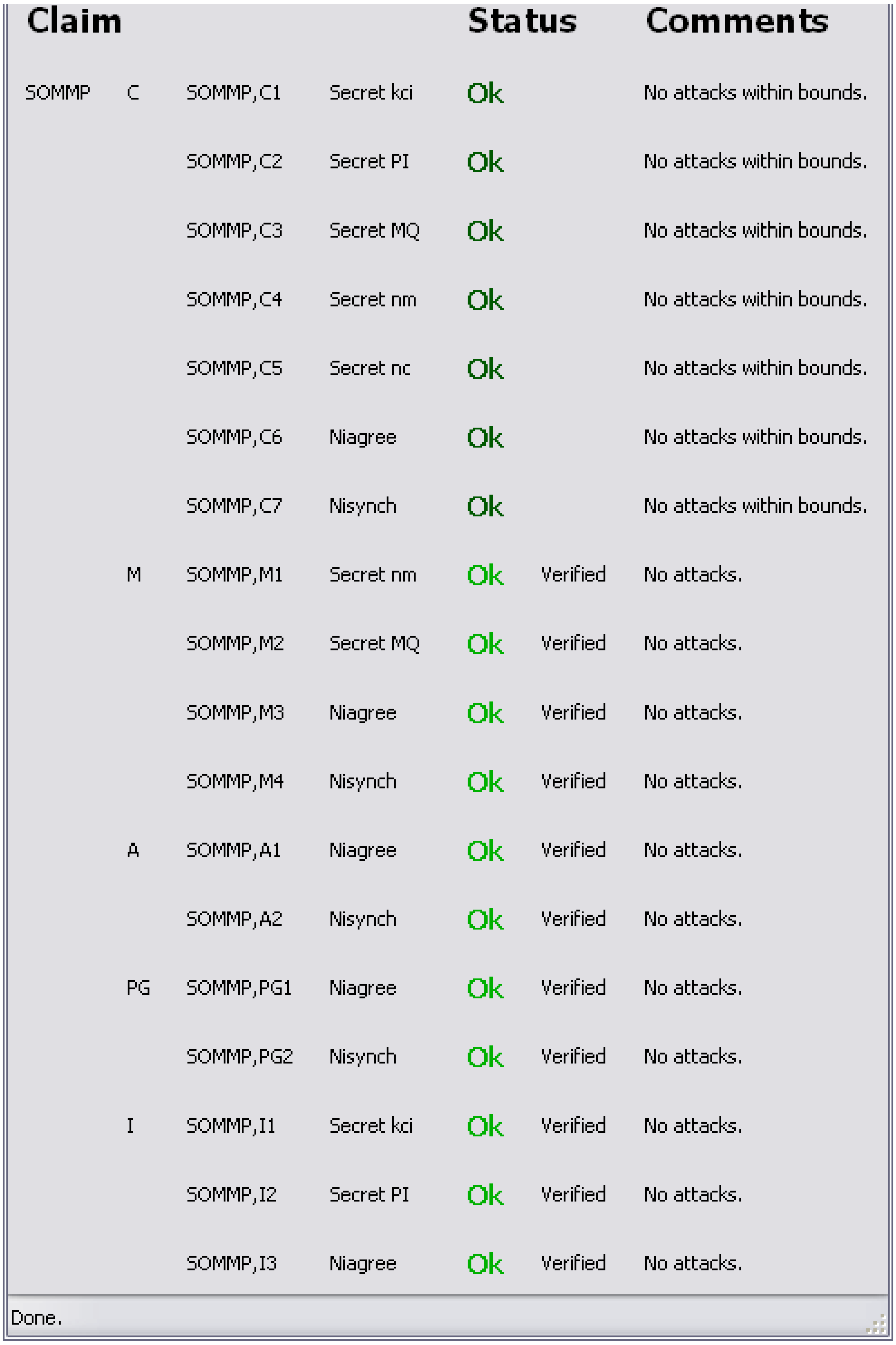
□ Verbose mode

Search Algorithm

Depth first
Breadth first

Figure 5.4: Result of SOMMP using 'ATSE' backend in AVISPA Tool

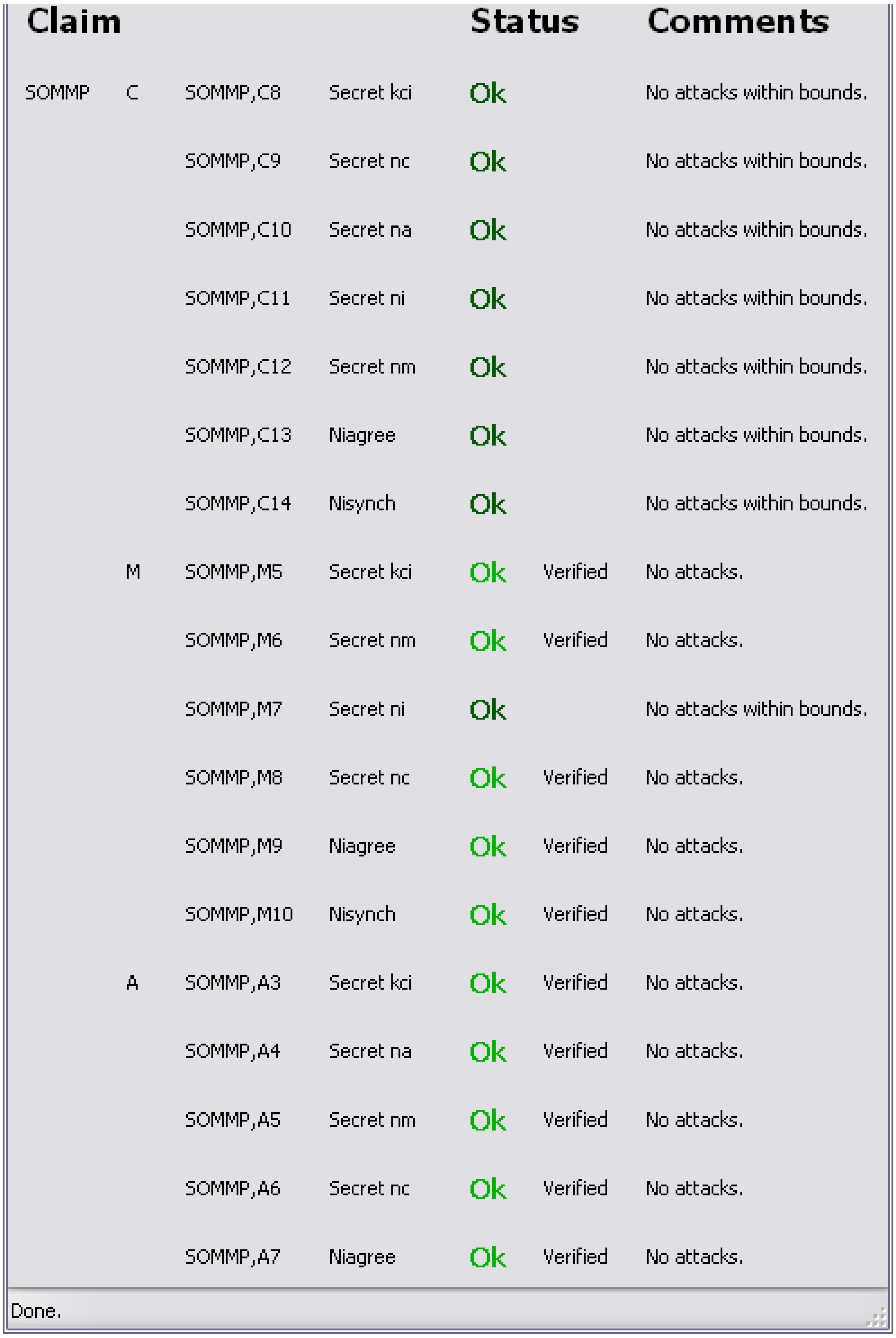
```
const pk: Function;
secret sk: Function;
inversekeys(pk,sk);
usertype Timestamp;
usertype transCertC,PO,PI,TID,Amt,MQ,Tc,Tm;
usertype success;
usertype Sessionkey;
protocol SOMMP(C,M,A,PG,I)
{
   role C
{
      const nc: Nonce;
      var nm,ni,na: Nonce;
      const kci: SessionKey;
      send_1(C,M,{{nc,PO,Tc}sk(C)}pk(M));
      read_2(M,C,{{TID,MQ,Tc,Amt,nc,nm,Tm}sk(M)}pk(C));
      send_3(C,M,{{transCertC,MQ,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(C)}pk(M));
      read_10(M,C,{{success,Amt,TID}sk(M)}pk(C));
      claim_C1(C,Secret,kci);
      claim_C2(C,Secret,PI);
       claim_C3(C,Secret,MQ);
      claim_C4(C,Secret,nm);
       claim_C5(C,Secret,nc);
      claim_C6 (C,Niagree);
      claim_C7(C,Nisynch);
  }
role M
   {
     const  nm: Nonce;
     var  nc,ni: Nonce;
      const kci: SessionKey;
      read_1 (C,M,{{nc,PO,Tc}sk(C)}pk(M));
      send_2(M,C,{{TID,MQ,Tc,Amt,nc,nm,Tm}sk(M)}pk(C));
      read_3(C,M,{{transCertC,MQ,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(C)}pk(M));
      send_4(M,A,{{transCertC,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(M)}pk(A));
      read_9(A,M,{{success,Amt,TID}sk(A)}pk(M));
      send_10(M,C,{{success,Amt,TID}sk(M)}pk(C));
      claim_M1(M,Secret,nm);
       claim_M2(M,Secret,MQ);
       claim_M3(M,Niagree);
       claim_M4(M,Nisynch);
     }
role A
    {
      const na: Nonce;
      var nc,nm: Nonce;
      const kci: SessionKey;
      read_4(M,A,{{transCertC,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(M)}pk(A));
      send_5(A,PG,{{transCertC, Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(A)}pk(PG));
      read_8(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
      send_9(A,M,{{success,Amt,TID}sk(A)}pk(M));
      claim_A1(A,Niagree);
     claim_A2(A,Nisynch);
     }
  role PG
    {
      const npg: Nonce;
      var nc,nm: Nonce;
      const kci: SessionKey;
      read_5(A,PG,{{transCertC,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(A)}pk(PG));
      send_6(PG,I,{{transCertC,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(PG)}pk(I));
      read_7(I,PG,{{success,Amt,TID}sk(I)}pk(PG));
      send_8(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
      claim_PG1(PG,Niagree);
     claim_PG2(PG,Nisynch);
     }
  role I
    {
      const ni: Nonce;
      var nc,nm: Nonce;
      const kci: SessionKey;
      read_6(PG,I,{{transCertC,Tc,Amt,nc,nm,Tm,{PI}kci,TID}sk(PG)}pk(I));
      send_7(I,PG,{{success,Amt,TID}sk(I)}pk(PG));
      claim_I1 (I,Secret,kci);
       claim_I2(I,Secret,PI);
      claim_I3 (I,Niagree);
     claim_I4 (I,Nisynch);
     }
  }
const C,M,A,PG,I,E: Agent;
untrusted E;
const ne: Nonce;
const te: Timestamp;
compromised sk(E);
```



Claim				Status		Comments
SOMMP	C	SOMMP,C1	Secret kci	Ok		No attacks within bounds.
		SOMMP,C2	Secret PI	Ok		No attacks within bounds.
		SOMMP,C3	Secret MQ	Ok		No attacks within bounds.
		SOMMP,C4	Secret nm	Ok		No attacks within bounds.
		SOMMP,C5	Secret nc	Ok		No attacks within bounds.
		SOMMP,C6	Niagree	Ok		No attacks within bounds.
		SOMMP,C7	Nisynch	Ok		No attacks within bounds.
	M	SOMMP,M1	Secret nm	Ok	Verified	No attacks.
		SOMMP,M2	Secret MQ	Ok	Verified	No attacks.
		SOMMP,M3	Niagree	Ok	Verified	No attacks.
		SOMMP,M4	Nisynch	Ok	Verified	No attacks.
	A	SOMMP,A1	Niagree	Ok	Verified	No attacks.
		SOMMP,A2	Nisynch	Ok	Verified	No attacks.
	PG	SOMMP,PG1	Niagree	Ok	Verified	No attacks.
		SOMMP,PG2	Nisynch	Ok	Verified	No attacks.
	I	SOMMP,I1	Secret kci	Ok	Verified	No attacks.
		SOMMP,I2	Secret PI	Ok	Verified	No attacks.
		SOMMP,I3	Niagree	Ok	Verified	No attacks.

Done.

Figure 5.5: Result of SOMMP using "Verification Claim" Procedure of Scyther Tool

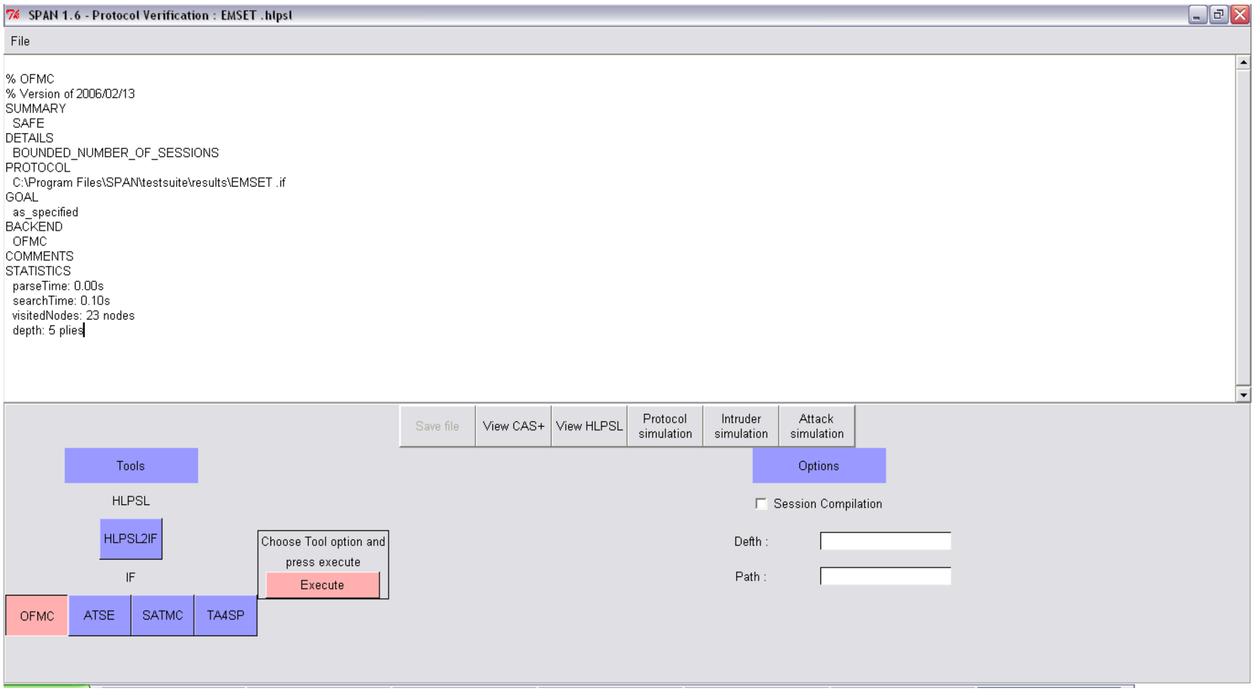


Claim				Status		Comments
SOMMP	C	SOMMP,C8	Secret kci	Ok		No attacks within bounds.
		SOMMP,C9	Secret nc	Ok		No attacks within bounds.
		SOMMP,C10	Secret na	Ok		No attacks within bounds.
		SOMMP,C11	Secret ni	Ok		No attacks within bounds.
		SOMMP,C12	Secret nm	Ok		No attacks within bounds.
		SOMMP,C13	Niagree	Ok		No attacks within bounds.
		SOMMP,C14	Nisynch	Ok		No attacks within bounds.
	M	SOMMP,M5	Secret kci	Ok	Verified	No attacks.
		SOMMP,M6	Secret nm	Ok	Verified	No attacks.
		SOMMP,M7	Secret ni	Ok		No attacks within bounds.
		SOMMP,M8	Secret nc	Ok	Verified	No attacks.
		SOMMP,M9	Niagree	Ok	Verified	No attacks.
		SOMMP,M10	Nisynch	Ok	Verified	No attacks.
	A	SOMMP,A3	Secret kci	Ok	Verified	No attacks.
		SOMMP,A4	Secret na	Ok	Verified	No attacks.
		SOMMP,A5	Secret nm	Ok	Verified	No attacks.
		SOMMP,A6	Secret nc	Ok	Verified	No attacks.
		SOMMP,A7	Niagree	Ok	Verified	No attacks.

Done.

Figure 5.6: Result of SOMMP using "Automatic Claim" Procedure of Scyther Tool

Formal Verification Results of Chapter 6



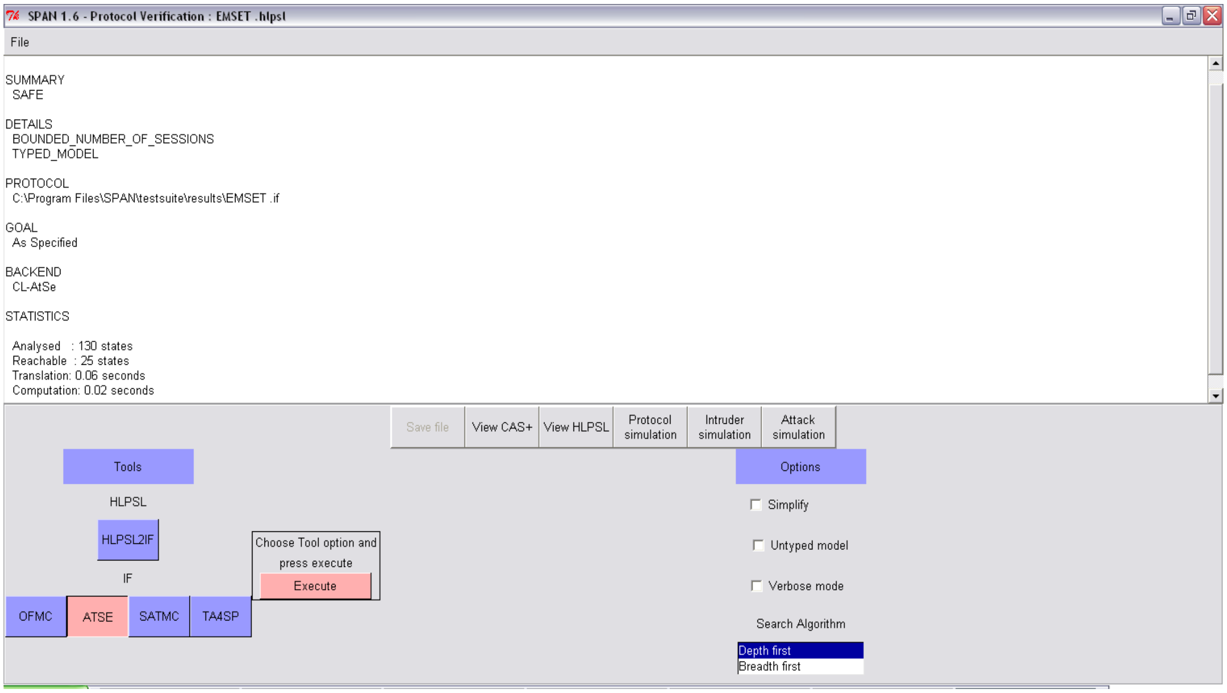
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\EMSET .if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 23 nodes
  depth: 5 plies
```

□ Session Compilation

Defth :

Path :

Figure 6.3: Result of EMSET using OFMC backend of AVISPA Tool



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\Program Files\SPAN\testsuite\results\EMSET .if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 130 states
  Reachable  : 25 states
  Translation: 0.06 seconds
  Computation: 0.02 seconds
```

□ Simplify

□ Untyped model

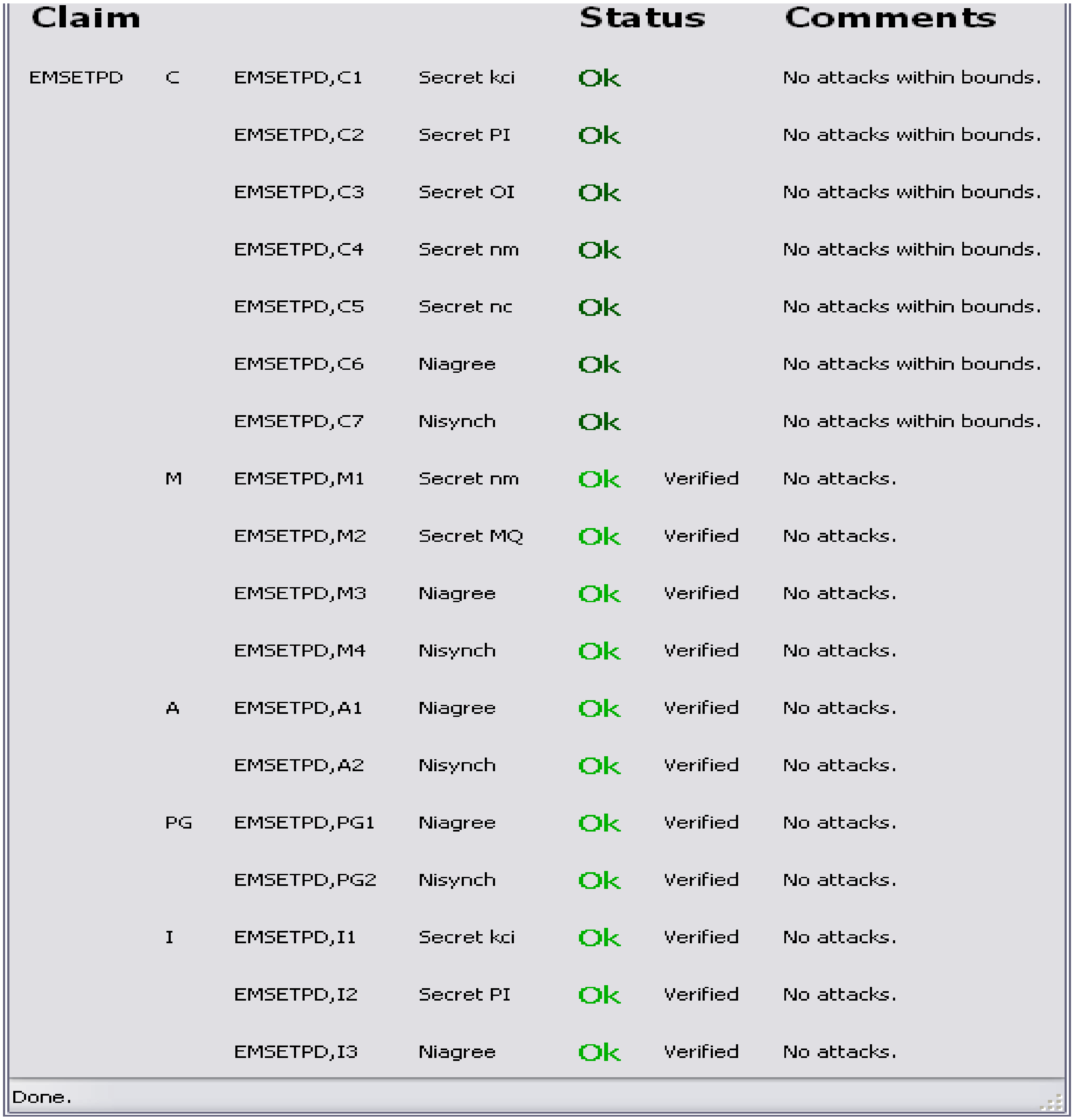
□ Verbose mode

Search Algorithm

Depth first

Breadth first

Figure 6.4: Result of EMSET using ATSE backend of AVISPA Tool

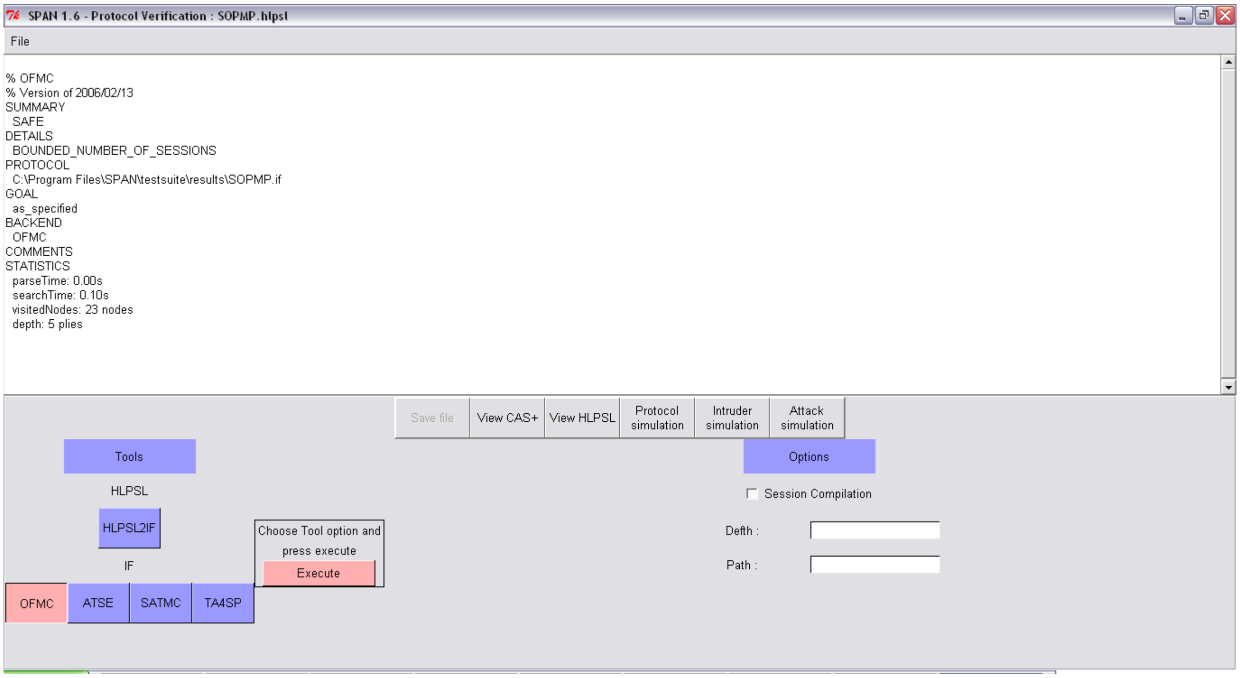


Claim				Status		Comments
EMSETPD	C	EMSETPD,C1	Secret kci	Ok		No attacks within bounds.
		EMSETPD,C2	Secret PI	Ok		No attacks within bounds.
		EMSETPD,C3	Secret OI	Ok		No attacks within bounds.
		EMSETPD,C4	Secret nm	Ok		No attacks within bounds.
		EMSETPD,C5	Secret nc	Ok		No attacks within bounds.
		EMSETPD,C6	Niagree	Ok		No attacks within bounds.
		EMSETPD,C7	Nisynch	Ok		No attacks within bounds.
	M	EMSETPD,M1	Secret nm	Ok	Verified	No attacks.
		EMSETPD,M2	Secret MQ	Ok	Verified	No attacks.
		EMSETPD,M3	Niagree	Ok	Verified	No attacks.
		EMSETPD,M4	Nisynch	Ok	Verified	No attacks.
	A	EMSETPD,A1	Niagree	Ok	Verified	No attacks.
		EMSETPD,A2	Nisynch	Ok	Verified	No attacks.
	PG	EMSETPD,PG1	Niagree	Ok	Verified	No attacks.
		EMSETPD,PG2	Nisynch	Ok	Verified	No attacks.
	I	EMSETPD,I1	Secret kci	Ok	Verified	No attacks.
		EMSETPD,I2	Secret PI	Ok	Verified	No attacks.
		EMSETPD,I3	Niagree	Ok	Verified	No attacks.

Done.

Figure 6.5: Result of Payment and Deposit Phase in EMSET Protocol using "Verification Claim" Procedure of Scyther Tool

Formal Verification Results of Chapter 7



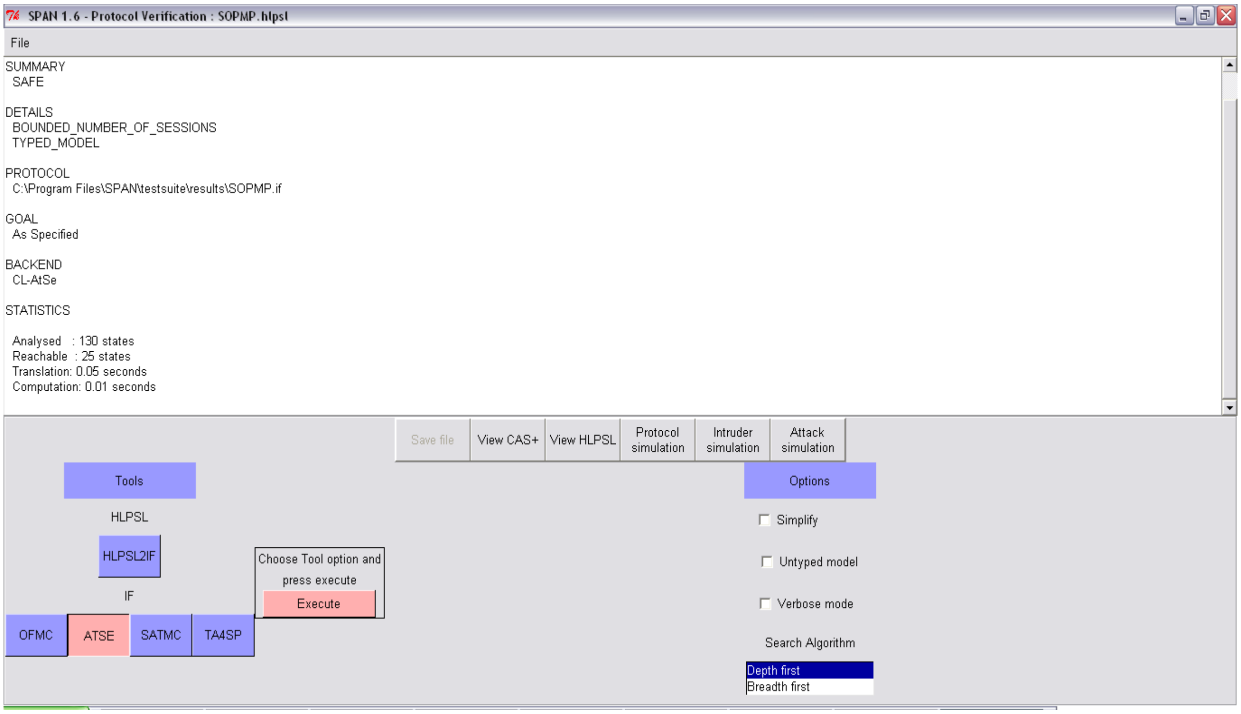
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SOPMP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 23 nodes
  depth: 5 plies
```

□ Session Compilation

Defth :

Path :

Figure 7.2: Result Using OFMC backend using AVISPA Tool



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\Program Files\SPAN\testsuite\results\SOPMP.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 130 states
  Reachable  : 25 states
  Translation: 0.05 seconds
  Computation: 0.01 seconds
```

□ Simplify

□ Untyped model

□ Verbose mode

Search Algorithm

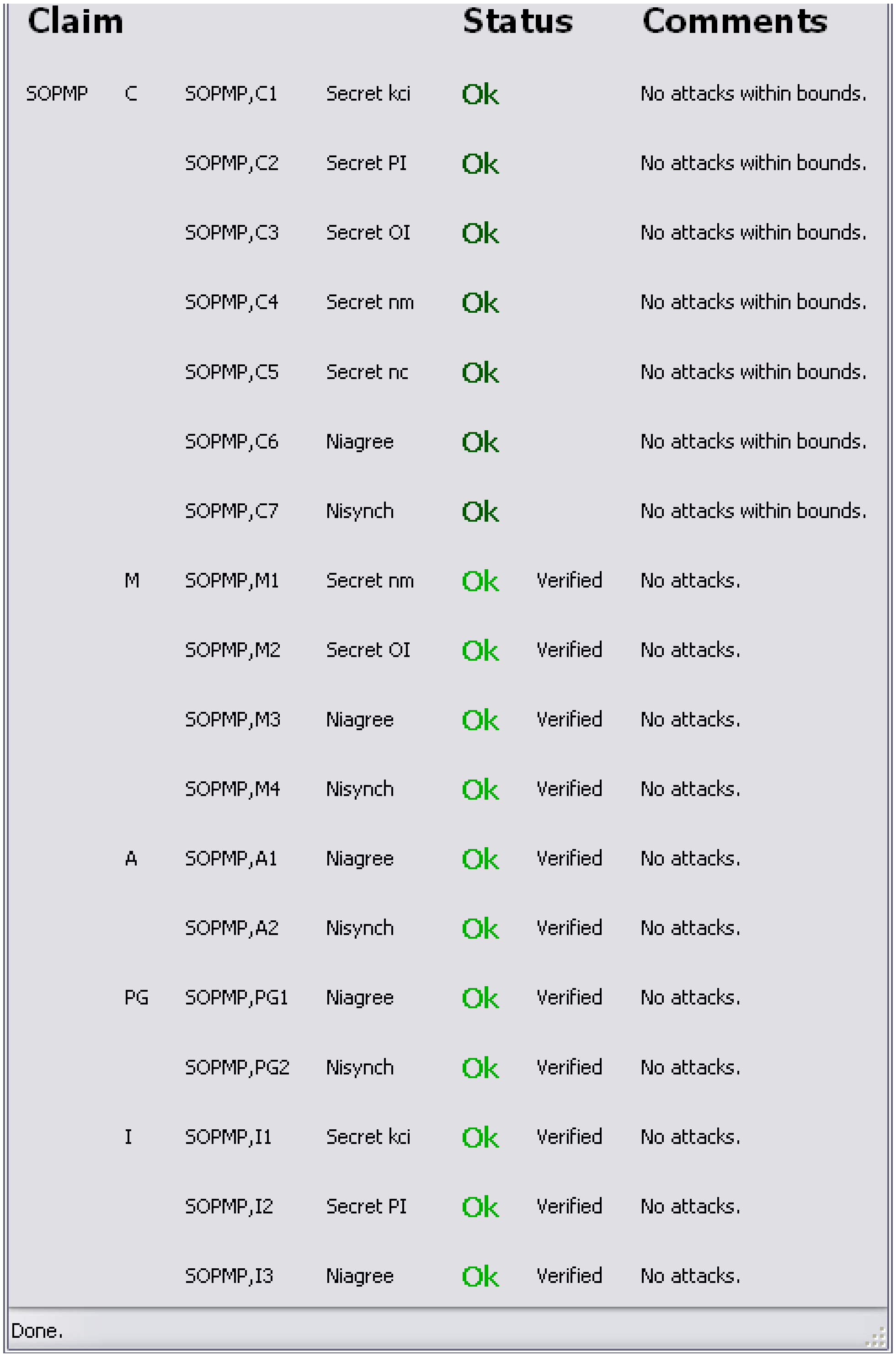
Depth first

Breadth first

Figure 7.3: Result Using CL-Atse backend using AVISPA Tool

```
const pk: Function;
secret sk: Function;
inversekeys(pk,sk);
usertype Timestamp;
usertype DIC,OI,PI,TID,Amt,Tc,Tm;
usertype success;
usertype Sessionkey;
protocol SOPMP(C,M,A,PG,I)
{
role C
{
const nc: Nonce;
        var nm,ni,na: Nonce;
        const kci: SessionKey;
        send_1(C,M,{{nc,OI,Tc}sk(C)}pk(M));
        read_2(M,C,{{DIC,Tc,Amt,nc,nm,Tm}sk(M)}pk(C));
        send_3(C,M,{{DIC,OI,Tc,Amt,nc,nm,Tm,{PI}kci}sk(C)}pk(M));
        read_10(M,C,{{success,Amt,TID}sk(M)}pk(C));
        claim_C1(C,Secret,kci);
        claim_C2(C,Secret,PI);
         claim_C3(C,Secret,OI);
        claim_C4(C,Secret,nm);
         claim_C5(C,Secret,nc);
        claim_C6 (C,Niagree);
        claim_C7(C,Nisynch);
   }
 role M
    {
        const  nm: Nonce;
       var  nc,ni: Nonce;
        const kci: SessionKey;
        read_1(C,M,{{nc,OI,Tc}sk(C)}pk(M));
        send_2(M,C,{{DIC,Tc,Amt,nc,nm,Tm}sk(M)}pk(C));
        read_3(C,M,{{DIC,OI,Tc,Amt,nc,nm,Tm,{PI}kci}sk(C)}pk(M));
        send_4(M,A,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(M)}pk(A));
        read_9(A,M,{{success,Amt,TID}sk(A)}pk(M));
        send_10(M,C,{{success,Amt,TID}sk(M)}pk(C));
        claim_M1(M,Secret,nm);
         claim_M2(M,Secret,OI);
         claim_M3(M,Niagree);
         claim_M4(M,Nisynch);
      }
role A
{
        const na: Nonce;
```

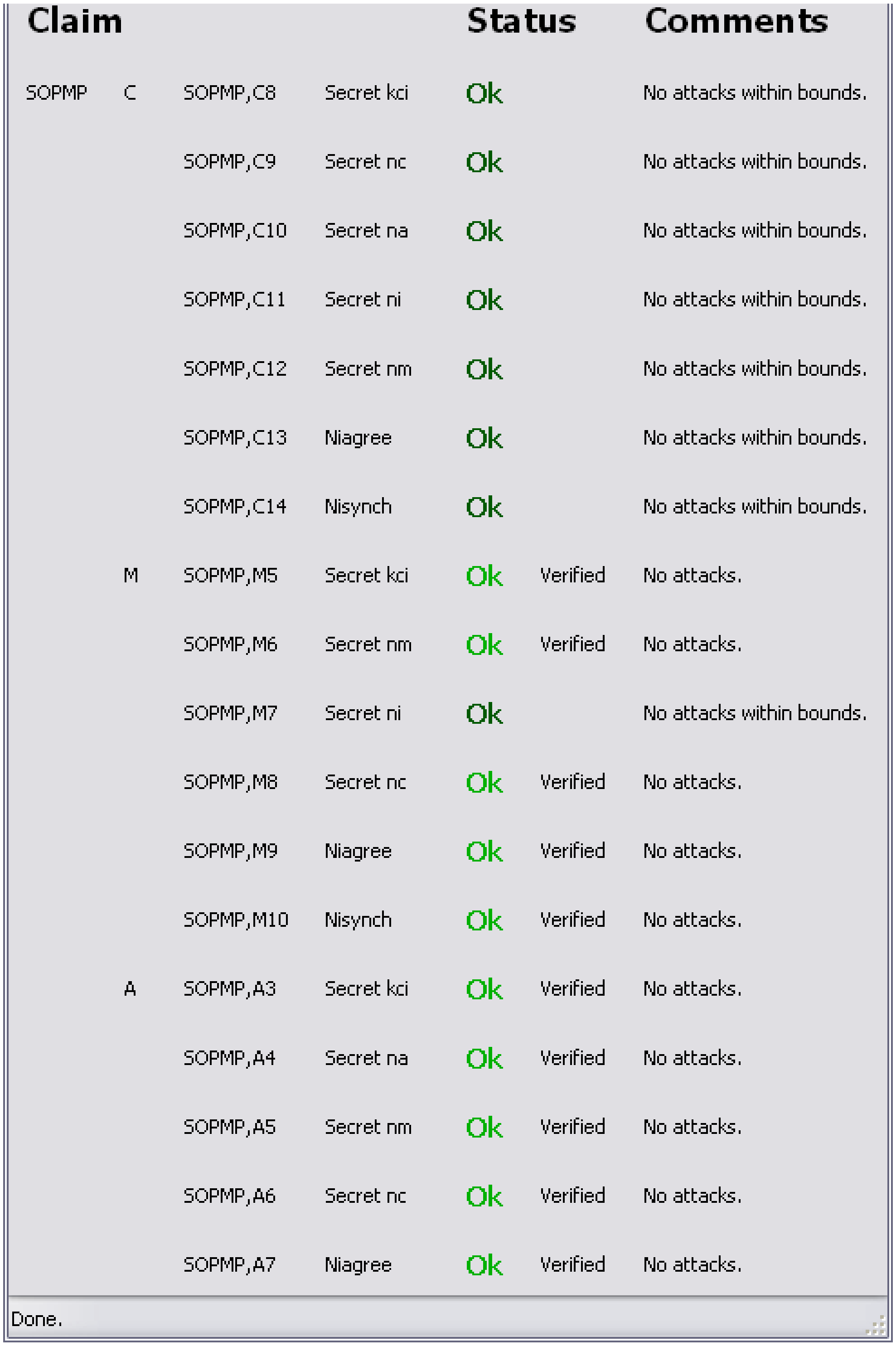
```
          var nc,nm: Nonce;
          const kci: SessionKey;
         read_4 (M,A,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(M)}pk(A));
         send_5(A,PG,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(A)}pk(PG));
          read_8(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
          send_9(A,M,{{success,Amt,TID}sk(A)}pk(M));
          claim_A1(A,Niagree);
         claim_A2(A,Nisynch);
        }
     role PG
   {
          const npg: Nonce;
          var nc,nm: Nonce;
          const kci: SessionKey;
          read_5(A,PG,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(A)}pk(PG));
          send_6(PG,I,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(PG)}pk(I));
          read_7(I,PG,{{success,Amt,TID}sk(I)}pk(PG));
          send_8(PG,A,{{success,Amt,TID}sk(PG)}pk(A));
          claim_PG1(PG,Niagree);
         claim_PG2(PG,Nisynch);
        }
    role I
   {
          const ni: Nonce;
          var nc,nm: Nonce;
          const kci: SessionKey;
          read_6(PG,I,{{DIC,Tc,Amt,nc,nm,Tm,{PI}kci}sk(PG)}pk(I));
          send_7(I,PG,{{success,Amt,TID}sk(I)}pk(PG));
          claim_I1 (I,Secret,kci);
           claim_I2(I,Secret,PI);
          claim_I3 (I,Niagree);
         claim_I4 (I,Nisynch);
        }
}

   const C,M,A,PG,I,E: Agent;
   untrusted E;
   const ne: Nonce;
   const te: Timestamp;
   compromised sk(E);
```



Claim				Status		Comments
SOPMP	C	SOPMP,C1	Secret kci	Ok		No attacks within bounds.
		SOPMP,C2	Secret PI	Ok		No attacks within bounds.
		SOPMP,C3	Secret OI	Ok		No attacks within bounds.
		SOPMP,C4	Secret nm	Ok		No attacks within bounds.
		SOPMP,C5	Secret nc	Ok		No attacks within bounds.
		SOPMP,C6	Niagree	Ok		No attacks within bounds.
		SOPMP,C7	Nisynch	Ok		No attacks within bounds.
	M	SOPMP,M1	Secret nm	Ok	Verified	No attacks.
		SOPMP,M2	Secret OI	Ok	Verified	No attacks.
		SOPMP,M3	Niagree	Ok	Verified	No attacks.
		SOPMP,M4	Nisynch	Ok	Verified	No attacks.
	A	SOPMP,A1	Niagree	Ok	Verified	No attacks.
		SOPMP,A2	Nisynch	Ok	Verified	No attacks.
	PG	SOPMP,PG1	Niagree	Ok	Verified	No attacks.
		SOPMP,PG2	Nisynch	Ok	Verified	No attacks.
	I	SOPMP,I1	Secret kci	Ok	Verified	No attacks.
		SOPMP,I2	Secret PI	Ok	Verified	No attacks.
		SOPMP,I3	Niagree	Ok	Verified	No attacks.

Done.

Figure 7.4: Result Using "Verification Claim" procedure in Scyther Tool



Claim				Status		Comments
SOPMP	C	SOPMP,C8	Secret kci	Ok		No attacks within bounds.
		SOPMP,C9	Secret nc	Ok		No attacks within bounds.
		SOPMP,C10	Secret na	Ok		No attacks within bounds.
		SOPMP,C11	Secret ni	Ok		No attacks within bounds.
		SOPMP,C12	Secret nm	Ok		No attacks within bounds.
		SOPMP,C13	Niagree	Ok		No attacks within bounds.
		SOPMP,C14	Nisynch	Ok		No attacks within bounds.
	M	SOPMP,M5	Secret kci	Ok	Verified	No attacks.
		SOPMP,M6	Secret nm	Ok	Verified	No attacks.
		SOPMP,M7	Secret ni	Ok		No attacks within bounds.
		SOPMP,M8	Secret nc	Ok	Verified	No attacks.
		SOPMP,M9	Niagree	Ok	Verified	No attacks.
		SOPMP,M10	Nisynch	Ok	Verified	No attacks.
	A	SOPMP,A3	Secret kci	Ok	Verified	No attacks.
		SOPMP,A4	Secret na	Ok	Verified	No attacks.
		SOPMP,A5	Secret nm	Ok	Verified	No attacks.
		SOPMP,A6	Secret nc	Ok	Verified	No attacks.
		SOPMP,A7	Niagree	Ok	Verified	No attacks.

Done.

Figure 7.5: Result Using “Automatic Claim” procedure in Scyther Tool

Bibliography

1) Abadi, M., Burrows, M., Kaufman, C. and Lampson, B. (1993). Authentication and delegation with smart-cards. *Science of Computer Programming, Vol. 21*, 93–113.

2) Armando et al., (2005). The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. *Paper presented at 17th International Conference on Computer Aided Verification* (*CAV*) (LNCS 3576), 281-285.

3) Aydos, M., Yanik, T. and Koc, C.K. (2001). High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor. *IEE Proceedings Communications 148 (5),* 273–279.

4) Batina, L., S.B. O rs, B. Preneel, J. Vandewalle. (2003). Hardware architectures for public key cryptography. *Integration the VLSI Journal 34 (1–2),* 1–64.

5) Blanchet, B. (2009). An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. *Paper presented at* 14th *IEEE Computer Security Foundations Workshop CSFW*), Cape Breton, IEEE Computer Society, 82-96.

6) Block, R. (2011). *First Release of Google Wallet with MasterCard Now Available for Sprint Nexus S 4G customers*. Retrieved from http://pymnts.com/news/businesswire-feed/2011/september/19/First-Release-of-Google-Wallet-with-MasterCard-Now-Available-for-Sprint-Nexus-S-4G-customers.

7) Burrows, M., Abadi, M. and Needham, R. (1989). A Logic of authentication. In *Proceedings of the Royal Society of London A*, (Vol. 426, 233–271, A preliminary version appeared as Digital Equipment Corporation Systems Research Center report No.39)

8) Buyya, Rajkumar; Ma, Tianchi; Safavi-Naini, Reihaneh; Steketee, Chris and Susilo, Willy (2006). Formal analysis of card-based payment systems in mobile devices. *Paper presented at Fourth Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the Fourth Australasian Information Security Workshop (Network Security) (AISW 2006)* (Hobart, Tasmania, Australia), 213-220.

9) Chang, C-W., Pan, H. and Jia, H-Y. (2006). A secure short message communication protocol. *International Journal of Automation and Computing 5 (2)*, 202–207

10) Chanson, S. and Cheung, T. (2001). Design and implementation of a PKI-based end-to-end secure infrastructure for mobile e-commerce. *World Wide Web 4(4)*, 235 - 253

11) Chen, W., Hancke, G., Mayes, K., Lien, Y. & Chiu, J. (2010). Using 3G network components to enable NFC mobile transactions and authentication. *Paper presented at IEEE International Conference on Progress in Informatics and Computing (PIC)*, (IEEE COMPUTER SOCIETY), 441 -448.

12) Chun Chen, Daojing He, Sammy Chan, Jiajun Bu, Yi Gao, Rong Fan. (2011). Lightweight and provably secure user authentication with anonymity for the global mobility network. *International Journal of Communication Systems 24(3)*, 347-362.

13) Chung-Ming Ou, C.R.Ou (2010). SETNR/A: an agent-based secure payment protocol for mobile commerce. *International Journal of Intelligent Information and Database Systems 4(3)*, 212 - 226

14) Cremers, C. and Lafourcade, P. (2009). Comparing State Spaces in Automatic Security Protocol Verification. *Paper presented at 7th International Work-shop on Automated Verification of Critical Systems (AVoCS'07),* 49 - 63.

15) C. J. F. Cremers, (2006). *Scyther-Semantics and Verification of Security Protocols*. (Ph.D. Thesis). Eindhoven University of Technology, Eindhoven, The Netherlands.

16) D.R. Stinson (2006). *Cryptography-Theory and Practice*. 3rd edition, Chapman & Hall/CRC.

17) Emir Husni, Kuspriyanto, Noor Basjaruddin (2012). Mobile Payment Protocol for Tag-to- Tag Near Field Communication (NFC). *International Journal of Interactive Mobile Technologies (IJIM) 6(4)*, 34-38.

18) Ernst-Joachim Steffens, Axel Nennker, Dr. Zhiyun Ren, Ming Yin (2009). The SIM based Mobile Wallet. *Paper presented at 13th International Conference on Intelligence in Next Generation Networks*. 1-6.

19) Gianluigi Me, Alex Schuster (2005). A mobile local payment system Bluetooth based. *Paper presented at International Symposium on Wireless Communications (ISWSN'05).* KFUPM. 1-5.

20) Gianluigi Me, Maurizio A. Strangio (2005). EC-PAY: An Efficient and Secure ECC-basedWireless Local Payment Scheme. *Paper presented at the 3rd International Conference on Information Technology and Applications, ICITA (2).* 442-447.

21) GlobalPlatform (2009). Proposition for NFC mobile: secure element management and messaging. White paper, April.

22)Google Wallet *Launch Partners*. (2011). Retrieved from http://www.google.com/wallet/current-partners.html.

23) Gu, J., Park, S., Song, O., Lee, J., Nah, J. and Sohn, S. (2003). Mobile PKI: a PKI-based authentication framework for the next generation mobile communications. *Paper presented at 8th Australasian Conference on Information Security and Privacy, (Wollongong, Australia),* 180-191.

24) Hassinen, M. (2005). SafeSMS - end-to-end encryption for SMS messages. *Paper presented at 8th International Conference on Telecommunications (Vol. 2),* 359–365.

25) Hassinen, M. and Hypponen, K. (2005). Strong mobile authentication. *Paper presented at the Second International Symposium on Wireless Communication Systems* (5–7 September), 96–100.

26) He, R., Qin, Z. and Qin, X. (2008). A secured Mobile access scheme for SMS message. *Information Technology Journal* 7*(2)*, 261–268.

27) Horn-Twu Liaw, Lin, J-F. and Wu, W-C. (2007). A new electronic traveler's check scheme based on one-way hash function. *Electronic Commerce Research and Applications Journal 6(4)*, 499-508.

28) Hsien, J-E., Hsueh, C-C. and Chan, C-Y. (2001). An electronic check system. *Paper presented at the Conference on Theory and Practice for Electronic Commerce (Tai*wan), 164–169.

29) Huda Ubaya (2012). Design of Prototype Payment Application System with near Field Communication (NFC) Technology based on Android. *Computer Engineering and Applications 1(1),* 1-12.

30) Hwang, M.S., Lin, I.C. and Li, L.H. (2001). A simple micro-payment scheme. *The Journal of Systems and Software 55(3)*, 221–229.

31) Hwang, Ren-Junn., Chih-Hua Lai, Feng-Fu Su. (2005). An efficient signcryption scheme with forward secrecy based on elliptic curve. *Applied Mathematics and Computation 167*, 870–881.

32) Juul, N.C. and Jorgenson, N.H. (2002). Security issues in mobile commerce using WAP. *Paper presented at the Fifteenth Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy*.1-14.

33) Karnouskos, S. (2004). Mobile payment: a journey through existing procedures and standardization initiatives. *IEEE Communication Surveys 6(4)*, 44–66.

34) Kungpisdan, B. Srinivasan, and P.D. Le. (2003). Lightweight mobile credit-card payment protocol. *Paper presented at the International Conference on Progress in Cryptology (LNCS 2904)*, 295-308.

35) Kungpisdan, S., Srinivasan, B. and Le, P.D. (2004b). A secure account-based mobile payment protocol. *Paper presented at the International Conference on Information Technology: Coding and Computing (ITCC '04)*. 35 – 39.

36) Lee, C-s., Kim, Hq. and Lee, J. (2006). Design of user authentication system based on WPKI. *Paper presented at the Tenth International Conference on CSCW in Design*. 979-983.

37) Lo, J.L.C. et al. (2008). SMSSec: an end-to-end protocol for secure SMS. *Computers and Security 27 (5–6)*, 154–167.

38) Longyi Li and Lihua Tao (2009). Security Study of Mobile Business Based on WPKI. *Paper presented at the 8th International Conference on Mobile Business (ICMB 2009),* (Guangzhou, China). 301-304.

39) M. Massoth and T. Bingel, (2009). Performance of different mobile payment service concepts compared with a NFC-based solution. *Paper presented at the 4th International Conference on Internet and Web Applications and Services (ICIW '09)*. 205–210.

40) Mastercard and Visa. SET Protocol Specifications. Retrieved from http://www.setco.org/set_specifications.html

41) Mateja Jovanovic and Mario Muñoz Organero (2011). Analysis of the Latest Trends in Mobile Commerce using NFC Technology. *Journal of Selected Areas in Telecommunications (JSAT) May Edition (2011),* 1-12.

42) Menezes, A.J., van Oorschot, P.C. and Vanstone, S.A. (1996). *Handbook of Applied Cryptography*. CRC Press.

43) MobeyForum (2008). Best practice for mobile financial services – enrolment business model analysis.

44) Muhammad, S., Furqan, Z. and Guha, R.K. (2006). Understanding the intruder through attacks on cryptographic protocols. *Paper presented at the 44th ACM Southeast Conference* (*ACMSE2006*).667–672.

45) Muhammad, S., Furqan, Z. and Guha, R.K. (2007). A logic-based verification framework for authentication protocols. *International Journal of Internet Technology and Secured Transactions 1(1/2)*, 49–80.

46) Ngo, Huy Hoang. Osama Dandash, Phu Dung Le, Bala Srinivasan and Campbell Wilson. (2011). Formal Verification of a Secure Mobile Banking Protocol. *Advances in Networks and Communications in Computer and Information Science 132(2)*, 410-421

47) Senn, J.A. (2000). The emergence of m-commerce. *IEEE Computer 33(12)*, 148–150.

48) NTT DoCoMo (2012). *Osaifu-Keitai (mobile phones with wallet functions) service*. Retrieved from http://www.nttdocomo.com/services/osaifu/index.html.

49) OMA (2001) *Wireless Application Protocol Architecture Specification, WAP-210-WAPARch*, July.

50) OMA (2001) *Wireless Application Protocol WAP 2.0 Technical White Paper*, April.
OMA, WAP Certificate and CRL, WAP-211-X.509, March 2000.

51) Romao A, M, da Silva (2001). Secure mobile agent digital signatures with proxy certificates. E-Commerce Agents, *LANI 2033*, 206-220.

52) Romão A and Miguel Mira da Silva (1998). An Agent-Based Secure Internet Payment System for Mobile Computing. Trends in Distributed Systems for Electronic Commerce. 80-93.

53) Shiang-Feng Tzenga, Min-Shiang Hwang (2004). Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem. *Computer Standards & Interfaces 26 (2)*. 61–71.

54) Song, D. (1999). Athena: A New Efficient Automatic Checker for Security Protocol Analysis. *Paper presented at the 12th IEEE Computer Security Foundations Workshop (CSFW)*, IEEE Computer Society, 192-202.

55) Supakorn Kungpisdan, Bala Srinivasan, Phu Dung Le (2003). A Practical Protocol for Mobile SET Payment. *Paper presented at the IADIS International E-Society Conference*, (Lisbon, Portugal), 321-328.

56) Supakorn Kungpisdan, Bala Srinivasan, and Phu Dung Le (2003). Lightweight Mobile Credit-Card Payment Protocol. *Paper presented at INDOCRYPT, LNCS 2904*, 295–308.

57) Supakorn Kungpisdan, Bala Srinivasan, and Phu Dung Le (2004). A Secure Account-Based Mobile Payment Protocol. *Paper presented at the International Conference on Information Technology: Coding and Computing (ITCC'04*), 35-39.

58) Syverson.P and Cervesato.I. (2001). The logic of authentication protocols. Foundations of Security Analysis and Design, Bertinoro, LNCS, vol. 2171, 63–136.

59) T. S. Fun, L. Y. Beng, J. Likoh and R. Roslan. (2008). A Lightweight and Private Mobile Payment Protocol by Using Mobile Network Operator. *Paper presented at the International Conference on Computer and Communication Engineering (ICCCE 2008).* 162–166.

60) Téllez, J., & Sierra, J. (2007). Anonymous payment in a client centric model for digital ecosystems. *Paper presented at the 1st IEEE international conference on digital ecosystems and technologies (IEEE-DEST 2007),* 422–427.

61) Téllez, J., Sierra, J., Izquierdo, A., & Carbonell, M. (2006a). Payment in a kiosk centric model with mobile and low computational power devices. *Paper presented at the Computational science and its applications (ICCSA 2006).*798–807.

62) Téllez, J., Sierra, J., Izquierdo, A., & Torres, J. (2006b). Anonymous payment in a Kiosk centric model using digital signature scheme with message recovery and low computational power devices. *Journal of Theoretical and Applied Electronic Commerce Research 1(2),* 1–11.

63) Téllez, J., Sierra, J., Zeadally, S., & Torres, J. (2008). A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. *Computer Communications 31(10)*, 2478–2484.

64) Tseng, Y., J. Jan, H. Chien, (2003). Digital signature with message recovery using selfcertified public keys and its variants, *Applied Mathematics and Computation 136 (2–3)*, 203–214

65) Turowski, K. and Pousttchi, K. (2004). Mobile Commerce: Basics and Techniques. *Mobile,* Vol. 3576 of Lecture Notes in Computer Science, Springer, 281–285.

66) Vedat Coskun, Kerem Ok, and Busra Ozdenizci. (2012). *Near Field Communication: from theory to practice*. John Wiley & Sons Ltd

67) Wang X. F. et al (1999). Secure Agent-Mediated Mobile Payment. *Paper presented at the PRIMA 99,* LNAI 1599, 162-173.

68) Weidar Chen, Yuanhung Lien, Keith Mayes, Gerhard Hancke and Jung-Hui Chiu. (2010). NFC Mobile Transaction and Authentication based on GSM Network. *Paper presented at the Second International Workshop on Near Field Communication (NFC 2010),* 83-89.

69) Xiaolin Pang, Kian–Lee Tan, Yan Wang, and Jian Ren, (2002). A Secure Agent-Mediated Payment Protocol. *Paper presented at the Fourth International Conference on Information and Communications Security*, volume LNCS 2512, 422-433.

70) Y. Chang, C. Chang, H. Huang (2005). Digital signature with message recovery using self-certified public keys without trustworthy system authority. *Applied Mathematics and Computation 161 (1),* 211–227.

71) Yannis Labrou, Jonathan Agre, Lusheng Ji, Jesus Molina, Wei-lun Chen (2004). Wireless Wallet. *Paper presented at the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (MobiQuitous 2004). 32-41.

72) Yong Lee, Jeail Lee, JooSeok Song (2007). Design and implementation of wireless PKI technology suitable for mobile phone in mobile-commerce. *Computer Communications 30(4),* 893-903.

73) Zhao, H., & Muftic, S. (2011). Design and implementation of a mobile transactions client system: Secure UICC mobile wallet. *International Journal for Information Security Research 1(3)*, 113–120.

74) Zuhua Shao (2004). Improvement of digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem. *Computer Standards & Interfaces 27(1)*, 61–69.