A Secure Cloud-based NFC Mobile Payment Protocol

Pardis Pourghomi.[1] Muhammad Qasim Saeed[2]. Gheorghita Ghinea[1]

Abstract Near Field Communication (NFC) is one the most recent technologies in the area of application development and service delivery via mobile phone. NFC enables the mobile phone to act as identification and a credit card for customers. Dynamic relationships of NFC ecosystem players in an NFC transaction process make them partners in a way that sometimes they should share their access permissions on the applications that are running in the service environment. One of the technologies that can be used to ensure secure NFC transactions is cloud computing which offers wide range advantages compare to the use of a Secure Element (SE) as a single entity in an NFC enabled mobile phone. In this paper, we propose a protocol based on the concept of NFC mobile payments. Accordingly, we present an extended version of the NFC cloud Wallet model [14], in which, the Secure Element (SE) in the mobile device is used for customer authentication whereas the customer's banking credentials are stored in a cloud under the control of the Mobile Network Operator (MNO). In this circumstance, MNO plays the role of network carrier which is responsible for controlling all the credentials transferred to the end user. The proposed protocol eliminates the requirement of a shared secret between the Point Of Sale (POS) and the MNO before execution of the protocol, a mandatory requirement in the earlier version of this protocol [16]. This makes it more practicable and user friendly. At the end, we provide a detailed analysis of the protocol where we discuss multiple attack scenarios.

Keywords: Near Field Communication . Security . Mobile transaction . Cloud.

1 Introduction

Technical standards and fundamental interoperability are essential to be achieved for industries working with NFC technology in order to establish a positive cooperation in the service environment. Indeed, lack of interoperability in the complex application level of the service environment [1] has resulted in the slow adoption

[1] P. Pourghomi. G. Ghinea
Department of Computer Science, Brunel University
Uxbridge, Middlesex, UK
Email: pardispourghomii@gmail.com
G. Ghinea
Email: George.ghinea@brunel.ac.uk
[2] M.Q. Saeed
Information Security Group (ISG)
Royal Holloway University of London
Egham, UK
Email: Muhammad.saeed2010@rhul.ac.uk

of NFC technology within societies. Moreover, the current service applications do not provide a unique solution for the ecosystem, therefore the service environment does not meet the right conditions [8].

The current situation is that many independent business players are making decisions based on their own benefits which may not be acceptable by other business players. Reorganizing and describing what is required for the success of this technology have motivated us to extend the current NFC ecosystem models to accelerate the development of this business area.

Our goal is to provide a concept for an NFC ecosystem that is technically feasible, is accepted by all parties involved and thus provides a business case for each player in this ecosystem. Our proposed work is based on the conjecture that the MNO is a key player in the NFC ecosystem. The main advantage of the MNO over other parties is that it owns an SE (SIM card) that fulfils about all the security parameters. Unlike other forms of SEs, the SIM card can be easily managed by the MNO Over-the-Air (OTA). Thus we foresee that the MNO will play a major role in future in the NFC ecosystem.

1.1 Our contribution

We extend the earlier proposed mobile transaction mechanisms mentioned in [14, 16, 5]. The key contribution of our work is the elimination of the requirement of shared secret between the shop and the MNO, a prerequisite in the initially proposed protocols. This makes our work more practicable as the shop does not need to get itself registered with the MNO to perform mobile transaction.

We partitioned the SE into two sections: one stored in the SIM for authentication of a customer and the other stored in the cloud to store the credit/debit card details of the customer. This helps in managing multiple cards against a single customer. The authentication of the customer to the MNO is based on GSM authenticating mechanism with improved security features. The customer selects one of the already registered accounts to be used for transaction. Our protocol works on a similar pattern to 'PayPal': the MNO acts as the PayPal and a user registers multiple banking cards for monetary transactions with the MNO. The user then selects a single card for monetary transactions at the time of the payment.

This paper is organized as follows: Section 2 includes an introduction to SEs and a brief consideration of their functionalities. Also, a discussion is provided regarding management issues in SEs and advantages of having cloud environment for mobile payment transactions are highlighted. Section 3 describes the related work which has been carried out in this area. Subsequently, section 4 discusses GSM authentication which is used in our extended model. Section 5 then introduces our proposed transaction protocol in detail. Section 6 provides the analysis of our proposed protocol from multiple security aspects. This analysis encompasses the authentication and security of the messages among customer, shop POS terminal and the MNO. Finally, Section 7 presents our conclusion.

2 SE management

The security of NFC is supposed to be provided by a component called security controller, in which takes the form of a SE. The SE is an attack resistant microcontroller more or less like a chip that can be found in a smart card [15]. The SE provides storage within the mobile phone and it contains hardware, software, protocols and interfaces. The SE provides a secure area for the protection of the payment assets (e.g. keys, payment application code, and payment data) and the execution of other applications. In addition, the SE can be used to store other applications which require security mechanisms and it can also be involved in authentication processes.

To be able to handle all these, the installed operating system has to have the capability of personalizing and managing multiple applications that are provided by multiple Service Providers (SPs) preferably OTA. Still, the ownership and control of the SE within the NFC ecosystem may result in a commercial and strategic advantage

but some solutions are already in place [15] and researchers are developing new models to overcome the complexity of interactions among ecosystem's stakeholders.

2.1 Advantages of cloud-based approach

The NFC cloud-based approach introduces a new method of storing, managing and accessing sensitive transaction data by storing data in the cloud rather than the mobile phone [20]. When a transaction is carried out, the required data is pulled out from a remote virtual SE which is stored within the cloud environment and pushed into the mobile phone's SE in an encrypted format. The mobile phone's SE provides temporary storage and authentication assets for the transaction to take place. After reaching the SE in an NFC phone, data are again pulled out from the handset and reach the vendor's terminal. In general, the communication between the cloud provider and the vendor's terminal is established through the NFC phone.

The storage capacity of the SE should be large enough in order to store user applications with unknown sizes. As the user may wish to add more applications to his NFC phone, this issue brings a limitation for existing solution as each SE supports certain storage capacity.

The other issue with the SE is that companies have to meet the requirements of organisations such as EMVco [13] to provide high level security in order to store a card's data. This approach makes the SE expensive for the companies, while the cloud-based approach reduces this cost. In the NFC cloud-based approach, the SE which is stored in the NFC phone can only be responsible for user/device authentication and not for storing data. This solution increases the cost efficiency compared to the current costs that SE makes for a company. Also, the NFC controller chips will be smaller and cheaper as they would not have to support all functionalities.

The NFC cloud-based approach also makes the business simpler for companies in terms of the integration of SE card provisioning. It would be much easier for businesses to implement NFC services without having to perform card provisioning for every single SE. An NFC phone user will be able to access an unlimited number of applications as they are stored within a cloud secure server and not in a physical SE. In terms of flexibility, all users would be able to access all their applications from all their devices (e.g. phones, tablets or laptops) since the applications are stored in a cloud environment that provides a secure storage space. Moreover, fraud detection would be instant as the system fully runs in an online mode.

3 Related works

A. Google Wallet

One of the major companies which operate the concept of Mobile Wallet is Google. They named this service as "Google Wallet" [7, 18]. The communication between the mobile phone and the POS is carried out through NFC technology that transmits the payment details to merchant's POS. Customer credentials are not stored in the mobile phone; rather, they are stored online. Google Wallet takes the form of an application stored on the customer's mobile phone. The customer will have an account with Google Wallet which includes the relevant registered credit/debit cards. Accordingly, the Google Wallet device has a chip /SE which stores encrypted payment card information. Linked credit or debit card credentials are not stored on the SE; rather, the virtual prepaid credit/debit card which is created during the setup is stored on the SE. The transaction then operates through the virtual prepaid credit/debit card that transfers funds from Google Wallet into the merchant's POS when customer taps his phone on POS.

B. MasterPass

"MasterPass" [10, 3] is a service which has been developed by MasterCard as an extended version of PayPass Wallet Services [12] and provides digital wallet service for secure and convenient online shopping. In MasterPass, delivery information and transaction data are stored in a central and secure location. The latest MasterPass provides the following services [12]:

- *MasterPass checkout services:* This service enables the vendor's payment acceptance in a consistent way irrespective of the client's location. This means vendors have the ability to accept a payment without having to know where the client is. For instance, when the client is in store, he can use this service since it supports NFC, QR codes, tags, and mobile devices to pay for products at a vendor's POS. Thus, in online shopping scenarios, the client can use this service to pay for a product without having to enter the card and delivery details every time he intends to make a purchase.

- *MasterPass-connected wallets:* Vendors, financial institutions, and partners are able to provide their own wallets using this service. The client's card information, address books, etc. can be saved in a secure cloud provided by a party they trust. Thus, clients can use other credit and debit cards in addition to their Mastercard cards.

- *MasterPass value added services:* the purpose of this service is to improve the client's shopping experience before, during and after checkout. Value added services include account balances, offers, loyalty programs, and real-time alerts.

3.1 Our approach

The general overview of the cloud-based NFC payments is described in [14] where the NFC Cloud Wallet model is also proposed. We then proposed an extension to the previously proposed NFC Cloud Wallet model and designed an NFC payment protocol which was based on a GSM network [16]. This protocol was the improved version of Chen's protocol [5] where user interaction with the system was improved, making it more user friendly.

An additional layer of security was added by introducing PIN authentication by the user [4, 17]. Mutual authentication was improved by adding freshness by the mobile device in order to resist replay attack. We also added digital signatures with the transaction messages for data integrity and non-repudiation [16, 9].

Since there were multiple options applicable to this model, we designed our protocol based on the following assumptions:

- The SE is part of SIM
- The cloud is part of the MNO
- The MNO manages the SE/SIM
- Banks, etc. are linked to the MNO

The key issue in this payment model was the connection between POS and MNO which makes it different from the protocol that we have designed in this paper. In this paper, we designed our protocol based on the following assumptions:

- The SE is part of the SIM
- The cloud is part of the MNO
- The MNO manages the SE/SIM
- *Financial institutions are linked to the MNO*
- *The POS has no connection with the MNO*
- *The communication is carried over a single channel: MNO, mobile device and POS*

4 GSM authentication

When a mobile device signs into a network, the MNO first authenticates the device (specifically the SIM). The authentication stage verifies the identity and validity of the SIM and ensures that the subscriber has authorized access to the network. The Authentication Centre (AuC) of the MNO is responsible for authenticating each SIM that attempts to connect to the GSM core network through the Mobile Switching Centre (MSC). The AuC stores

two encryption algorithms A3 and A8, as well as a list of all subscribers' identity along with corresponding secret key K_i. This key is also stored in the SIM. The AuC first generates a random number known as R. This R is used to generate two responses, signed response S and key K_c as shown in figure 1, where $S = E_{Ki}(R)$ using A3 algorithm and $K_c = E_{Ki}(R)$ using A8 algorithm [6].

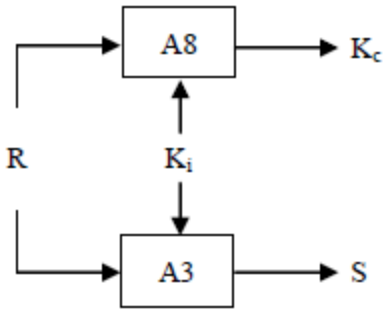

Fig. 1 Generation of K_c and S from R

The triplet (R, S, K_c) is known as Authentication triplet generated by the AuC. The AuC sends this triplet to MSC. On receiving a triplet from the AuC, the MSC sends R (first part of the triplet) to the mobile device. The SIM of the mobile device computes the response S from R, as K_i is already stored in the SIM. Mobile device transmits S to MSC. If this S matches the S in the triplet (which it should in case of a valid SIM) then the mobile is authenticated. K_c is used for communication encryption between the mobile station and the MNO. Table 1 describes the abbreviations used in the proposed protocol.

Table1 Abbreviations

AuC	Authentication Centre (subsystem of MNO)
App_{ID}	Approval ID. Generated after credit approval
Acc_{ID}	Account ID of the customer
C_{r_req}	Credit Request Message
C_{r_app}	Credit Approved Message
IMSI	Internet Mobile Subscriber Identity
K_i	SIM specific key. Stored at a secure location in SIM and at AuC
K_c	E_{ki} (R) using A8 algorithm
K_1	Encryption key generated by shop
K_2	MAC key generated by shop
K_{pub}	Public key of MNO
K_{pr}	Private key of MNO
K_{sig}	Signing key of MNO
K_{ver}	Verification key of MNO
LAI	Local Area Identifier
MNO	Mobile Network Operator
R	Random Number (128 bits) generated by MNO
R_s	Random number generated by SIM (128 bits)
SE	Secure Element
TM_m	Transaction Message for mobile
TM_s	Transaction Message for shop
TMSI	Temporary Mobile Subscriber Identity
TP	Total Price
T_{SID}	Temporary Shop ID
TS_s	Shop Time Stamp
TS_t	Transaction Time Stamp

5 Proposed protocol

The proposed protocol is based on a cloud architecture, in which the cloud is managed by the MNO. The SE used in this protocol is divided into two sections: one, being a part of SIM, is used for authentication of a customer, whereas the other section, being a part of cloud, is used to store sensitive banking information of the customer. The customer has registered his credit/debit card details with the respective MNO. Since our protocol supports multiple accounts against a single customer, a customer can register more than one credit/debit card with the MNO. Each account of a customer is identified by a unique account ID, Acc_{ID}. The Acc_{ID} is intimated to a customer when he registers his debit/credit card with the MNO. MNO stores these details in a cloud. The mobile device has a valid SIM and is connected to respective MNO through GSM network. The communication over the GSM network is encrypted as specified in GSM standard. The mobile device is connected to the shop terminal over an NFC link. The NFC link is not secure and can be eavesdropped.

Although the shop has no link with the MNO, the shop trusts the MNO. A message digitally signed by the MNO is considered authentic and its contents are trusted by the shop. When dealing with the signed data, one has to distinguish between data authenticity and trust in the message contents. An authentic data may not be true [20]. For example, a valid signature with the message 'Sun revolves around the earth' will prove the message as authentic but its contents are not true. We assume that the messages signed by the MNO are not only authentic but the contents are also considered trustworthy by the shop. For simplicity, we refer to the mobile device and SIM as a single unit 'mobile device'. K_{sign}, K_{ver} are the signing and verification keys respectively of MNO. K_{pr}, K_{pub} are the private and public keys respectively of the MNO. The proposed protocol executes in three different phases as shown in figure 2:

A. Phase 1: Authentication

Step 1: The mobile device sends *TMSI*, *LAI* as its ID to the shop terminal. The shop terminal determines the user's mobile network from this information. The network code is available in *LAI* in the form of Mobile Country Code (*MCC*) and Mobile Network Code (*MNC*). An *MNC* is used in combination with *MCC* (also known as a '*MCC/MNC tuple*') to uniquely identify a mobile phone operator/carrier [19].

Step 2: Shop terminal sends a message to the mobile device containing Total Price (*TP*), a temporary shop ID (T_{SID}), and Time Stamp (TS_s) of current time. The T_{SID} acts as one time ID of the shop and gets updated after each transaction.

Step 3: The mobile device initiates a mutual authentication protocol with the MNO. It sends TMSI, LAI as its identifier. The MNO identifies its customer and generates an authentication triplet (R, S, K_c).

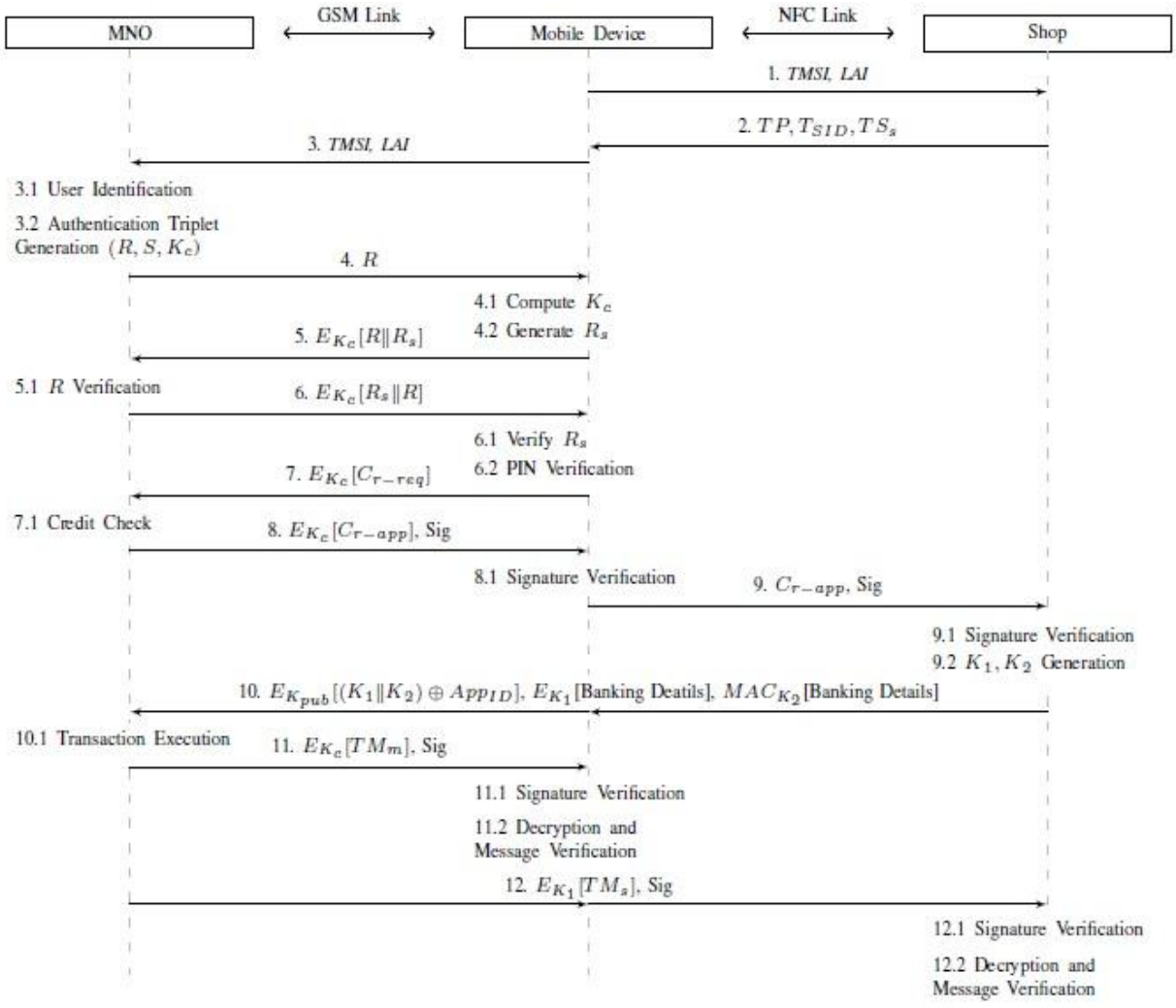

Fig. 2 The proposed protocol

Steps 4-5: The MNO sends *R*, a part of the authentication triplet, to the mobile device. The mobile device computes K_c from *R* as explained in Section 4. The mobile device generates a random number R_s and concatenates with *R*, encrypts with key K_c and sends it to the MNO. The MNO decrypts the message using K_c, the key it already has in the authentication triplet. The MNO compares *R* in the authentication triplet with the *R* in the response. If both *Rs* are same, then the mobile is authenticated for a valid SIM.

Step 6: After successful SIM (or mobile device) authentication, the MNO swaps *R* and R_s, encrypts with K_c and sends it to mobile device. This step authenticates the MNO to the mobile device. The mobile device receives the response E_{Kc} $(R_s||R)$ and decrypts it with the key K_c already computed in Step 4.1. The mobile device compares both *R* and R_s. If both are same, then the MNO is authenticated. After successful authentication, the user is asked by the mobile device to enter the PIN. The PIN is stored in the SIM at a secure location. The SIM compares both PINs and if both are same, the user is authenticated as the legitimate user of the mobile device.

B. Phase 2: Financial Approval

Step 7: After successful authentication, the customer selects the account Acc_{ID} for payment. The mobile device forms a credit request message $C_{r\text{-}req}$ for credit approval from the MNO as:

$$C_{r\text{-}req} = TP||T_{SID}||TS_s||TMSI||Acc_{ID}$$

The mobile device encrypts $C_{r\text{-}req}$ with the key K_c (the encryption key used in GSM communication) and sends it to the MNO. The MNO receives the message, decrypts and communicates with the cloud for a credit check against the account ID Acc_{ID} of the customer.

Step 8: Once the credit is approved from the financial entities through cloud, an approval ID (App_{ID}) is generated by the approving authority. App_{ID} acts as in index to a table storing information about the amount to be credited, destination Shop ID, the time stamp and the customer ID (TMSI). This helps in resolving any disputes in future. The MNO forms a new string $C_{r\text{-}app}$ indicating credit approval as:

$$C_{r\text{-}app} = TP||T_{SID}||TS_s||TMSI||App_{ID}$$

The MNO encrypts the string $C_{r\text{-}app}$ with the key K_c and computes signature with the signing key K_{sign} over the plaintext. The encrypted $C_{r\text{-}app}$ along with its signature is transmitted to the mobile device.

The mobile device decrypts the message to get $C_{r\text{-}app}$. It compares the contents of $C_{r\text{-}app}$ with the contents of $C_{r\text{-}req}$, as the only difference between both messages is that the Acc_{ID} is the former is replaced by the App_{ID} in the latter. It provides an assurance the $C_{r\text{-}app}$ is generated by a legitimate authority. Mobile device, then, verifies the signature as the signature was computed over the plaintext. The signature provides data integrity, data origin authentication and non-repudiation of the $C_{r\text{-}app}$ message. After successful verification, the mobile device forwards $C_{r\text{-}app}$ to the shop along with the corresponding signature.

Step 9: The shop terminal verifies the signature by the verification key K_{ver} to detect any alteration. In case of an invalid signature, the shop discards the message. A valid signature provides data integrity and data origin authentication. In this case, the shop believes that the message is authentic and the MNO has agreed to pay for the customer. This is like a three party contract where a middle party, trusted by both other parties, provides an assurance that the other party is willing to pay the price.

Phase 3: Transaction Execution

Step 10: After successful authentication and message contents verification, the shop generates two keys K_1 and K_2 for data encryption and MAC calculation respectively. It forms a string $(K1||K2) \oplus App_{ID}$ and encrypts it with the public key, K_{pub}, of the MNO. The shop encrypts its banking details with the key K_1 and computes its MAC with the key K_2. The banking details may include bank account title, account number, bank code, branch code etc. The MNO needs banking details in order to transfer amount from the customer account to the shop account. This detail is transmitted to the MNO through the mobile device but the latter cannot decrypt this information. This forms a virtual tunnel between the shop and the MNO through the mobile device.

Once the MNO receives this message, it decrypts first part to extract the K_1 and K_2. The role of App_{ID} in this step is to bridge the authentication phase to the transaction execution phase. The MNO checks the validity of the MAC and if successful, it decrypts the banking details. It forwards the banking details to the cloud for the monetary transaction.

Steps 11-12: After a successful transaction, the MNO generates a transaction number TSN and corresponding time stamp TS_t and forms Transaction Message for mobile device TM_m and Transaction Message for shop TM_s as:

$$TM_m = TSN||TP||T_{SID}||TMSI||TS_t$$
$$TM_s = TM_m || \text{ [Banking Details]}$$

The MNO encrypts TM_m with the key K_c and computes the signature over the ciphertext. It sends encrypted TM_m and the corresponding signature to the mobile device. The mobile device first verifies the signature. In case of

an invalid signature, the mobile device discards the message without decrypting it. Otherwise, it decrypts the message and verifies the contents.

The MNO forms the Transaction Message for the shop TM_s by appending Shop Banking Details to the earlier formed TM_m. It encrypts TM_s with the key K_1 and computes signature over the ciphertext. The MNO sends the encrypted message along with its signature to the mobile device to further relay it to the shop. The mobile device can neither decrypt this message as it does not possess K_1, nor alter any contents as they are protected by the signature. The shop verifies the signature and if invalid, discards the message without decrypting the message. Otherwise, the shop decrypts the message and verifies its contents. The contents consist of important transaction information exchanged during the transaction. If the shop wants any clarification, it can approach the MNO quoting the Transaction Number *TSN* and Approval ID App_{ID} received in step 9.

6 Protocol analysis

In this section, we analyse this protocol from multiple perspectives. This analysis encompasses the authentication and security of the messages. We assume that the MNO is trust worthy, whereas the customer or the shop can be dishonest. We analyse multiple attack scenarios to ascertain the strength of our protocol.

A. Dishonest Customer

Scenario 1: A dishonest customer plans to buy some products with payment from someone else account. So, he sends a fake but valid ID (for example *TMSI, LAI* of a mobile of a target customer) in step 1 to shop. Shop replies with step 2 providing information about the total price, its temporary ID and the time stamp. In step 3, the dishonest customer has two options in the authentication phase. Either he communicates with his legitimate MNO for authentication or with the target customer's MNO. In the former case, the amount will be deducted from his account (which is what he is not willing to do) whereas, the amount will be deducted from the target customer's account in the latter case. If he goes for the latter option, however, he fails the authentication process in step 5 as he lacks the legitimate K_c. Thus, someone else's ID cannot be successfully used in this protocol.

Scenario 2: A dishonest customer plans buy goods without any payment. So, he provides his own banking details, rather than the shop banking details, to the MNO in step 10. If case of a successful transaction, the MNO deducts amount from the customer account and pays back in the customer amount (both accounts may be different to avoid detection). The transaction receipt is then transmitted to the shop as a proof of payment. To accomplish this attack, the dishonest customer blocks step 10, in which the shop banking details are transmitted to the MNO through the mobile device. The customer cannot alter this message as it is encrypted with keys K_1 and K_2. Both these keys are encrypted with the public key K_{pub} of the MNO, so no other than the MNO can get these keys. Therefore, rather than altering this information, the dishonest customer discards this message and designs his own message as:

$$Ek_{pub}\,[(K'_1 \,||\, K'_2) \oplus App_{ID}],\; E_{K'1}\,[Banking\ Details],$$
$$MAC_{K'2}\,[Banking\ Details]$$

Where the banking details are customer's banking details rather than the shop's, the MNO has to rely on the information provided by the mobile device as the former does not share any secret with the shop prior to the execution of the protocol.

The MNO performs transaction against the information provided by the mobile device. After the transaction execution, the MNO sends 'receipts' in messages 11 and 12. The mobile device blocks message 12 as this message contains the information of the bank that was used during the transaction.

Since the customer's banking details were used during transaction, the dishonest customer needs to replace the banking details in this massage with the shop banking details. The customer can decrypt message in step 12 as it

is now encrypted with the customer's malicious key $K'_{1.}$ He needs to change the banking details and encrypt with the shop generated key K_1 in step 9.2. Since the customer lacks this key, he cannot generate a valid ciphertext. Moreover, the original message is protected by the digital signature. If the customer makes any alteration to change the banking details, it will void the signature. If the customer does not alter the message to maintain the validity of the signature, the shop can verify the signature but cannot decrypt the message (as it is encrypted with the customer's malicious key K'_1). In both cases, the shop cannot verify the transaction and a failure message is sent at the end. Hence, a dishonest customer is again unsuccessful.

There may be another approach to accomplish the above attack where the dishonest customer plans to buy some goods without payment. The dishonest customer does not communicate with the MNO since it is not successful as described above; rather the customer impersonates the MNO to the shop in this scenario. The target of the customer is to send fake but acceptable receipts to the shop at the end of the protocol by replaying old legitimates messages or fabricating new messages. Since the customer is not communicating with the MNO, his account cannot be debited. In the original protocol, the shop receives three messages from the mobile device, message 1, 9 and 12. Message 1 is originated by the mobile device, whereas message 9 and 12 are actually originated by the MNO but are relayed by the mobile device to the shop. A dishonest customer needs to design or replay the latter two messages in such a way that they are acceptable to the shop. Both messages are digitally signed by the MNO. These messages contain a Temporary Shop ID (T_{SID}) and a Time Stamp (TS_s). T_{SID} is a random value generated by the shop every time in the start of the protocol. This value does not only serve as a shop ID during protocol, but also it adds freshness to the protocol messages. TS_s is updated too in every protocol round, but it may be predictable to some extent. A combination of these two values, along with the digital signatures of the MNO, does not allow either replay or alteration of the messages. Hence the dishonest is again unsuccessful.

Scenario 3: A dishonest customer plans to pay less than the required amount but intimates to shop of full payment. To accomplish this attack, the mobile device sends TP´ in Credit Request message, $C_{r\text{-}req}$, in step 7 to MNO, where $TP' < TP$. The mobile device receives Credit Approve message, $C_{r\text{-}app}$, in step 8 from the MNO confirming that the initially requested amount TP' has been approved for transaction. However, the mobile device needs to intimate the shop in step 9 that the original amount, TP, is approved for transaction. Since the approved price is digitally signed, it cannot be amended by the mobile device. So the actual price that is approved by the MNO is transmitted to the shop. Hence, this attack fails on proposed protocol.

B. Dishonest Shop

Scenario 4: The shop is dishonest and plans to draw more than the required amount without intimation to the customer. The information about the amount to be transferred is intimated to the MNO by the mobile device in Credit Request message, $C_{r\text{-}req}$, in step 8. A mobile device cannot send more than the required price unless the device itself is compromised. Therefore, a shop cannot get more than the required amount in this protocol.

Scenario 5: The shop is dishonest and repudiates the receipt of transaction execution message in step 12. In this way, the shop does not deliver goods despite receiving the required amount. In such scenario, the mobile device has the signed receipt from the MNO indicating a Transaction Serial Number *TSN* in step 11. The *TSN* is linked to the Approval ID App_{ID} generated in step 8. Since both the values are digitally signed by the MNO, the mobile device can approach MNO regarding any dispute.

C. Messages Security

Apart from the above-mentioned scenarios, we also analysed our protocols from various other angles. The data over the GSM network is encrypted according to GSM specification. The key K_c used for the data encryption is fresh in each round of transaction. The data over NFC link in Authentication and Approval phase (Step 1, 2 and 9) is sent in clear. This data does not contain any sensitive information. Total Price may be considered sensitive information but it is also displayed on the shop terminal for visual information of the customer. The read range

of the displayed price is much more than the range of the NFC link. Therefore, we graded TP as not so sensitive information to be protected over NFC link. However, once the TP is transmitted over GSM network, it is encrypted with the key K_c.

Information that is sent in clear over the NFC link is the Credit Approval ID (App_{ID}) in the ($C_{r\text{-}app}$) message (step 9). The App_{ID} is a random string generated by the credit approval authority. From an attacker's perspective, its only significance is its assurance that the customer has, at least, TP amount in his account. This assurance can also be achieved if a customer successfully pays for some goods. Therefore, App_{ID} is also not sensitive information in this scenario.

Role of Approval ID in Message 10: App_{ID} acts as a bridge between the Financial Approval phase and the Transaction phase. It adds freshness to message 10, so it cannot be replayed in the future. App_{ID} is XORed to avoid increase in the message length. Any alternation in the first part of the message 10 (Ek_{pub} $[(K_1||K_2) \oplus App_{ID}]$) results in invalid keys K'_1 and K'_2. This invalidates the MAC and hence detection.

Non-repudiation of Transaction Messages: Transaction Execution messages (Step 11, 12) are digitally signed by the MNO. In case of any dispute about payment, the MNO has to honour both messages. So both the customer and the shop are completely secured about the transaction.

Disclosure of Relevant Information: Shop banking details represent sensitive information as they contains the bank account number etc. It is encrypted not only on the GSM link but also on the NFC link. This information is transmitted after the credit approval information is received by the shop. The banking detail is transmitted through the mobile device to the MNO, yet the former cannot decrypt this information. Since the mobile device does not need this information, it is not disclosed to the mobile device. Similarly, the account information of the customer is not communicated to the shop in $C_{r\text{-}app}$ message.

New set of Keys for every transaction: The encryption key over GSM network, K_c, is generated from R. Since R is changed in each round of transaction protocol, the K_c is also fresh. The encryption keys K_1 and K_2 are generated by the shop in each round. So both these keys are also fresh.

Encryption and MAC Keys: Separate keys are used for encryption and MAC calculation making the protocol more secure. Encrypt-then-MAC is an approach where the ciphertext is generated by encrypting the plaintext and then appending a MAC of the encrypted plaintext. This approach is cryptographically more secure than other approaches [2]. Apart from cryptographic advantage, the MAC can be verified without performing decryption. So, if the MAC is invalid for a message, the message is discarded without decryption. This results in computational efficiency.

7 Conclusion

In this paper we have proposed a transaction protocol that provides a secure and trusted communication channel to the communication parties. The proposed protocol was based on the NFC Cloud Wallet model [14], NFC payment application [16] and W. Chen et al [5] for secure cloud-based NFC transactions.

We considered a cloud-based approach for managing sensitive data to ensure the security of NFC transactions over the use of a SE within the cloud environment as well as considering the role of SE within the NFC phone architecture. The operations performed by the vendor's reader, an NFC enabled phone and the cloud provider (in this paper MNO) are provided and such operations are possible by the current state of the technology as most of these measures are already implemented to support other mechanisms.

We considered the detailed execution of the protocol and we showed our protocol performs reliably in cloud-based NFC transaction architecture. The main advantage of this paper is to demonstrate another way of payment for all those people who do not have bank accounts. This way of making payments eases the process of

purchasing for ordinary people as they only have to top up with their MNO without having to follow all the banking procedures.

As a part of future work, a proof of concept prototype can be implemented in order to determine the reliability of the proposed protocol in terms of number of factors such as timing issues. This implementation refers to the performance domain of the proposed protocol which can be taken into the account to consider the performance of the protocol rather than its security, which is discussed in this paper. The idea of the proposed protocol can also be extended to a multi-party protocol. Furthermore, other possible architectures in this area should be explored and defined in order to finalize the most reliable architecture for cloud-based NFC payment applications.

References

[1] Antoniou, G., and Batten, L. (2011). E-commerce: protecting purchaser privacy to enforce trust. *Journal of Electronic Commerce Research*, vol. 11, issue. 4, pp. 421 – 456.

[2] Bellare, M., and Namprempre, C. (2000). Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *proceedings of the International Conference on Advances in Cryptology* (pp. 531–545).

[3] Bodhani, A. (2013). New ways to pay [Communications Near Field]. *Engineering & Technology*, vol.8, no.7, pp.32 – 35.

[4] Buccafurri, F., and Lax, G. (2011). Implementing disposable credit card numbers by mobile phones. *Journal of Electronic Commerce Research*, vol. 11, issue. 3, pp. 271 – 296.

[5] Chen, W., Hancke, G., Mayes, K., Lien, Y., and Chiu, J.H. (2010). NFC mobile transactions and authentication based on GSM network. *In proceedings of the International Workshop on Near Field Communication*. IEEE Computer Society, Monaco, France (pp. 83–89).

[6] ETSI Specification of the Subscriber Identity Module (1996). Mobile Equipment (SIM - ME) interface (GSM 11.11), *European Telecommunications Standards Institute (ETSI Std. Version 5.3.0)* http://www.etsi.org/deliver/etsi_gts/11/1111/05.03.00_60/gsmts_1111v050300p.pdf. Accessed 8 January 2012.

[7] Google (2013). Goole Wallet. http://www.google.co.uk/wallet/faq.html. Accessed 3 April 2013.

[8] Kauffman, R. J., Liu, J., and Ma, D. (2013). Technology investment decision-making under uncertainty: the case of mobile payment systems. In *proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, 4-7 January.

[9] Mascha, M. F., Miller, C. L., and Janvrin, D. J. (2011). The effect of encryption on Internet purchase intent in multiple vendor and product risk settings. *Journal of Electronic Commerce Research*, vol. 11, issue. 4, pp. 401 – 419.

[10] MasterCard (2013). MasterPass. https://masterpass.com/online/Wallet/Help?cid=127568. Accessed 7 April 2013.

[11] NFC World (2012). MasterCard enters the mobile wallet market. http://www.nfcworld.com/2012/05/09/315600/mastercard-enters-the-mobile-wallet-market/. Accessed 1 June 2012.

[12] NFC World (2013). MasterCard unveils MasterPass digital wallet and mobile payments platform. http://www.nfcworld.com/2013/02/25/322610/mastercard-unveils-masterpass-digital-wallet-and-mobile-payments-platform/. Accessed 3 March 2013.

[13] Pailles, J., Gaber, C., Alimi, V., and Pasquet, M. (2010). Payment and privacy: A key for the development of NFC mobile. *In proceedings of the Collaborative Technologies and Systems International Symposium*, Chicago, Illinois, USA (pp. 378 – 385).

[14] Pourghomi. P., and Ghinea, G. (2012). Managing NFC payments applications through cloud computing. In *proceedings of the 7th International Conference for Internet Technology and Secured Transactions*, IEEE, London, UK (pp. 772 –777).

[15] Pourghomi, P., and Ghinea, G. (2012). Challenges of managing secure elements within the NFC ecosystem. In *proceedings of the 7th International Conference for Internet Technology and Secured Transactions*, IEEE, London, UK, pp. 720 –725.

[16] Pourghomi, P., Saeed, M., and Ghinea, G. (2013). A proposed NFC payment application. *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 8, pp. 173 – 181.

[17] Ren, Y., Cheng, F., Peng, Z., Huang, X., and Song, W. (2011). Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks. *Journal of Electronic Commerce Research*, vol. 11, issue. 1, pp. 103 – 121.

[18] Roland, M., Langer, J., and Scharinger, J. Applying relay attacks to Google Wallet. In *proceedings of the 5th International Workshop on Near Field Communication (NFC)*, Zurich, Switzerland, Feb. 2013.

[19] Technical specification group core network. (2012) Numbering, addressing and identification, 3rd Generation Partnership Project (3GPP Std. Version 3.18.0) http://www.arib.or.jp/english/html/overview/doc/STD-T63v10_00/5_Appendix/R99/21/21101-3i0.pdf. Accessed 8 January 2012.

[20] Urien, P., Piramuthu, S. (2013). Towards a secure cloud of Secure Elements concepts and experiments with NFC mobiles. In *proceedings of International Conference on Collaboration Technologies and Systems*. San Diego, California, USA (pp.166 – 173).

[21] Vincent, O. R., Folorunso, O., and Akinde, A.D. (2010). Improving e-payment security using Elliptic Curve Cryptosystem. *Journal of Electronic Commerce Research*, vol. 10, issue. 1, pp. 27, 41.