Mobile Transaction over NFC and GSM

Muhammad Qasim Saeed*, Pardis Pourghomi†,

*Information Security Group (ISG)
Royal Holloway University of London, Egham, UK
Email: muhammad.saeed.2010@live.rhul.ac.uk

†School of Information Systems, Computing and Mathematics
Brunel University, Uxbridge, Middlesex, UK
Email: pardis.pourghomi@brunel.ac.uk

***Abstract*—Although NFC mobile services have great potential for growth, they have raised a number of issues which are of concern to researchers and are preventing the wide adoption of this technology within society. Dynamic relationships of NFC ecosystem players in an NFC transaction process make them partners in a way that sometimes requires that they share access permission to applications that are running in the service environment. One of the technologies that can be used to ensure secure NFC transactions is cloud computing. This offers a wider range of advantages than the use of a Secure Element (SE) as a single entity in an NFC enabled mobile phone. In this paper, we propose a protocol for NFC mobile payments, namely an extended version of the NFC cloud Wallet model [1]. In our protocol, the 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). The proposed protocol eliminates the requirement for 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[2]. This makes it more practicable and user friendly. A detailed analysis of the protocol discusses multiple attack scenarios.**

***Index Terms*—Near Field Communication; Security; Mobile transaction; Cloud.**

I. Introduction

Agreed technical standards and fundamental interoperability are essential basics to achieve for industries working with NFC technology in order to establish positive cooperation in the service environment. Lack of interoperability in the complex application level has resulted in the slow adoption of NFC technology within society. Current service applications do not provide a unique solution for the ecosystem: many independent business players are currently making decisions based too closely on their own advantage which may not be acceptable to other players. Consequently, the service environment does not meet the optimal conditions for take-up. Reorganizing and describing what is required for the success of this technology has motivated us to extend 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 an improved business case for each of the players in this ecosystem. One of the main players in the NFC ecosystem is the Mobile Network Operator (MNO). The main advantage an MNO has over other parties is that it owns a Secure Element (SE, a SIM card) that stores and protects the security parameters. Unlike other forms of Secure Element (SE), 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. Our proposed work is based on the conjecture that the MNO is a key player in the NFC ecosystem.

A. Our Contribution

Here we extend the earlier proposed mobile transaction mechanisms mentioned in [1], [2] and [3]. The major contribution of our work is the elimination of the requirement for a shared secret between the point of sale (PoS) and the MNO, a prerequisite in the initially proposed protocols. This makes our work more practicable as a shop does not need to have itself registered with the MNO to perform mobile transaction. We partition the SE into two sections: one stored in the SIM for authentication of a customer and the other stored in the cloud to hold credit/debit card details of the customer. This helps in managing multiple cards for a single customer. The authentication of the customer by the MNO is based on a GSM authenticating mechanism with improved security features. Our protocol works on a similar pattern to that of 'PayPal': the MNO, acting in the same way as PayPal, registers multiple banking cards against a user for monetary transactions. The user, then, selects a single card at the time of the payment. An overview of this model was proposed in [4].

This rest of this paper is organized as follows:

- Section II includes an introduction to our Secure Element (SE) and a brief consideration of its functionalities. Also, a discussion is provided regarding management issues in the SE, and some advantages of having a cloud environment for mobile payment transactions are highlighted.
- Section III describes related work which has been carried out in this area.
- Section IV provides an introduction to GSM authentication because our model needs to use a more secure

version of GSM authentication.

- Section V introduces our proposed transaction protocol in detail.
- Section VI provides the analysis of our proposed protocol from multiple security view points. This analysis encompasses the authentication and security of the messages between the customer, the shop's POS terminal and the MNO.
- Finally, Section VII places our solution in context, summarises how it operates, and draws some conclusions.

II. Management Of SE

The security of NFC is supposed to be provided by a component called the "security controller" that is in the form of an SE. The SE is an attack resistant microcontroller, more or less like a chip, that can be found in a smart card [5]. The SE provides storage within the mobile phone and it contains hardware, software, protocols and interfaces. It provides a secure area for the protection of 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 over-the-air. Still, the ownership and control of the SE within the NFC ecosystem may result in a commercial and strategic advantage. However, some solutions are already in place [5] and researchers are developing further models to overcome this problem.

A. Advantages of Cloud-Based Approach

Our NFC cloud-based approach introduces a new method of storing, managing and accessing sensitive transaction data by storing data in the cloud rather than in the mobile phone. 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 SE in an encrypted form. The mobile phone SE provides temporary storage and authentication assets for the transaction to take place. After reaching the SE in an NFC phone, the data are pulled out from the handset and sent to the vendor terminal. In general, the communication between the cloud provider and the vendor terminal is established through the NFC phone.

Ideally, the storage capacity of the SE should be large enough to store a number of user applications with unknown size. As the user may wish to add more applications to his NFC phone, space is a limitation as each SE supports only a certain storage capacity. Another issue with SEs is that companies have to meet the requirements of organisations such as EMVco [6] to provide high level security in order to store card's data. This makes the SE expensive for companies. On the other hand, a cloud-based approach would reduce this cost. In an NFC cloud-based approach, the SE in the NFC phone need only be responsible for user/device authentication and not for storing data. This increases the cost efficiency of the SE compared with present, enabling many more secure applications to be supported. Also, the NFC controller chips would be smaller and cheaper as they no longer have to support all previous functionality.

The NFC cloud-based approach makes 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. The NFC phone user will be able to access a practically 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 runs only in a fully online mode.

III. Related Work

One of the major companies which operates the concept of a Mobile Wallet is Google, whose name for this service is "Google Wallet" [7]. 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 is in the form of an application that is stored on customer's mobile phone. The customer will have an account with Google Wallet which includes the relevant registered credit/debit cards. The transaction takes place in the form of a virtual prepaid credit (MasterCard) card that is transfered from the Google Wallet into the merchant's POS when customer taps his phone on the PoS. Google Wallet stores credit and debit cards online using secure servers. The Google Wallet device has a chip called the Secure Element that stores encrypted payment card information. Linked credit or debit card credentials are not stored on the SE but rather in the virtual prepaid card, which is created during the setup, and is stored on the SE.

"MasterPass" [8] is a service which has been developed by MasterCard as an extended version of PayPass Wallet Services [9] that provides a digital wallet service for safe and easy online shopping. MasterPass stores all the payment and shipping information in one central, secure location. The new MasterPass service comprises three elements [10]:

MasterPass checkout services. For in-store scenarios either at the register or in the aisle, MasterPass will support the use of NFC, QR codes, tags and mobile devices at the PoS. For online purchases, MasterPass provides shoppers with a simple check-out process by eliminating the need to enter detailed shipping and card information with every purchase.

MasterPass-connected wallets. These enable banks, merchants and partners to offer their own wallets. Consumers can securely store card information, address books and more in a secure cloud, 'hosted by an entity they trust' [[10]]. The wallet is open, which means that in addition to MasterCard cards,

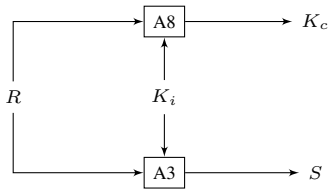

Fig. 1. Generation of K_c and S from R

consumers can use other branded credit, debit and prepaid cards.

MasterPass value added services. These are designed 'to enrich the shopping experience before, during and after checkout' [[10]] and will include information like account balances, real-time alerts and loyalty programs as well as Priceless offers and experiences.

The NFC Cloud Wallet model was proposed and described in [1], [2]. They used an improved version of Chen's protocol [3] for customer authentication. An overview of an improved model of the NFC Cloud Wallet was proposed in [4].

IV. GSM Authentication

When a mobile device signs into a network, the Mobile Network Operator (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 a Mobile Switching Centre (MSC). The AuC stores two encryption algorithms, A3 and A8, as well as a list of all subscriber identities along with their corresponding secret keys K_i. The key K_i is also stored in the SIM. The AuC first generates a random number, denoted by R. This R is used to generate two responses: a signed response S and a key K_c as shown in Figure 1, where $S = E_{K_i}(R)$ uses the A3 algorithm and $K_c = E_{K_i}(R)$ uses the A8 algorithm [11].

(R, S, K_c) is known as the *Authentication triplet* generated by the AuC. The AuC sends this triplet to the MSC. On receiving a triplet from the AuC, the MSC forwards its first element R to the mobile device. The SIM of the mobile device computes the response S from R, using K_i which is stored in the SIM. The mobile device transmits S to the MSC. If this S matches the S in the triplet, then the mobile is authenticated. K_c is then used for communication encryption between the mobile device and the MNO.

Table I describes the abbreviations used in the proposed protocol.

V. The Proposed Protocol

This section describes our proposed protocol for micropayments based on NFC and cloud architecture. The assumptions are outlined as follows:

The proposed protocol is based on a cloud architecture where the cloud elements are managed by MTD (MNO Transaction Department). The MTD, under control on the respective MNO, is a dedicated financial department that deals with NFC transactions of the customers. The SE used in this protocol is divided into two sections: one part, residing in the SIM, is used for authentication of a customer, whereas the other part, residing in the cloud, is used to store sensitive banking information of the customer. The customer registers his credit/debit card details with the MTD through respective MNO. Since our protocol supports multiple accounts for a single customer, a customer can register more than one credit/debit card with the MTD. 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 MTD, and this is stored in the SE of his SIM. The MTD stores these details in the cloud. The mobile device has a valid SIM and is connected to the respective MNO through the GSM network. Communication over the GSM network is encrypted as specified in GSM standard. The communication between different entities of the GSN network is considered to be secure. The MNO may be linked to the customer through its own Base Station or through a Base Station of some other network. In the latter case, the proposed protocol should not disclose any sensitive information to the Base Station. The mobile device is connected to the shop terminal over an NFC link, but note that the NFC link is not secure and can be eavesdropped. The shop does not use

TABLE I
Abbreviations

App_{ID}	Approval ID. Generated after credit approval
Acc_{ID}	Account ID of the customer
AuC	Authentication Center (subsystem of MNO)
Cr_{req}	Credit Request Message
Cr_{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	$E_{K_i}(R)$ using A8 algorithm
K_1	Encryption key generated by the SIM
K_2	MAC key generated by the SIM
K_3	Encryption key generated by shop (the PoS)
K_4	MAC key generated by shop
K_{pub}	Public key of MTD
K_{pr}	Private key of MTD
K_{sign}	Signing key of MTD
K_{ver}	Verification key of MTD
LAI	Local Area Identifier
MNO	Mobile Network Operator
MTD	MNO Transaction Department
PI	Payment Information
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_a	Approval Time Stamp
TS_s	Shop Time Stamp
TS_t	Transaction Time Stamp

any link with the MNO for transactions. However the shop needs to trust the MNO so that a message digitally signed by the MNO is considered authentic and its contents are trusted by the shop. For simplicity, we refer to the mobile device and SIM as a single unit called the 'Mobile Device *(MD)*'. K_{sign}, K_{ver} are the signing and verification keys respectively of the MTD, whereas K_{pr}, K_{pub} are the private decryption and public encryption keys respectively of the MTD.

Table I describes the abbreviations used in the proposed protocol.

The protocol executes in three different phases; customer identification and credit check, customer authentication, and transaction execution.

A. Customer Identification and Credit Check

This phase is initiated once the customer agrees with the total price displayed on the shop terminal and places his cell phone on the shop NFC enabled point. An NFC link is established between the mobile device and the shop terminal.

Step 1: The mobile device sends payment Information *PI* request message to the shop terminal.

Step 2: The shop terminal forms *PI* message containing Total Price (TP), a temporary shop ID (T_{SID}), and a Time Stamp (TS_s) and sends it to the mobile device.

$$PI = TP \| T_{SID} \| TS_s$$

The T_{SID} acts as one time ID of the shop and gets updated after each transaction.

Step 3-4: Once the payment information is received from the shop, the application installed on the mobile device askes for PIN authentication from the user. This is for assurance that the customer is the legal owner of the mobile device. After a successful PIN verification, the mobile device needs credit approval from respective MTD indicating that the customer has sufficient funds in his account to pay the required amount. This information does not need to be disclosed to BS or any other entity of GSM network other than the MTD. As assumed earlier, the communication over the GSM link in encrypted according to GSM standard, but BS decrypts all information. To avoid decryption at BS level, the mobile device generates two keys K_1, K_2 for encryption and MAC respectively. The mobile device forms a credit request message Cr_{req} for credit approval from the MMTD, namely,

$$Cr_{req} = PI \| IMSI \| Acc_{ID}$$

The Cr_{req} message is encrypted with the key K_1 and MAC is computed with the key K_2 to provide data integrity. Both the keys, K_1 and K_2, are digitally signed with the public key of the MTD and the entire message is sent to the MTD..

Step 5: On receipt of this message, the MTD first decrypts it with its private key K_{pr} to extract the encryption and MAC keys, K_1 and K_2. It verifies the MAC and if successful, decrypts the Cr_{req} message. The MTD identifies the customer from IMSI in the Cr_{req} and communicates it to the cloud for a credit check against the account ID of the customer. If the customer has sufficient funds in his mentioned account, the MTD requests a fresh authentication of the customer prior to proceed to any transaction process.

Step 6-11: The MTD sends an authentication request message to MSC/AuC. The MSC follows standard procedure to authenticate a customer as described in the GSM standard. In case of successful authentication, an authentication success message is sent to the MTD.

Step 12: Once the customer is authenticated, an approval ID (App_{ID}) is generated by the MTD. App_{ID} acts as in index to a table storing information about the amount to be credited, the destination Shop ID, the time stamp and the customer ID (IMSI). This helps in resolving any disputes in future. The MTD forms a new string Cr_{app} indicating credit approval, namely,

$$Cr_{app} = PI \| TS_a \| App_{ID}$$

The MTD computes a signature with the signing key K_{sign} over the plaintext and encrypts the string Cr_{app} with the key K_1. The encrypted Cr_{app} along with its signature is transmitted to the mobile device. Cr_{app} cannot be decrypted by the BS as the BS lacks the encryption key K_1.

Step 13-16: The mobile device decrypts the message with the encryption key K_1 to obtain Cr_{app}. It compares the PI contents in both Cr_{req} and Cr_{app} messages. Moreover, the approval time stamp, TS_a, must be in a defined time window.

The mobile device also verifies the signature which was computed over the plaintext. This provides data integrity, data origin authentication and non-repudiation of the Cr_{app} message. After successful verification, the mobile device forwards Cr_{app} to the shop along with the corresponding signature.

Shop verifies the signature sing K_{ver} to detect any alteration and compares the PI contents in the Cr_{app} message to the one it initially sent in message 2. In the case of an invalid signature, the shop discards the message and rejects the payment. A successful verification indicates that the customer is legitimate and that the MTD has obtained agreement from the customer to pay. 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. The shop now needs to send its banking details to the MTD for transaction. The banking details may include bank account title, account number, bank code, branch code etc. The banking details are sensitive information and should not be disclosed to any entity other than the MTD, even the mobile device. The shop generates encryption and MAC keys, K_3 and K_4 to secure banking details. It encrypts the banking details with the key K_3, and computes MAC over the ciphertext with the key K_4. It also form a string, K_{info}, containing the information about the keys as follows:

$$K_{info} = (K_3 \| K_4) \oplus App_{ID}$$

The role of App_{ID} in this step is to bridge the authentication phase to the transaction execution phase. The shop encrypts the string K_{info} with the public key of the MTD K_{pub} and sends it to the MTD. This detail is transmitted to the MTD through the mobile device but the latter cannot decrypt this

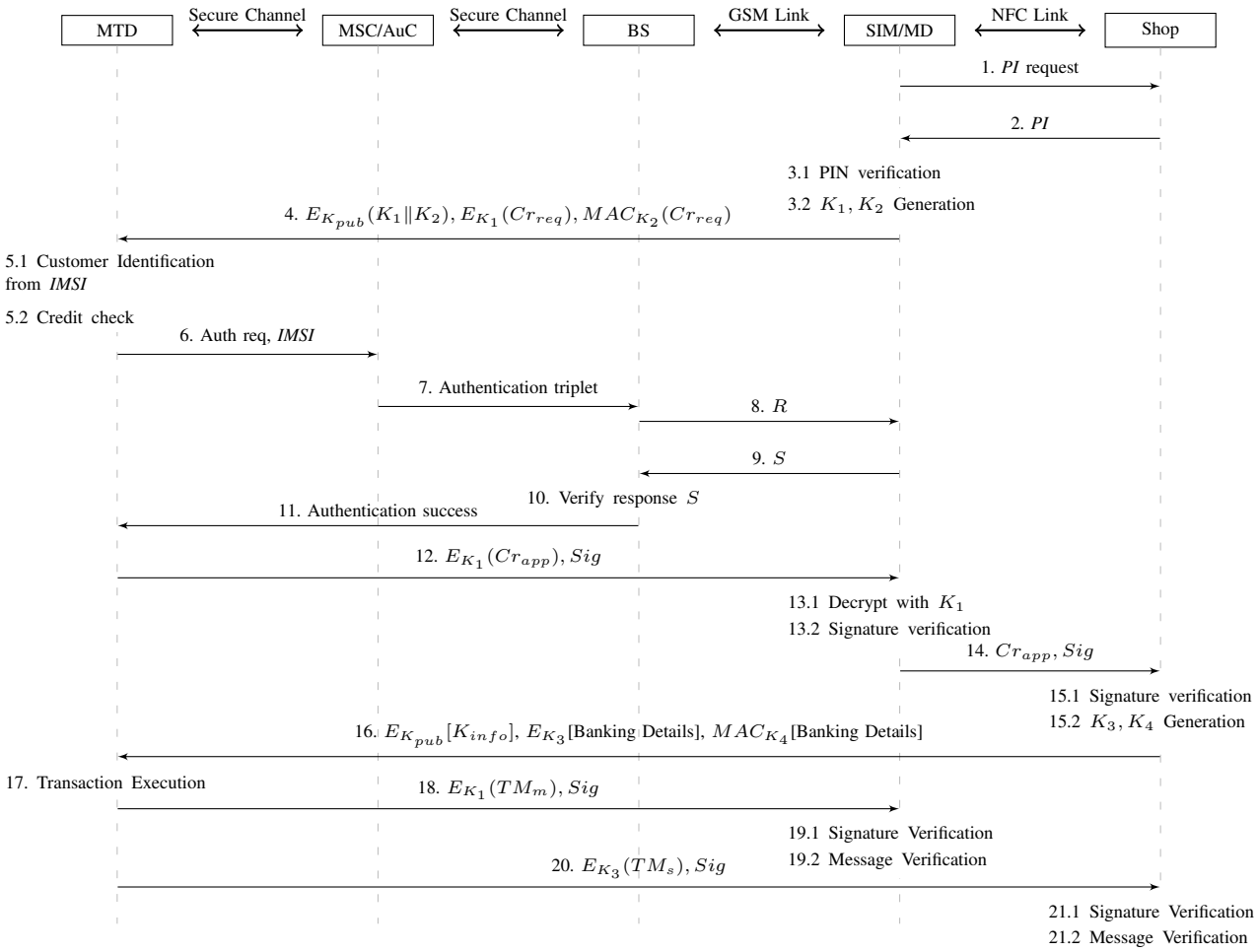

Fig. 2. The Proposed Customer Authentication Protocol

information. This forms a virtual tunnel between the shop and the MTD through the mobile device.

Step 17: Since the MTD knows the App_{ID}, it can get K_3 and K_4 to decrypt the banking details of the shop. The MTD transfers the requested amount from the customer account to the shop account.

Step 18-21 After a successful transaction, the MTD generates a transaction number TSN and corresponding time stamp TS_t and forms a Transaction Message for the mobile device TM_m and a Transaction Message for shop TM_s as follows:

$$TM_m = PI \| TSN \| TS_t$$
$$TM_s = TM_m \| [\text{Banking Details}]$$

The MTD encrypts TM_m with the key K_1 and computes a signature over the ciphertext. It sends encrypted TM_m and 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 and exits the transaction. Otherwise, it decrypts the message and verifies the contents.

The MTD forms the Transaction Message TM_s for the shop by appending the Shop Banking Details to the previously formed TM_m. It encrypts TM_s with the key K_3 and computes a signature over the ciphertext. The MTD sends the encrypted message along with its signature to the mobile device which relays it to the shop. The mobile device can neither decrypt this message as it does not possess K_3, 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 subsequent clarification, it can approach the MNO quoting the Transaction Number TSN and Approval ID App_{ID} received in step 14.

VI. Analysis

In this section, we analyze 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 analyze 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. Let assume that the dishonest knows the IMSI and Acc_{ID} ($IMSI'$, Acc'_{ID}) of the target victim. The dishonest customer fabricates Cr'_{req} message in step 4 as:

$$Cr'_{req} = PI \| IMSI' \| Acc'_{ID}$$

As this message can be decrypted only by the MTD, the malicious contents remain undetected by all other entities of the GSM network. The MTD decrypts the message identifies the customer from $IMSI'$. Since the target victim is a legitimate customer and has sufficient funds in his account, the MTD proceeds to fresh authentication of $IMSI'$. The MSC/AuC provide authentication triplet in step 7 corresponding to $IMSI'$. The attacker cannot compute a valid response S as he lacks the valid key K_i to compute the response. So, the attacker's response S' in step 9 to the random challenge R is different from the S in the authentication triplet. This fails the authentication and the protocol stops. Thus, someone else ID cannot be successfully used in this protocol.

Scenario 2. A dishonest customer plans buy goods without any payment. He accomplish this plan by providing his own banking details, instead of the shop, as the recipient. He blocks the legitimate message 16. The attacker, then, generates his own set of keys, K'_3 and K'_4, and fabricates message 16 with own banking details and sends it to the MTD. The MTD performs transaction against the information provided by the mobile device by deducting amount from the customer account and paying back in the same customer's account (both accounts may be different to avoid detection).

After the transaction execution, the MTD sends 'receipts' in message 18 and 20. The mobile device blocks message 20 as this message contains the information of the customer bank details as it was used during the transaction. The dishonest customer needs to replace the banking detail in this massage with the shop banking details. The customer can decrypt message 20 as it is now encrypted with the customer's malicious key K'_3. He needs to change the banking details and encrypt with the shop generated key K_3 in step 15.2. 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 the banking details, it will void the signature. If the customer does not alter the message in order to keep 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'_3). 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 above attack where the dishonest customer plans to buy some goods without payment. The dishonest customer does not communicate with the MTD since it is not successful as described above; rather the customer impersonates as MTD 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 MTD, his account cannot be debited. In the original protocol, the shop receives three messages from the mobile device, message 1,14 and 20. Message 1 is originated by the mobile device, whereas message 14 and 20 are actually originated by the MTD 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 MTD. These messages contains a Temporary Shop ID (T_{SID}) and a Time Stamp of the shop (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 MTD, 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 the Credit Request message, Cr_{req}, in step 4 to MTD, where $TP' < TP$. The mobile device receives the Credit Approve message, Cr_{app}, in step 12 from the MTD confirming that the initially requested amount TP' has been approved for transaction. But the mobile device needs to intimate the shop in step 14 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 MTD is transmitted to the shop. Hence, this attacks fails on proposed protocol.

Scenario 4. A dishonest customer wants to pay through a mobile device which he does not own. He might have stolen that device or found it as lost property. If the SIM is still valid, it can be used for transaction. a After the device receives payment information *(PI)* from the shop in step 2, the application installed on the mobile device require PIN verification from the customer. Since the customer does not owns the mobile device, he does have the knowledge about the PIN. So the protocol does not proceeds further. Additionally, the application can be designed to get blocked after a limited number of failed attempts of PIN verification. This provides security to the customers who feels secure that their lost mobile device could not be used for any monetary transaction even if the SIM is active.

B. Dishonest Shop

Scenario 5. 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 the Credit Request message, Cr_{req}, in step 4. A mobile device cannot send more than the required price unless the device itself is

compromised. Therefore, a shop can not get more than the required amount in this protocol.

Scenario 6. The shop is dishonest and repudiates the receipt of transaction execution message in step 20. 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 MTD indicating a Transaction Serial Number TSN (received in step 18). The TSN is linked to the Approval ID App_{ID} generated in step 12. Since both the values are digitally signed by the MTD, the mobile device can approach the MTD regarding any dispute.

C. Messages Security

Apart from the above-mentioned scenarios, we also analyzed 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 over GSM link. The data over NFC link in *Authentication* and *Credit Approval* phase (Step 1, 2 and 14) is sent in clear. This data does not contain any sensitive information. Total Price may be considered a 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 considered TP as not so sensitive information to be protected over NFC link.

Another information that is sent in clear over the NFC link is the Credit Approval ID (App_{ID}) in the (Cr_{app}) message (step 14). 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 a sensitive information in this scenario.

Role of Approval ID in message 16. App_{ID} acts as a bridge between the Financial Approval phase and the Transaction phase. It adds freshness to message 16, so it cannot be replayed in future. App_{ID} is XORed to avoid increase in the message length. Any alternation in first part of the message 16 (K_{info}) results in invalid keys K_3' and K_4'. This invalidates the MAC and hence detected.

Non-repudiation of Transaction Messages. Transaction Execution messages (Step 18, 20) are digitally signed by the MTD. In case of any dispute about payment, the MTD has to honour both messages. So the customer and the shop, both are completely secured about transaction.

Disclosure of Relevant Information.

Cr_{req} message containing price information is not disclosed to the base station or any other GSM entity apart from the MTD.

Shop banking detail is a sensitive information as it contains the bank account number etc. It is encrypted not only over GSM link but also over 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 MTD, yet the former cannot decrypt this information.

Similarly, the account information of the customer is not communicated to the shop in Cr_{app} message.

New set of Keys for every transaction.

The encryption and MAC keys for Cr_{req} message, K_1 and K_2, are freshly generated by mobile device in each round. Similarly, the keys K_3 and K_4, generated by the shop are fresh for each transaction.

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 [12]. 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.

VII. 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 NFC Cloud Wallet model [1], NFC payment application [2] and W. Chen et al [3] 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.

References

[1] P. Pourghoumi and G. Ghinea, "Managing NFC Payments Applications through Cloud Computing," in *7th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2012, pp. 772–777.

[2] P. Pourghomi, M. Saeed, and G. Ghinea, "A Proposed NFC Payment Application," in *International Journal of Information Security*, Under review 2013.

[3] W. Chen, G. Hancke, K. Mayes, Y. Lien, and J.-H. Chiu, "NFC Mobile Transactions and Authentication Based on GSM Network," in *International Workshop on Near Field Communication*. IEEE Computer Society, 2010, pp. 83–89.

[4] P. Pourghomi, M. Saeed, and G. Ghinea, "Trusted Integration of Cloud-based NFC Transaction Players," in *Conference on Communications and Network Security*. IEEE, Under review 2013.

[5] P. Pourghoumi and G. Ghinea, "Challenges of Managing Secure Elements within the NFC Ecosystem," in *7th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2012, pp. 720–725.

[6] J. Pailles, C. Gaber, V. Alimi, and M. Pasquet, "Payment and privacy: A key for the development of nfc mobile," in *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, may 2010, pp. 378 –385.

[7] Google, "Goole Wallet," March 2013. [Online]. Available: http://www.google.co.uk/wallet/faq.html

[8] MasterCard, "MasterPass," March 2013. [Online]. Available: https://masterpass.com/online/Wallet/Help?cid=127568

[9] NFC World, "MasterCard enters the mobile wallet market," May 2012. [Online]. Available: http://www.nfcworld.com/2012/05/09/315600/mastercard-enters-the-mobile-wallet-market/

[10] ——, "MasterCard unveils MasterPass digital wallet and mobile payments platform," February 2013. [Online]. Available: http://www.nfcworld.com/2013/02/25/322610/mastercard-unveils-masterpass-digital-wallet-and-mobile-payments-platform/

[11] *ETSI Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.11)*, European Telecommunications Standards Institute (ETSI Std. Version 5.0.0, December 1995. [Online]. Available: http://www.ttfn.net/techno/smartcards/gsm11-11.pdf

[12] M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," in *International Conference on Advances in Cryptology ASIACRYPT*, 2000, pp. 531–545.