# A secure NFC mobile payment protocol based on biometrics with formal verification

Shaik Shakeel Ahamad*

College of Computer and Information Sciences,
Majmaah University,
Al Majmaah, Kingdom of Saudi Arabia
Email: s.ahamad@mu.edu.sa
Email: ahamadss786@gmail.com
*Corresponding author

Ibrahim Al-Shourbaji

Computer Network Department,
Computer Science and Information System College,
Jazan University,
Jazan, Kingdom of Saudi Arabia
Email: i_shurbaji@yahoo.com

Samaher Al-Janabi

Department of Information Networks,
Faculty of Information Technology (IT),
University of Babylon,
Babylon 00964, Iraq
Email: samaher@itnet.uobabylon.edu.iq

**Abstract:** In this paper, we propose a secure NFC mobile payment protocol based on biometrics (SNMPBs) using wireless public key infrastructure (WPKI) and universal integrated circuit card (UICC). Electronic signatures generated in this protocol are considered qualified signatures as they are generated in UICC which is tamper resistant device. A procedure for the personalisation of mobile payment application (on the UICC) (by the issuer/bank) is proposed. Our SNMPB resolves disputes efficiently among stakeholders by collecting evidence using transaction counters, transaction log, forensics mode and cryptographic audit log techniques. SNMPB ensures end-to-end security (i.e., from mobile payments application in UICC to the bank server) thereby achieving confidentiality, authentication, integrity and non-repudiation properties, prevents double spending and over spending. Our proposed SNMPB protocol withstands replay, man in the middle (MITM), impersonation and multi-protocol attacks as SNMPB is formally verified successfully using BAN logic and Scyther tool.

**Keywords:** secure NFC mobile payment protocol based on biometric; SNMPB; wireless public key infrastructure; WPKI; universal integrated circuit card; UICC; BAN logic; Scyther tool; man in the middle; MITM; multi-protocol attacks.

**Reference** to this paper should be made as follows: Ahamad, S.S., Al-Shourbaji, I. and Al-Janabi, S. (2016) 'A secure NFC mobile payment protocol based on biometrics with formal verification', *Int. J. Internet Technology and Secured Transactions*, Vol. 6, No. 2, pp.103–132.

**Biographical notes:** Shaik Shakeel Ahamad is currently working as an Assistant Professor in CCIS, Majmaah University, Kingdom of Saudi Arabia. He was a Professor in the Department of CSE, KL University, Guntur, India. He holds a PhD in Computer Science from the University of Hyderabad, India in the realm of secure mobile payments protocols and formal verification. His research interests include cloud-based mobile commerce, secure mobile healthcare frameworks and protocols, wireless public key infrastructure and digital forensics.

Ibrahim Al-Shourbaji received his BSc in Computer Science from the Al-Zaytoonah University, Amman, Jordan in 2004 and MSc in Information, Network, and Computer Security from the New York Institute of Technology, USA in 2007. He is a Lecturer at the Jazan University in Saudi Arabia. His research interests are security and privacy, data mining and machine intelligence. He has published more than ten scientific papers.

Samaher Al-Janabi received her BSc, MSc and PhD in Computer Science from the Science College, University of Babylon, Iraq. Through her years of study, she specialised in the design, implementation and performance measurement and intelligent analysis of huge/big data, databases. She is a Lecturer in the Departments of Software and Computer Science, respectively. Her research interests span topics concerning intelligent data analysis, knowledge discovery in databases, soft computing techniques, artificial intelligence, data mining, predication techniques, mobile services, and network security. She has published well over 35 scientific papers and authored two books on new trends of KDD and one book on soft computing techniques. She is a one from five women winner the L'Oreal-UNESCO for Women in Science Levant and Egypt Regional Fellowships. She is a reviewer of several local and international journals.

This paper is a revised and expanded version of a paper entitled 'A biometric based secure mobile payment framework' presented at the 4th IEEE International Conference on Computer and Communication Technology, ICCCT 2013, Allahabad, India, 20–22 September 2013.

## 1 Introduction

Mobile phones (MPs) are increasingly being used for accessing a variety of services over the internet; these services include mobile commerce and mobile payments. Mobile commerce and mobile payments require the ability to make payments with the help of a mobile handset anytime, anywhere and for any reason. These services should ensure an end-to-end security which includes message integrity, authentication, confidentiality, and non-repudiation. Many MPs these days come with fingerprint scanners (FSs). Such as Nokia 5800 and N97, Motorola, which has a FS for authentication of transactions. In this paper, we propose a secure NFC mobile payment protocol based on biometrics (SNMPB) using wireless public key infrastructure (WPKI) and universal integrated circuit card (UICC). We propose a procedure for personalising UICC by the client, a procedure of provisioning and personalisation (mutual authentication and key agreement protocol) of mobile payments application with the fingerprint template (FPT) (which is on UICC). By adopting the above procedures end-to-end security with message integrity, authentication, confidentiality, and non-repudiation is ensured. Digital signatures are used to ensure integrity, authentication, and non-repudiation. Symmetric or asymmetric encryption is used to ensure privacy. Client authentication is a method to verify the client's identity. Client authentication can be accomplished in various ways: by using what the client knows such as a password, what the client possesses such as a smart card, or what the client is regarding biometrics. Traditional methods of client authentication such as passwords or smart cards authenticate the user using their knowledge or possessions. A password or a smart card can be easily stolen or given away to others. Thus, we cannot be certain whoever accesses the system is the person who has authorisation. This leads to the consideration of using biometrics as an attribute to authenticate the user. Biometric authentication process guarantees that whoever presents the biometric data is an authentic user but a client's biometric data is in the public domain. A person will leave his fingerprint on any surface he touches or on any computer he operates. Hence, the user cannot keep his biometric data secret in the same way as he can with a password. Once the biometric data is compromised or stolen, it cannot be replaced or regenerated as other methods of authentication (such as a password or smart card authentication) can. Therefore, it is critical to maintain the privacy of biometric data during authentication. Proposed mobile payment protocol is formally verified because 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. Therefore, the design of security protocol is an intuitive process which is severely error-prone so a more rigid protocol is required within which we can safely design secure protocols. The network is assumed to be hostile as it contains 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 out to have security breaches (Muhammad et al., 2006). So formal verification of security protocols is essential as it can detect flaws that lead to protocol failure. So we have verified successfully our proposed mobile payment protocol using Scyther tool and presented with results.

## *1.1 Threat model*

A protocol is a set of rules that followed 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 network is assumed to be hostile as it contains 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 out to have security breaches (Cremers, 2006; Cremers and Lafourcade, 2009). With the advent of electronic and mobile commerce, cryptographic protocols are being adapted for implementing commercial transactions, and there is a need to provide full proof security for these protocols. So formal verification of cryptographic protocols is essential as it can detect flaws that have led to protocol failure. There are some formal methods like BAN logic and automated formal verification tools such as AVISPA and Scyther tools (Me and Strangio, 2005; Ahamad et al., 20012; Armando et al., 2005) for verifying the security protocols. These tools differ in their input language and also in the way they verify the protocols and provide the output. Though these tools eliminate the possibility of human error, but still the selection of these automated tools is crucial in verifying the correctness of security protocols. Our proposed mobile payment protocol is verified using BAN logic and Scyther tool, and the results show that the proposed protocol is free from attacks.

### *1.2 Contributions made by us*

1 A mobile payment protocol is proposed between the personalised mobile payment application (MPA) on UICC.

2 Our protocol proposed in the UICC of mobile device is considered a tamper resistant device UICC is, therefore, a secure signature creation device (SSCD) because the signature processes are performed in the UICC and the private key never leaves the WIM. Non-repudiation, as a result, is ensured in devices where the private key is stored in WIM.

3 Merchant communicates with the acquirer but not with the payment gateway (PG).

4 Our proposed mobile payment protocol originating from MPA (which is on UICC) to the bank server realises fair exchange, which ensures confidentiality, authentication, integrity, and non-repudiation, prevents double spending, overspending and money laundering. In addition to these, SNMPB withstands replay, man in the middle (MITM) and impersonation attacks.

5 Our proposed mobile payment protocol achieves the accountability properties of mobile payment transaction as we have successfully completed the accountability analysis of the proposed mobile payment protocol using BAN logic.

6 Our proposed mobile payment protocol is modelled using the high-level formal language security protocol description language (SPDL) and which were verified successfully using Scyther tool.

The rest of the paper is organised as follows: In Section 2, we present gaps founded in the related work. In Section 3, we propose a SNMPB framework based on biometrics using NFC containing a procedure for personalising MPA in the UICC (by the issuer/bank) and a SNMPBs making use of WPKI and UICC. In Section 4, we present formal analysis for the proposed protocol. In Section 5, we present a security analysis of the proposed framework, in Section 6, we present a comparative analysis of our proposed protocol with related work and Section 7 concludes our work.

## 2 Related work

### 2.1 Gaps found in the related work

a Current mobile payment solutions based on biometrics (Gordon and Sankaranarayanan, 2010; Ngo et al., 2011; Ahamad et al., 2013, 2014; Isaac and Zeadally, 2012; Plateaux et al., 2014; Rohunen et al., 2014) store client's credentials in the memory of MPs or on the SIM, MPs and SIM with PKI functionality is personalised by the issuer (usually by MNSP) and service providers like banks install MPAs with the help of MNSP's on the SIM. MPAs cannot be personalised by banks without the intervention of MNSP's.

b Gordon and Sankaranarayanan (2010) have proposed a biometric security mechanism in mobile payments which has the following drawbacks:

- Biometric based authentication using fingerprint takes place at the client side only.
- FPT which is hashed and stored at the time of registration may not match with the FPT at the time of verification because hashing will change if there is a slight change in the FPT also.
- FPT is stored in the memory of the MP (which can be infected by virus)
- Authors did not elaborate how an end-to-end security is ensured.

c Ngo et al. (2011) have proposed a biometric-based secure mobile banking protocol which has the following drawbacks:

- Biometric-based authentication takes place at the client side only.
- Client needs a separate smart card in addition to MP.
- It is not clear how the shared symmetric keys are generated and exchanged between client and bank.
- The biometric template is stored in the memory of the MP.
- Prone to replay attacks as nonce and client's identity are sent in unencrypted form.

d In the existing mobile payment solutions based on biometrics (Gordon and Sankaranarayanan, 2010; Ngo et al., 2011; Ahamad et al., 2013, 2014; Isaac and Zeadally, 2012; Plateaux et al., 2014; Rohunen et al., 2014), client's credentials are generated and stored in the in the memory of MP and could be infected by viruses or can be maliciously replaced, does not ensure secure and reliable communication security, does not ensure end-to-end security in the application layer.

e Existing mobile payment solutions based on biometrics (Gordon and Sankaranarayanan, 2010; Ngo et al., 2011; Ahamad et al., 2013, Isaac and Zeadally, 2012; Plateaux et al., 2014; Rohunen et al., 2014) do not achieve non-repudiation property.

f Gordon and Sankaranarayanan (2010), Ngo et al. (2011), Ahamad et al. (2013), Isaac and Zeadally (2012), Plateaux et al. (2014) and Rohunen et al. (2014) were not formally verified with automated tools like Scyther (Ahamad et al., 2012; Armando et al., 2005) or AVISPA (Me and Strangio, 2005) tools.

# 3 Our proposed mobile payment protocol (SNMPB)

## *3.1 Preliminaries*

Mobile network service provider (MNSP)/mobile network operator (MNO), bank/issuer, certifying authority (CA), PG, UICC, FS and acquirer (A) are the entities involved in this framework; their roles can be found in Tseng et al. (2003). This protocol follows the procedure given (Tseng et al., 2003) for personalisation of UICC by the client and MPA (on the UICC) by the issuer/bank.

*FS*: The FS is an important part of our protocol, by which the raw fingerprint is collected. In this process, the MP acts as the data collection and the signalling processing subsystems. It starts by acquiring raw biometric data from the FS at the beginning of the transaction. Then, the raw fingerprint data is processed in the MP. After that, the extracted features are sent to the UICC via an APDU command. The MBA has the responsibility of matching the acquired sample using matching algorithms such as Euclidean Space and City block distance with the client's stored template and makes a decision accordingly. The result of the matching is sent back to the MP for further action.

### *3.1.1 Assumptions and proposed internal architecture of UICC*

a Every UICC will have its platform certificate issued by CA.

b Every client will have his/her own certificate issued by CA.

c Every MPA will have its own certificate issued by CA.

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

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

f CA maintains certificates in its directory, OCSP, and CRL

g CA also acts a trusted service manager (TSM).

Bank maintains personalisation and provisioning server which is responsible for transferring the application and personalisation data to UICC. Provisioning and personalisation of MPA is done over the air (OTA) by the bank.

### *3.1.2 Proposed internal architecture of UICC*

UICC contains the following applications:

- *Mobile biometric application (MBA)*: MBA is a separate application realising the function of biometric authentication situated inside the UICC. Since the UICC card contains confidential information, such as credit card data, client's credentials and MPAs, the role of the MBA is to guarantee that this information can only be accessed if the legitimate phone holder is successfully authenticated. This is achieved by combining the use of the MP and a biometric device. To do this, the MBA must receive fresh biometric data from the cell phone. We assume that the client has already stored his/her FPT data inside the UICC.
- *MPA*: MPA module is a separate application realising the function of payment situated inside the UICC. After the successful authentication (using NRP and fingerprint) of the client by MBA, the MP sends a message to this application requesting the credit card information according to the APDU command. MPA stores the client's personal sensitive information such as the credit card number, shared symmetric keys and issuer's certificate.
- *Wireless identity module (WIM)*: 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 the WIM application be 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 the mobile browser and can establish a session with WIM application residing inside the 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 the crypto functions with the private key residing in the smart card. However, a significant restriction with this solution is that the manufacturer has to provide functionality for communication between WIM and the script running in the browser.

### *3.1.3 Procedure for the personalisation of MPA (on the UICC) by the issuer/bank in the proposed SNMPB protocol*

The architecture of our proposed mobile payment protocol SNMPB is shown in Figure 1. There are three layers of security in our proposed SNMPB protocol; physical infrastructure layer security, communication layer security, and application layer security. Physical infrastructure layer security is about GSM and GPRS security which is vulnerable to many attacks. Secure and reliable end-to-end communication between UICC and the remote bank server is ensured using SSL/TLS and TCP at the communication layer. Security at the application layer is ensured using HTTPS and our proposed mobile payment protocol. Provisioning is the process of installing a payment application on a UICC. Personalisation is the process of putting data unique to a client into the MPA. This includes providing the necessary cryptographic material required by the UICC or application to allow installation or personalisation. 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 algorithm of the personalisation of a MPA is as follows.

**Algorithm 1** Personalisation of MPA (which h is in UICC) by the bank/issuer as shown in Figure 1

Step 1: Client initiates the process of personalisation of mobile payment application (MPA) by keeping his/her finger on the fingerprint scanner (FPS).

Step 2: Fingerprint scanner (FPS) sends the captured fingerprint template (FPT) to the mobile biometric application (MBA)

Step 3: Fingerprint template (FPT) is stored in the mobile payment application (MPA).

Step 4: $UC \to B : \{M1; SIG_B^C(MS1)\}_{k_B}, cert_C$ Where $MS1 = \{AI, phno, FPT, NRP, T_c, N_c\}$

/* before initiating the process of personalisation, the client validates the bank's certificate using the certificate validation procedure (Cremers and Lafourcade, 2009)*/

Step 5: $B \to UC : \{M2; SIG_C^B(MS2)\}_{kC}, cert_B$ *Where*
$MS2 = \{AI, phno, K_{bc}, FPT, T_b, N_c, N_b\}$.

/* upon receiving the message B checks the authenticity of the message, if the checks are successful then the client sends $Step5 : B \to UC : \{M2; SIG_C^B(MS2)\}_{kC}, cert_B$ */

```
  IF Verification (of digital signature on M1) $(IG_B^C(MS1)) = TRUE \cdots$ {
      /*Authenticity and Integrity of message MS2 is not compromised*/
      Go to Step 5}
      Else   {
      /* Authenticity and Integrity of message MS1 is compromised*/
      Exit}
```

$\{M1, SIG_B^C(M1)\}_{k_B}, cert_C$ Where $MS1 = \{AI, phno, FPT, NRP, T_c, N_c\}$, Bank decrypts the received message from UICC using his private key and checks the authenticity of $SIG_B^C(M1)$, checks the timestamps and nonce if all the checks are successful then it generates a shared symmetric key $K_{bc}$ between the B and UC. Bank sends $\{M2, SIG_C^B(M2)\}_{k_B}, cert_B$ to UC containing $MS2 = \{AI, phno, K_{bc}, FPT, T_b, N_c, N_b\}$ 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 (Cremers and Lafourcade, 2009). The key set $K_{bc}$ (with {1, 2, 3, n}) is generated from the secret key $K_{bc}$ and is stored in mobile payment application of the UICC at the client end and in the bank server.*/

All the six steps are shown in Figure 1

Step 6: $UC \to B : \{M3, MAC0\}_{K_{cb}}$ Where M3 = {FPT, AI, Ack, $K_{bc}$, $N_B$, $T_B$, $N_c$}

/* B receives $\{M3, MAC0\}_{K_{cb}}$ from UC containing M3 = {FPT, AI, Ack, $K_{bc}$, $N_B$, $T_B$, $N_c$}
$MAC0 = h(K_{cb}, M3)$*/

```
  IF (Verf(MAC0) = TRUE) {
/*Authentication of/Client, Confidentiality and Integrity of M0 are successfully verified*/
        Bank maps $K_{cb}$ to AI of Client}
          Else   {
        /* Authentication of/Client, Confidentiality and Integrity of M0 are not verified*/
        Exit   }}
```

**Figure 1** Personalisation of MPA in UICC (see online version for colours)

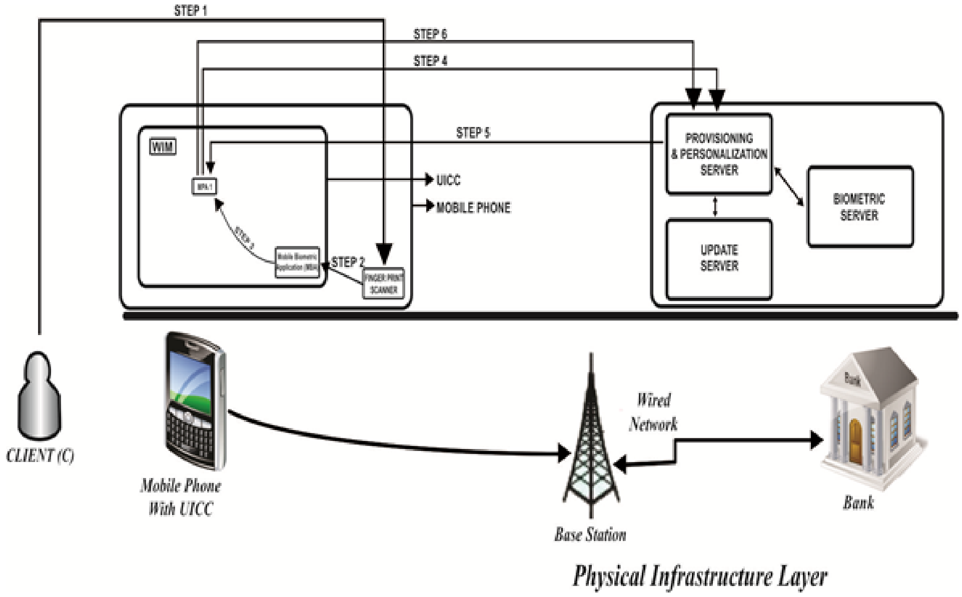
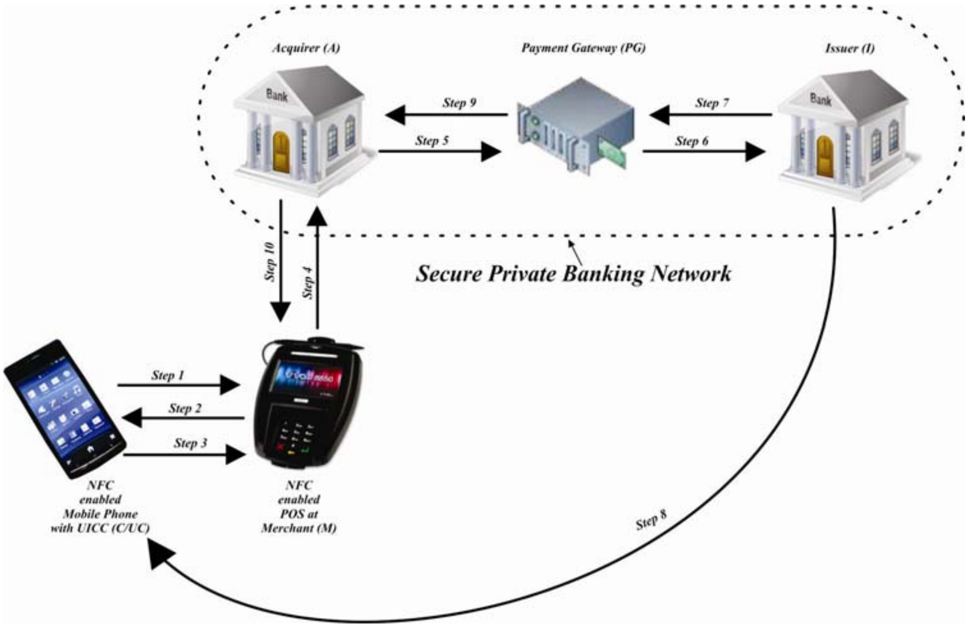


**Figure 2** Biometric based mobile payment protocol (see online version for colours)



## *3.2 Proposed biometric based proximity mobile payment protocol*

We consider a scenario in which a client has an NFC-enabled MP with UICC as secure element. The merchant has the NFC-enabled point of sale (POS) and all the items (in the store) chosen by the client are all tagged with NFC tags. The 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.

All the steps (i.e., from 1 to 10) involved in our proposed biometric-based proximity mobile payment protocol are shown in Figure 2.

Step 1 $C \rightarrow M : \{MS4, SIG_M^C(MS4)\}_{K_M}$

Where $MS4 = \{ID_C, N_C, T_C\}$

Client (C) approaches one of the several NFC enabled POS with chosen tagged items at a department store and sends $\{MS4, SIG_M^C(MS4)\}_{K_M}$ from his NFC enabled MP (with UICC as secure element) to NFC enabled POS containing $\{ID_C, N_C, T_C\}$.

Step 2 $M \rightarrow C : \{MS5, SIG_C^M(MS5)\}_{K_C}$

Where $MS3 = \{OI_M, HOI_M, TID_M, Amt_{M, IDC}, N_C, T_C, ID_M, N_M, T_M, LI_M\}$

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

Step 3 $C \rightarrow M : \{MS6, SIG_M^C(MS6)\}_{K_M}$

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

Where $PI = \{AI, FPT_C, HOI_C, TID_C, AMt_C, ID_C, N_C, T_C, ID_M, LI_M, LI_C\}$

Client (C) decrypts $\{MS5, SIG_C^M(MS5)\}_{K_C}$ using his private key and gets $MS5$ validates the certificate using the procedure (Cremers and Lafourcade, 2009). After successful verification and validation, the client sends $\{MS6, SIG_M^C(MS6)\}_{K_M}$ to the merchant.

Step 4 $M \rightarrow A : \{MS7, IG_A^M(MS7)\}_{K_A}$

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

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

Step 5 $A \rightarrow PG$: $\{MS8\}$

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

Acquirer decrypts $\{MS7, SIG_A^M(MS7)\}_{K_A}$ 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 nonce $N_C = N_M$.

If all the checks are found to be successful then it keeps a copy of the received message MS7 and authorises the order information (OI). Then, the acquirer forwards $\{(MS8)\}$ message to the PG $MS8 = \{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{ci}}, LI_M, LI_C\}$ via the secure private banking network (PBN).

Step 6 $PG \rightarrow I$: $\{MS9\}$

Where MS9 = $\{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{ci}}, LI_M, LI_C\}$

PG receives $MS8$ from the acquirer through the PBNs, which is very secure. PG will perform the following verifications from the M8 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 nonce $N_C = N_M$.

If all the checks are found to be successful then it keeps a copy of the received message MS8 and forwards MS9 = $\{HOI_M, TID_M, Amt_M, HOI_C, TID_C, Amt_C, ID_C, N_C, T_C, ID_M, N_M, T_M, (PI)_{K_{ci}}, LI_M, LI_C\}$ to the Issuer (I)

Step 7 $I \rightarrow PG$: $\{(MS10)\}$

Where $MS10 = = \{AutHorization\ of\ PI\}$

Issuer receives $MS9$ from the PG through the PBNs, which is very secure. Issuer (I) will perform the following checks from $M9$ it has received.

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

Decrypts the PI using the symmetric key shared between the issuer and client

a Checks the clients account for sufficient funds

b Checks if $FPT_C = FPT_I$

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 nonce $N_C = N_M$.

If all the checks are successful, it authorises the PI and sends $\{(MS10)\}$ to $PG$.

Step 8 $I \rightarrow C : \{MS11, SIG_C^I(MS11)\}_{K_C}$

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

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

Step 9 $PG \rightarrow A$: $\{(MS12)\}$

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

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

Step 10 $A \rightarrow M : \{MS13, SIG(MS13)\}_{K_M}$

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

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

### *3.3 Evidence for resolving disputes*

Our proposed mobile payment framework collects evidence which helps in resolving disputes among the stakeholders. We use the transaction counters, transaction log, forensics mode and cryptographic audit log techniques to collect evidence for resolving disputes.

- *Transaction counters*: MPA maintains transaction counters that are incremented at the start of every transaction. The use of the transaction counter as an investigation tool does not require any changes to the UICC but does require the development of procedures to extract it from the banks' logs and also from the legitimate MPA. Above all, we need a regulatory change. For example, banks instruct their customers to cut up the MPA at once if there is a dispute, which is contrary to the customer's interest.
- *Transaction log*: MPA maintains a log of recent transactions. If the UICC is still in the customer's possession then the presence or absence of the disputed transaction in the MPA log is convincing evidence as to whether the legitimate MPA was used for the transaction.
- *Forensics mode*: MPA is placed into a forensics mode MPA could unlock the transaction log so that it could be read, and allow access to internal risk analysis counters which could be correlated with bank logs.
- *Cryptographic audit log*: A weakness of all of the above approaches is that they still depend on the bank's logs for reliability and so do not meet the criterion of complete system disclosure. Past experience sadly suggests that banks in some countries will drag their feet over retaining logs and making them available; and that the regulators in these countries will be reluctant to force them. (The two properties are of course related.) So how can a bank in a well-regulated country protect its cardholders when they travel and transact in a poorly-regulated one? In order solve this problem, MPA implements forward secure audit log provides a lot of protection while storing log records on the MPA issuer's server to avoid limitations on bank card memory. The MPA would be initialised not just with a key used for authentication codes, but with an audit key that is also unique to each MPA (even if this card replaces a card which seemed to fail personalisation). The audit key is updated on each new transaction and a forward-secure MAC (Fun et al., 2008) is computed on the transaction (including the result of PIN verification).

# 4 Formal verification of SNMPB

A protocol is a set of rules that followed 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 network is assumed to be hostile as it contains 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 out to have security breaches (Cremers, 2006; Cremers and Lafourcade, 2009). With the advent of electronic and mobile commerce, cryptographic protocols are being adapted for implementing commercial transactions, and there is a need to provide accountability for protocol participants. So formal verification of cryptographic protocols is essential as it can detect flaws that have led to protocol failure. The accountability analysis of mobile payment protocol concerns about the ability to show that particular parties are responsible for transactions. Particularly, engaging parties must be able to prove to a dispute resolver (verifier) that they are honest for the transaction relevant to them. There are many accountability logics for analysing the accountability of mobile payment protocol. They are BAN logic, Kailar's logic, Kessler and Neumann's (KN) logic and Kungpisdan's (KP) logic. Out of these BAN logic is most appropriate to analyse mobile payment transaction in wireless and mobile networks. There are some Automated Verification tools such as AVISPA and Scyther tools (Me and Strangio, 2005; Ahamad et al., 2012; Armando et al., 2005) for verifying the security protocols. These tools differ in their input language and also in the way they verify the protocols and provide the output. Though these tools eliminate the possibility of human error, but still the selection of these automated tools is very important in verifying the correctness of security protocols. Our proposed mobile payment protocol has been analysed against the BAN logic and the results revealed that the proposed protocol achieved the accountability property in mobile transaction. In addition to these, we have successfully verified SNMPB using Scyther tool.

## *4.1 Using BAN logic*

### *4.1.1 Assumptions about keys and secrets:*

'P' is a set of engaging parties consisting of $\{C/W, I, M, A, PG\}$ where $C/W$ means client or WPKI-UICC, $I$ means issuer, $M$ means merchant, $A$ means acquirer and $PG$ means PG. These assumptions specify the initial setup of keying and secret material for authentication purposes. The certification authority knows the public keys of every participant and each participant knows the certification authority's public key (**AS1**, **AS2**). The customer can use the non-repudiation PIN, i.e., **NRP** to authenticate himself to his personal trusted device (**AS3**).

A1 CA **believes** $(\forall P \in \{A, M, C, I, PG\} \overset{K_p}{\mapsto} P)$. Certification authority CA believes that $K_a$, $K_m$, $K_c$, $K_i$ are the public keys to communicate with the acquirer $A$, the merchant $M$, the client $C$, issuer $I$ and $PG$, respectively.

A2 $\acute{\bar{a}}\, P\ \{A, M, C, I, PG\}$. $P$ **believes** $(\overset{K_{ca}}{\mapsto} CA)$. 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.

A3 *W*, *C* **believes** $C \overset{NRP}{\rightarrow} W$. The personal trusted device *D* believes that NRP is a secret between *C* and *W* and is only known to *C* and *W*.

A4 C, I **believes** $C \overset{K_{ci}}{\rightarrow} I$. The client *C* and *I* **believes** that $K_{ci}$ is a secret shared between client *C* and issuer *I* and is only known to *C* and *I*.

### *4.1.2 Assumptions about freshness*

Assumption **AS5** specifies fresh quantities. For instance, if the acquirer **A** sees quantity $n_a$ in a message then the acquirer can derive that the message is not a replay. Assumption A6 specifies that certificates are within their validity period and thus are not expired and timestamps which ensures timeliness.

AS5 *A* **believes fresh** $(n_a)$, *M* **believes fresh** $(n_m)$, *C* **believes fresh** $(n_c)$, *I* **believes fresh** $(n_i)$.

Every party believes that any nonce it generates is fresh, that is, the same nonce is never used in two different execution instances of the protocol.

AS6 *A* **believes fresh** $(T_c')$, *A* **believes fresh** $(T_m')$, *M* **believes fresh** $(T_a')$, *C* **believes fresh** $(T_i')$, *I* **believes fresh** $(T_c')$ and *D* **believes fresh** $(T_i')$, where $T_c', T_m', T_a'$ and $T_i'$ are validity periods in certificates every party believes certificates are within the validity period. $T_c$, $T_m$, $T_a$ and $T_i$ are the timestamps generated by the Client, Merchant, Acquirer and Issuer which ensures **timeliness** of the message.

### *4.1.3 Assumptions about channels*

These assumptions specify that the personal trusted device and the entities which are in the banking private network have secure input and output channels. Assumptions (AS7, AS8) talks about the secure channels of PTD (W) and assumptions (AS9–AS14) talks about secure input and output channels of the entities involved in the banking private network. Here *In*, *O* are input and output channel.

AS7 *C* believes $\overset{Ow}{\prec}$ *W*, *C* **believes timely** $(O_w)$.Client *C* believes that $O_w$ is a secure and timely channel from *C*, that is, messages on $O_w$ are known to have been sent by W recently.

AS8 *W* **believes** $\overset{Inw}{\prec}$ *C*, *W* **believes timely** $(In_w)$. W believes that $In_w$ is a secure and timely input channel, that is, any message on $In_w$ is known to have been sent by *C* recently.

Assumptions from 9 to 14 use PBN

AS9 **A believes** $\overset{o_x}{\prec}\begin{Bmatrix} I \\ PG \end{Bmatrix}$ **A believes timely** $O_x$, where ‘$x$’ is ‘$i$’ and ‘$pg$’

AS10 **I believes** $\overset{o_x}{\prec}\begin{Bmatrix} A \\ PG \end{Bmatrix}$ **A believes timely** $O_x$, where ‘$x$’ is ‘$a$’ and ‘$pg$’

AS11 **PG believes** $\overset{o_x}{\prec}\begin{Bmatrix} I \\ A \end{Bmatrix}$ **A believes timely** $O_x$, where ‘$x$’ is ‘$i$’ and ‘$a$’

AS12 **A believes** $\overset{Inp_x}{\prec}\begin{Bmatrix} I \\ PG \end{Bmatrix}$ **A believes timely** $Inp_x$, where ‘$x$’ is ‘$i$’ and ‘$pg$’

AS13 **I believes** $\overset{Inp_x}{\prec}\begin{Bmatrix} A \\ PG \end{Bmatrix}$ **I believes timely** $Inp_x$, where ‘$x$’ is ‘$a$’ and ‘$pg$’

AS14 **PG believes** $\overset{Inp_x}{\prec}\begin{Bmatrix} I \\ A \end{Bmatrix}$ **PG believes timely** $Inp_x$, where ‘$x$’ is ‘$i$’ and ‘$a$’.

**AS11** means acquirer **A believes** that the *I*’s and *PG*’s display is a secure and timely channel from *I* and *PG*, i.e., messages on $O_i$ and $O_{pg}$ are known to have been sent by I and PG recently.

**AS14** means acquirer **A believes** that $Inp_x$ is a secure and timely input channel to *I* and PG, i.e., any messages on $Inp_x$ (‘$x$’ can be ‘$i$’ and ‘$pg$’) known to have been sent by *I* and PG recently.

### 4.1.4 Assumptions about trust

These assumptions specify the level of trust for each participant. The certification authority is trusted to correctly certify principals (AS15).

AS15 $\forall P, Q \in \{A, M, C, I, PG\}$, *P* **believes** CA **controls** $\overset{K_q}{\mapsto} Q$. Every principal trusts the certification authority CA to correctly certify other principals.

AS16 *A* **believes** (*M* **controls** $MQ_m$). The acquirer trusts the merchant to control the merchant quote (MQ) it issues, *A* **believes** (*C* **controls** *PO*). The acquirer trusts the user to control the purchases for which he utters payment orders.

AS17 *M* **believes** *A* **controls** response, *C* believes *I* **controls** response, i.e., *M* and *C* trusts the acquirer and issuer to generate correct response.

AS18 $\acute{\bar{a}}$ belief *X*, CA **believes** (*W* **controls** (*C* **believes** *X*)). The certification authority CA trusts the client’s *C*’s personal trusted device to relay user’s beliefs.

AS19 $\acute{\bar{a}}$ belief *X*, *A*, *I* **believes** CA **controls** (*W* **controls** (*C* **believes** *X*)). The acquirer and issuer trust the Certification Authority to properly certify the ability of the trusted device to correctly forward Client’s beliefs.

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

The client (C) unlocks the mobile payment application software using his NRP, which is stored in the tamper resistant UICC (W). The client (C) uses non-repudiation PIN (NRP) to convince the UICC (W) of his identity as given in AS2. The idealised form of step 0 is

$$C \rightarrow W \langle Application\rangle_{NRP} \textbf{ On } Inp_w$$

Since W receives NRP through secure and timely channel as given in **AS7 and AS8** then the (W) achieves the belief 'W **believes** C **believes** (application)', i.e., W believes that the client (C) is authorised to use the application loaded in the W. '**on**' specifies secure channel from client (C) to W.

Step 1: $C \rightarrow M : SIG_{C_M}(MS1), WSLC_C$

$MS1 = \{PO, nc, Tc\}$

Step 2: $M \rightarrow C : SIG_{M_A}(MS2), WSLC_M$

$MS2 = \{MQ, TID, nc, nm, Tm, Tc\}$

The idealised form is $\{SIG_{C_M}(MS1)\}, \{\overset{Y_w}{\mapsto}, W$ **controls** C **believes** X}

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 $MS1 = \{PO, nc, TC\}$

From **AS6** and statement (1) we conclude

"M **believes fresh** $(T_c')$" …statement (2).

From **AS6** "M **believes fresh** $(T_c)$"……Statement (3)

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 (4),

From Statement (1) to Statement (4)

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

Step 3: $C \rightarrow M : SIG_{C_M}(MS3)$

$MS3 = \{TransCertC, (PI)K_{ci}, MQ, TID, nc, nm, Tm, Tc, Amt\}$

The idealised 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 (5)

Where $MS2 = \{MQ, TID, nc, nm, Tm, Tc\}$

From **A**S6 and statement (5) we conclude

"C believes fresh $(T_m')$" …statement (6).

From AS6

C believes fresh $(T_m)$……statement (7)

From statement (5) to statement (7)

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

Step 4: $M \rightarrow A : SIG_{M_A}(MS4), WSLC_A$

$MS4 = \{TransCertC, (PI)K_{ci}, MQ, TID, nc, nm, Tm, Tc, Amt\}$

The idealised 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 (8)

Where $MS3 = \{TransCertC, (PI)K_{ci}, MQ, TID, nc, nm, Tm, Tc, Amt\}$

From **AS6** and statement (8) we conclude

"M **believes** fresh $(T_c')$" …statement (9).

From **AS6**

M **believes fresh** $(T_c)$…...statement (10)

From statement (8) to statement (10)

M **believes** $\{SIG_{C_M}(MS3)\}$

Step 5: $A \rightarrow PG : SIG_{A_{PG}}(MS5)$

$MS5 = \{TransCertC, (PI)K_{ci}, TID, nc, nm, Tm, Tc, Amt\}$

The idealised 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 (11)

From **AS6** and statement (11) we conclude

"A **believes fresh** $(T_m')$" …statement (12)

From **AS6**

A **believes fresh** $(T_m)$…...statement (13)

From statement (11) to statement (13)

A **believes** $\{SIG_{M_A}(MS4), WSLC_A\}$

Step 6: $PG \rightarrow I : SIG_{PG_I}(MS6)$

$MS5 = \{TransCertC, (PI)K_{ci}, TID, nc, nm, Tm, Tc, Amt\}$

After receiving the message $SIG_{A_{PG}}(MS5)$ PG decrypts the signature and recovers MS5 using his private key.

PG **believes** A **said** {MS5})…...statement (14)

From **AS6** and statement (14) we conclude

"PG **believes fresh** $(T_m')$" …statement (15)

From **AS6**

PG **believes fresh** $(T_m)$…...statement (16)

From statement (14) to statement (16)

PG **believes** $\{SIG_{A_{PG}}(MS5)\}$

Step 7: $I \rightarrow PG : SIG_{I_{PG}}(MS7)$

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

After receiving the message $SIG_{PG_I}(MS6)$ I decrypts the signature and recovers MS6 using his private key.

I **believes** PG **said** {MS6})…...statement (17)

Where $MS6 = \{TransCertC, (PI)K_{ci}, TID, nc, nm, Tm, Tc, Amt\}$

From **AS6** and statement (17) we conclude

"I believes fresh $(T'_m)$" …statement (18)

From **AS6**

I **believes fresh** $(T_m)$…...Statement (19)

**C believes** $C \overset{Kci}{\leftrightarrow} I$

**I believes** $C \overset{Kci}{\leftrightarrow} I$

C **believes** I **believes** $C \overset{Kci}{\leftrightarrow} I$

**I believes** C **believes** $C \overset{Kci}{\leftrightarrow} I$

**I believes** C **said** $(PI)$ $(T_m)$ …...statement (20)

From statement (17) to statement (20)

I **believes** $\{SIG_{PG_I}(MS6)\}$

Step 8: $PG \rightarrow A : SIG_{PG_A}(MS8)$

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

The idealised form is $\{SIG_{I_{PG}}(MS7)\}$

After receiving the message $SIG_{I_{PG}}(MS7)$ PG decrypts the signature and recovers MS7 using his private key.

PG **believes** I **said** {MS7})…...statement (21)

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

From **AS6** and statement (21) we conclude

"PG **believes fresh** $(T'_m)$" …statement (22)

From **AS6**

PG **believes fresh** $(T_m)$…...statement (23)

From statement (21) to statement (23)

**PG believes** $\{SIG_{I_{PG}}(MS7)\}$

Step 9: $A \rightarrow M : SIG_{A_M}(MS9)$

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

After receiving the message $SIG_{PG_A}(MS8)$ **A** decrypts the signature and recovers MS8 using his private key.

A **believes** PG **said** {MS8})…...statement (24)

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

From **AS6** and statement (24) we conclude

"A believes fresh $(T'_m)$" …Statement (25)

From **AS6**

A *believes fresh* $(T_m)$…...statement (26)

From statement (24) to statement (26)

A **believes** $\{SIG_{PG_A}(MS8)\}$

Step 10: $M \rightarrow C : SIG_{M_C}(MS10)$

$MS10 = \{TID, Success, Amt\}$

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 (27)

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

From **AS6** and statement (27) we conclude

"M believes fresh $(T'_c)$" …statement (28)

From **AS6**

M **believes fresh** $(T_c)$…...statement (29)

From statement (27) to statement (29)

M **believes** $\{SIG_{A_M}(MS9)\}$

This proves that our proposed mobile payment protocol has been analysed against the BAN logic tool and the result revealed that the proposed protocol achieves the accountability property in mobile payments.

### *4.2 Formal verification of SNMPB using Scyther tool*

Scyther is a tool used for security protocol verification, where it is assumed that all the cryptographic functions are perfect. 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 here can be done using a bounded or an unbounded number of sessions. The language used to write protocols in Scyther is SPDL (Ahamad et al., 2012; Armando et al., 2005). We have evaluated our proposed SNMPB using the Scyther model checking security protocol verification tool. Scyther is an automatic push-button tool for the verification and falsification of security protocols. SNMPB protocol is written using the SPDL and then validated using 'automatic claim' and 'verification claim' procedures in the Scyther tool. Results are presented in Appendix A.

## 5 Security analysis of SNMPB

In order to analyse our proposed mobile payment protocol (SNMPB), a generic set of security goals are defined in subsequent subsections. The security goals are categorised into four sections namely data security, client security, bank (issuer and acquirer) security and merchant security. We present our analysis of the protocols with respect to each security goal.

## 5.1 Data security

### 5.1.1 Third party

*Goal*: In the proposed mobile payment protocol (SNMPB), any third party not involved in the payment system should not obtain access to the participant's transactional data or their secret keys that will lead to a successful execution of a payment (or deposit) protocol and the mobile payment system should withstand attacks.

All the entities involved in the SNMPB protocol store their credentials (private keys, NRP and certificates) in tamper resistant hardware tokens so their credentials cannot be compromised. All the entities involved in the SNMPB protocol will transmit data using encryption and digital signature so any third entity will not be able to gain access to the participant's transactional data thereby achieving data confidentiality, entity authentication, data integrity and non-repudiation. Our proposed mobile payment protocol (SNMPB) withstands the following attacks.

#### 5.1.1.1 Replay attacks

If an intruder (In) wants to impersonate a legitimate client by replaying the client's transmitted message, then the 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.

#### 5.1.1.2 Impersonating attack

An intruder (In) tries to impersonate a client C to CA, which results in CA being cheated. Since In does not have C's private key he fails in doing so. As a result, impersonating attacks fail in our protocol.

#### 5.1.1.3 MITM attack

MITM attack is a common attack of intercepting communications in banking protocols. The attacker is able to read, insert, and modify messages in the intercepted communication. The attack targets the integrity of the protocol. Our proposed protocol SNMPB withstands this attack because the intruder (In) does not have receiver's private key.

### 5.1.2 Secrecy

*Goal*: In the proposed mobile payment protocol (SNMPB), from the view of the client, the merchant should not have access to the client's payment information (PI), i.e., payment secrecy should be achieved and the bank (I, A, PG) should not have access to the client's OI, i.e., order secrecy should be achieved. In addition to this transaction privacy is achieved from PG and eavesdropper.

In our proposed mobile payment protocol (SNMPB) payment secrecy is achieved by encrypting the PI using the secret symmetric key which is shared between the client (C) and the issuer (I). The merchant will not be able to decrypt the 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 the eavesdropper.

```
claim_C2(C, Secret, PI);
claim_C3(C, Secret, OI);
claim_I2(I, Secret, PI);
claim_M2 (M, Secret, OI);
```

### 5.1.3 Uniqueness

*Goal*: In a mobile payment system, every transaction processed should be unique.

In our proposed mobile payment protocol (SNMPB) every transaction is unique. The uniqueness is obtained due to the fresh generation of transaction ID by the merchant, and its verification by the bank. Every transaction is also linked to the timestamps and nonce.

```
claim_C4(C, Secret, nm);
claim_C5(C, Secret, nc);
claim_M1 (M, Secret, nm);
```

## 5.2 Client security

### 5.2.1 Authentication

*Goal*: In a mobile payment protocol, the client should obtain an unforgeable proof of the other participant's authenticity before it engages in a protocol with that participant.

In our proposed mobile payment protocol (SNMPB) four factor authentication are ensured

a Knowledge of NRP for accessing the MPA.

b Biometric based authentication using fingerprint.

c Possession of private key and certificate to certify the authenticity of public keys held by the entities involved in the ecosystem.

d Possession of shared symmetric key between the client and the issuer.

```
claim_M3(M, Niagree);
claim_I3(I, Niagree);
claim_C1(C, Secret, Kci);
claim_I1 (I, Secret, Kci);
```

### 5.2.2 Authorisation

*Goal*: In a mobile payment system, the client should obtain unforgeable proof of transaction authorisation by the bank.

In our proposed mobile payment protocol (SNMPB), the client obtains unforgeable proof of transaction authorisation from the issuer at the end of protocol (i.e., whether the transaction is success or failure).

### *5.2.3 Identity protection from the merchant and eavesdropper*

*Goal*: In a mobile payment protocol, the client should be able to achieve identity protection from the merchant and eavesdropper.

Client will be issued an anonymous identity anonid$_{C}$ by the CA after a successful verification of the client's credentials. Client's certificate will have an anonymous identity instead of his/her real identity thereby achieving identity protection from the merchant and eavesdropper.

## *5.3 Bank security (issuer, acquirer and PG)*

### *5.3.1 Authentication*

*Goal*: In a mobile payment protocol, the bank should be presented with an unforgeable proof, certifying the authenticity of the other participants.

Our proposed mobile payment protocol (SNMPB) uses certification authorities, to certify the authenticity of the public keys held by the client, merchant and PG. The client's conversations are only with the merchant. Client, merchant and PG's authenticity is proved by verifying their certificates. In addition to this, the client and issuer shares a symmetric key between them.

```
claim_M3 (M, Niagree);
claim_C6(C, Niagree);
claim_PG1 (PG, Niagree);
claim_C1(C, Secret, Kci);
claim_I1 (I, Secret, Kci);
```

### *5.3.2 Authorisation*

*Goal*: In a mobile payment protocol, the bank (I, A and PG) before it authorises a transaction should obtain an unforgeable proof from the client and merchant, certifying that the client and merchant have agreed to the transaction details and are authorised to proceed with the transaction.

I, A and PG obtains an authorisation proof for transaction from the client and the merchant in the form of, and PI encrypted with the shared symmetric key between the client and issuer containing the payment details, merchant identity and hashed OI. The acquirer authorises OI of the transaction; I authorises OI and PI of the transaction.

```
claim_C2 (C, Secret, PI); claim_I2 (I, Secret, PI);
```

### *5.3.3 Prevents double spending, overspending and money laundering*

*Goal*: In a mobile payment protocol, the issuer should be able to prevent double spending, overspending and money laundering.

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

### *5.3.4 Issuer, acquirer and PG turning malicious*

If any one or all the entities in the PBN (issuer, acquirer and PG) turns malicious then they, also, will not succeed in performing the transaction on behalf of the client (C) because they have no knowledge about the private key of the client (C).

## *5.4 Merchant security*

### *5.4.1 Authentication*

*Goal*: In a mobile payment protocol, the merchant should be presented with an unforgeable proof, certifying the authenticity of the other participants.

Our proposed payment protocol (SNMPB) uses certification authorities, to certify the authenticity of the public keys held by the client, acquirer, issuer and PG.

```
claim_C6(C, Niagree);
claim_A1(A, Niagree);
```

### *5.4.2 Authorisation*

*Goal*: In a mobile payment protocol, the merchant before it authorises a transaction should obtain an unforgeable proof from the client.

Merchant checks the authenticity and integrity of the message *MS*3 received from *C*, verifies digital signature on the message, checks *OI*, *TID*, *nc*, *nm*, *Tm*, *Tc*. If the merchant is convinced about the *TID* and *OI* then only *M* authorises the *OI* thereby achieving order secrecy because *OI* is not known to any of the engaging entities other than *C* and *M*. Therefore, the merchant authorises the transaction after obtaining unforgeable proof from the client.

```
claim_C3(C, Secret, OI);
claim_M2 (M, Secret, OI);
```

## 6 Comparative analysis of SNMPB protocol with related works

See Appendix B.

## 7 Conclusions

This paper proposes

a a procedure for personalising MPA which is on the UICC (by the issuer/bank)

b a SNMPBs by making use of WPKI and UICC.

Client's credentials are generated and stored in the UICC which is a tamper resistant secure element so non-repudiation property is ensured. SNMPB resolves disputes efficiently among stakeholders by collecting evidences using transaction counters, transaction log, forensics mode and cryptographic audit log techniques. SNMPB ensures end-to-end security prevents double spending and over spending. SNMPB withstands replay, MITM, impersonation and multi-protocol attacks as SNMPB is formally verified successfully using BAN logic and Scyther tool.

## References

Ahamad, S.S., Sastry, V.N. and Nair, M. (2013) 'A biometric based secure mobile payment framework', *4th International Conference on Computer and Communication Technology (ICCCT)*, pp.239–246.

Ahamad, S.S., Sastry, V.N. and Udgata, S.K. (2014) 'Secure mobile payment framework based on UICC with formal verification', *IJCSE*, Vol. 9, No. 4, pp.355–370.

Ahamad, S.S., Udgata, S.K. and Sastry, V.N. (2012) 'A new mobile payment system with formal verification', *Int. J. Internet Technology and Secured Transactions*, Vol. 4, No. 1, pp.71–103.

Armando, A. et al. (2005) 'The AVISPA tool for the automated validation of internet security protocols and applications', *Proceedings of Computer Aided Verification'05 (CAV), Lecture Notes in Computer Science*, Springer, Vol. 3576, pp.281–285.

Cremers, C. and Lafourcade, P. (2009) 'Comparing state spaces in automatic security protocol verification', *Proceedings of the 7th International Workshop on Automated Verification of Critical Systems (AVoCS'07)*, pp.49–63.

Cremers, C.J.F. (2006) *Scyther-Semantics and Verification of Security Protocols*, PhD thesis, Eindhoven University of Technology.

Fun, T.S., Beng, L.Y., Likoh, J. and Roslan, R. (2008) 'A lightweight and private mobile payment protocol by using mobile network operator', *Proceedings of International Conference on Computer and Communication Engineering (ICCCE 2008)*, pp.162–166.

Gordon, M. and Sankaranarayanan, S. (2010) 'Biometric security mechanism in mobile payments', *Seventh International Conference on Wireless and Optical Communications Networks (WOCN)*, pp.1–6.

Ibrahim, M.U. and Abbas, M. (2014) 'Biometric authentication via facial recognition', *Journal of Information Security Research*, June, Vol. 5, No. 2, pp.61–68.

Isaac, J.T. and Zeadally, S. (2012) 'An anonymous secure payment protocol in a payment gateway centric model', *Procedia Computer Science*, Vol. 10, pp.758–765.

Me, G. and Strangio, M.A. (2005) 'EC-PAY: an efficient and secure ECC-based wireless local payment scheme', *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05).*

Muhammad, S., Furqan, Z. and Guha, R.K. (2006) 'Understanding the intruder through attacks on cryptographic protocols', *Proceedings of the 44th ACM Southeast Conference (ACMSE2006)*, March, pp.667–672.

Ngo, H.H., Dandash, O., Le, P.D., Srinivasan, B. and Wilson, C. (2011) 'Formal verification of a secure mobile banking protocol', *Advances in Networks and Communications, Communications in Computer and Information Science*, Vol. 132, Part 2, pp.410–421, DOI: 10.1007/978-3-642-17878-8_42.

Plateaux, A., Lacharme, P., Rosenberger, C. and Jųsang, A. (2014) 'One-time biometrics for online banking and electronic payment authentication', *International Conference on Availability, Reliability and Security (ARES), Workshop on Security and Cognitive Informatics for Home and Defense*, Fribourg, Switzerland, September, pp.179–193.

Rohunen, A., EtelĆ¤perĆ¤, M., Liukkunen, K., Tulppo, T. and Chan, K.W. (2014) 'Implementing and evaluating a Smart-M3 platform-based multi-vendor micropayment system pilot in the context of small business', *Journal of Digital Information Management*, February, Vol. 12, No. 1, pp.44–51.

Tseng, Y., Jan, J. and Chien, H. (2003) 'Digital signature with message recovery using selfcertified public keys and its variants', *Applied Mathematics and Computation*, Vol. 136, Nos. 2–3, pp.203–214.

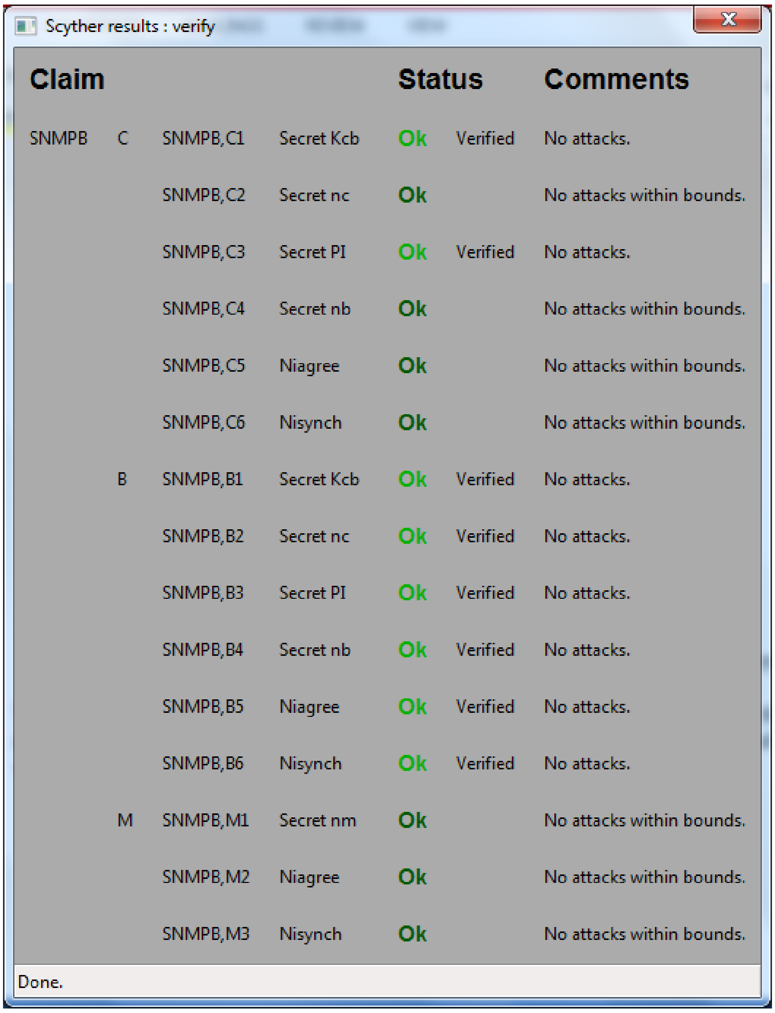
## Appendix A

For security protocol verification, Scyther, an automatic push-button tool, was used. This tool initially functions by assuming that all the cryptographic functions are flawless and can identify any flaws that do arise by the construction of the protocol and then validated using ‘automatic claim’ and ‘verification claim’ procedures in the Scyther tool. Additionally, Scyther can be used to produce potential trace patterns with verification being completed using either a bounded or an unbounded amount of sessions. To write the protocols in Scyther, initially the language used is SPDL (Ahamad et al., 2012; Armando et al., 2005).

```
/* SNMPB (Secure NFC mobile payment framework based on biometrics)*/
/* Mobile payment protocol */
    // PKI
        const pk: Function; secret sk: Function; Inversekeys (pk,sk); usertype Timestamp;
        usertype success;
        usertype PI,Amtc,Amtm,Tc,Tb,Tm,HOIc,HOIm,TIDc,TIDm,AuthOI; // Protocol
        description
        ProtocolSNMPB (C,B, M)
        {
        Role C
        {
        const nc: Nonce; var nb,nm: Nonce;
        const Kcb:SessionKey;
        /* Payment Phase of SNMPB framework */
        send_1 (C,B, {nc,{PI}Kcb,Tc,HOIc,TIDc,Amtc}pk(B)); read_4 (B,C,
        {TIDc,Amtc,nb,nm,HOIc,success}pk(C)); 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,nm: Nonce;
const Kcb: SessionKey;
/* Authentication and Key Agreement Protocol */ read_1 (C,B,
{nc,{PI}Kcb,Tc,HOIc,TIDc,Amtc}pk(B)); send_2 (B,M, {nc,nb,Tc,HOIc,TIDc,Amtc}pk(M));
read_3 (M,B,
{nc,nm,Tc,HOIc,TIDc,Amtc,Amtm,AuthOI,HOIm}pk(B)); send_4 (B,C,
{TIDc,Amtc,nb,nm,HOIc,success}pk(C));
send_5 (B,M, {TIDm,Amtm,HOIm,success}pk(M)); claim_B1 (B, Secret, Kcb);
claim_B2 (B, Secret, nc); claim_B3 (B, Secret, PI); claim_B4 (B, Secret, nb); claim_B5 (B,
Niagree); claim_B6 (B, Nisynch);
}
Role M
{
```

```
const nm: Nonce; varnb,nc: Nonce; constKcb:SessionKey;
/* Authentication and Key Agreement Protocol */ read_2 (B,M,
{nc,nb,Tc,HOIc,TIDc,Amtc}pk(M));
send_3 (M,B, {nc,nm,Tc,HOIc,TIDc,Amtc,Amtm,AuthOI,HOIm}pk(B)); read_5 (B,M,
{TIDm,Amtm,HOIm,success}pk(M));
claim_M1 (M, Secret, nm); claim_M2 (M, Niagree); claim_M3 (M, Nisynch);
}
}
// An untrusted agent, with compromised key const e: Agent;
untrusted e; compromisedsk(e);
```

**Figure 3** The results using 'verification claim' procedure in Scyther tool (see online version for colours)

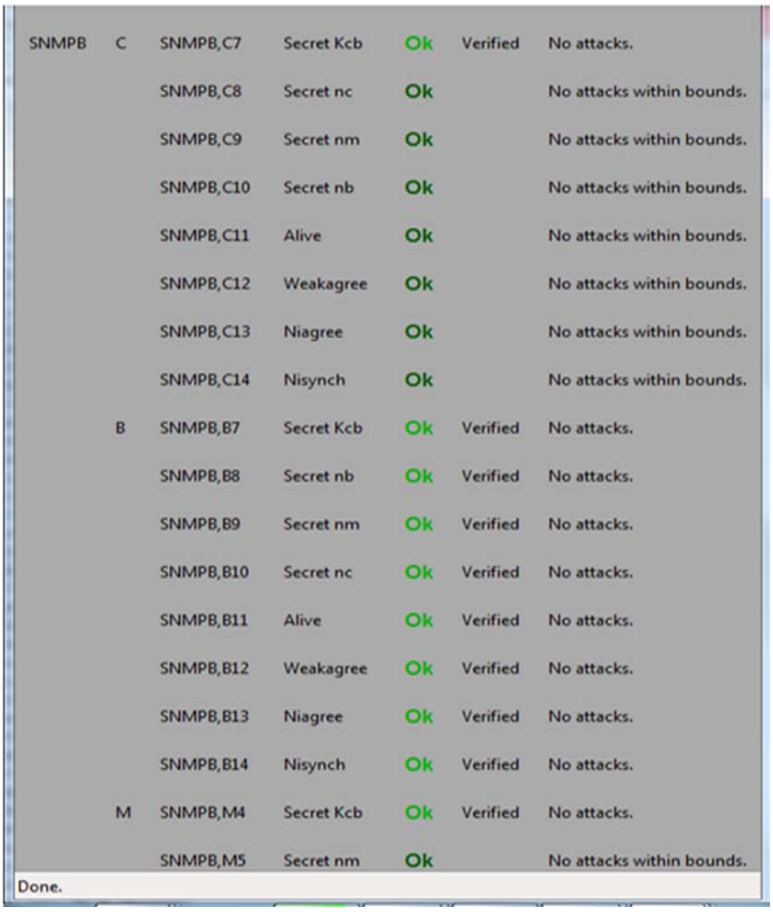


| Claim | | | | Status | | Comments |
|---|---|---|---|---|---|---|
| SNMPB | C | SNMPB,C1 | Secret Kcb | Ok | Verified | No attacks. |
| | | SNMPB,C2 | Secret nc | Ok | | No attacks within bounds. |
| | | SNMPB,C3 | Secret PI | Ok | Verified | No attacks. |
| | | SNMPB,C4 | Secret nb | Ok | | No attacks within bounds. |
| | | SNMPB,C5 | Niagree | Ok | | No attacks within bounds. |
| | | SNMPB,C6 | Nisynch | Ok | | No attacks within bounds. |
| | B | SNMPB,B1 | Secret Kcb | Ok | Verified | No attacks. |
| | | SNMPB,B2 | Secret nc | Ok | Verified | No attacks. |
| | | SNMPB,B3 | Secret PI | Ok | Verified | No attacks. |
| | | SNMPB,B4 | Secret nb | Ok | Verified | No attacks. |
| | | SNMPB,B5 | Niagree | Ok | Verified | No attacks. |
| | | SNMPB,B6 | Nisynch | Ok | Verified | No attacks. |
| | M | SNMPB,M1 | Secret nm | Ok | | No attacks within bounds. |
| | | SNMPB,M2 | Niagree | Ok | | No attacks within bounds. |
| | | SNMPB,M3 | Nisynch | Ok | | No attacks within bounds. |

Done.

The result window in Figure 3 shows that the secrecy of Kcb, PI, OI, nm, nc and the claim for non-injective agreement and non-injective synchronization of all the entities involved are successfully verified using the ‘verification claim’ procedure in Scyther tool.

**Figure 4** Result using ‘automatic claim’ procedure in Scyther tool (see online version for colours)



```
SNMPB   C   SNMPB,C7    Secret Kcb   Ok   Verified   No attacks.
            SNMPB,C8    Secret nc    Ok              No attacks within bounds.
            SNMPB,C9    Secret nm    Ok              No attacks within bounds.
            SNMPB,C10   Secret nb    Ok              No attacks within bounds.
            SNMPB,C11   Alive        Ok              No attacks within bounds.
            SNMPB,C12   Weakagree    Ok              No attacks within bounds.
            SNMPB,C13   Niagree      Ok              No attacks within bounds.
            SNMPB,C14   Nisynch      Ok              No attacks within bounds.
        B   SNMPB,B7    Secret Kcb   Ok   Verified   No attacks.
            SNMPB,B8    Secret nb    Ok   Verified   No attacks.
            SNMPB,B9    Secret nm    Ok   Verified   No attacks.
            SNMPB,B10   Secret nc    Ok   Verified   No attacks.
            SNMPB,B11   Alive        Ok   Verified   No attacks.
            SNMPB,B12   Weakagree    Ok   Verified   No attacks.
            SNMPB,B13   Niagree      Ok   Verified   No attacks.
            SNMPB,B14   Nisynch      Ok   Verified   No attacks.
        M   SNMPB,M4    Secret Kcb   Ok   Verified   No attacks.
            SNMPB,M5    Secret nm    Ok              No attacks within bounds.
Done.
```

The result window in Figure 4 shows that the secrecy of Kcb, PI, OI, nm, nc and the claim for non-injective agreement and non-injective synchronization of all the entities involved are successfully verified using the ‘automatic claim’ procedure in Scyther tool.

## Appendix B

**Table 1** Notations

| *Notation* | *Meaning* | *Notation* | *Meaning* |
|---|---|---|---|
| C, UC, M, I, A, PG, CA, In | Client, UICC, merchant, issuer, acquirer, payment gateway, certifying authority, intruder | NRP | Non-repudiation PIN |
| POS | Point of sale | $Anonid_c$ | Anonymous identity of C |
| $SIG_Y^X(MS)$ | Digital signature generated by entity 'X' intended to be verified by 'Y' and MS is the message received by 'Y' | $PI$ | Payment information |
| $K_{CI}$ | Symmetric key shared between entities Client & Issuer | $AI$ | Account information |
| $N_x$ | Nonce generated by entity 'x' | $LI_M$ | Location information given by merchant |
| $T$ | Timestamp | $LI_C$ | Location information given by client |
| $T_M$ | Timestamp generated by entity 'merchant' | $ID$ | Identity |
| $T_C$ | Timestamp generated by entity 'client' | $ID_C$ | Identity of client |
| $OI$ | Order information | $ID_M$ | Identity of merchant |
| $HOI_M$ | Hashed order information of merchant | $Amt$ | Amount |
| $HOI_C$ | Hashed order information of client | $Amt_M$ | Amount given by merchant |
| $TID$ | Transaction Identity | $Amt_C$ | Amount given by client |
| $TID_M$ | Transaction identity of merchant | UICC | Universal integrated circuit card |
| $TID_C$ | Transaction identity of client | | |

**Table 2** Comparative analysis of SNMPB protocol with the related work

| *Features* / *Protocols* | *Gordon and Sankaranarayanan (2010)* | *Ngo et al. (2011)* | *Ahamad et al. (2014)* | *Ahamad et al. (2013)* | *Isaac and Zeadally (2012)* | *Plateaux et al. (2014)* | *Rohunen et al. (2014)* | *Ibrahim and Abbas (2014)* | *SNMPB* |
|---|---|---|---|---|---|---|---|---|---|
| Authentication | Yes | Yes | Yes | Yes | No | No | No | No | Yes |
| Confidentiality | Yes | Yes | Yes | Yes | No | No | No | No | Yes |
| Integrity | Yes | Yes | Yes | Yes | Nr | No | No | Nr | Yes |
| Non-repudiation | Yes | Yes | Yes | Yes | Nr | No | No | Nr | Yes |
| Client's credentials are generated using OBKG procedure | No | No | Yes | Yes | No | No | No | No | Yes |
| WPKI is implemented in the memory of mobile phone | No | No | Yes | Yes | Yes | Yes | Yes | No | Yes |
| Ensures reliable and secure end to end communication | No | No | Yes | Yes | No | No | No | No | Yes |
| Ensures end to end security at application level | No | No | Yes | Yes | No | No | No | No | Yes |
| Withstands impersonation attack | Yes | No | Yes | Yes | Yes | No | No | Yes | Yes |
| Withstands MITM attack | Yes | No | Yes | Yes | No | No | No | No | Yes |
| Biometric authentication is ensured at the client's side | Yes | Yes | Yes | Yes | No | Yes | Yes | No | Yes |
| Biometric solution is proposed in a separate smart card | Yes | Yes | Yes | No | Yes | Yes | Yes | No | No |
| Transaction privacy protection from eavesdropper | No | Yes | No | No | No | No | No | No | Yes |
| Transaction privacy protection from PG | No | No | No | No | No | No | No | No | Yes |