Mobile Transactions over NFC and GSM

CONFERENCE PAPER · AUGUST 2014

4 AUTHORS, INCLUDING:

Pardis Pourghomi
The American University of the Middle East
10 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Gheorghita Ghinea
Brunel University London
251 PUBLICATIONS **1,260** CITATIONS
SEE PROFILE

Mobile Transactions over NFC and GSM

Muhammad Qasim Saeed
and Colin Walter

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

Pardis Pourghomi
and Gheorghita Ghinea

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

***Abstract*—Dynamic relationships between Near Field Communication (NFC) ecosystem players in a monetary transaction make them partners in a way that they sometimes require to 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 only a Secure Element (SE) in an NFC enabled mobile phone. In this paper, we propose a protocol for NFC mobile payments over NFC using Global System for Mobile Communications (GSM) authentication. 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. This elimination makes the protocol more practicable and user friendly. A detailed analysis of the protocol discusses multiple attack scenarios.**

Keywords–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. 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 over other players. Consequently, the service environment does not meet the optimal conditions for take-up. This has motivated us to extend current NFC ecosystem models to accelerate development. Our goal is to provide a concept for an NFC ecosystem that is technically feasible, accepted by all parties involved and provides an improved business case for each of the players. One of the main players in the NFC ecosystem is the Mobile Network Operator (MNO). The advantage an MNO has over other parties is that it owns a Secure Element (SE), the Subscriber Identity Module (SIM) card, that stores and protects the security parameters. Unlike other forms of SE, the SIM card can be easily managed by the MNO over-the-air. Thus, we foresee that the MNO will play a major role in future in the NFC ecosystem.

A. Our Contribution

Here, we extend the earlier proposed mobile transaction mechanism mentioned in [1]. 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 protocol. This makes our work more flexible and it can even be used for monetary transfer between two individuals provided that the payer has registered a bank account with his MNO. 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 cutomer account details. The authentication of the customer by the MNO is based on a GSM authenticating mechanism. The GSM standard, although not so secure, is still widely used for mobile communication, accounting for more than five billion subscriptions [2]. The idea is to reuse the existing cryptographic functionalities of the GSM standard thus reducing a need of additional cryptographic modules. 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. But, unlike PayPal, our system uses the existing features of GSM standard for secure transactions. An overview of this model was proposed in [3].

This paper is structured as follows: Section II introduces the SE with a discussion of its management issues. We also highlight some advantages of having a cloud environment for mobile payment transactions. Section III describes related literature while Section IV recalls the essentials of GSM authentication. Then Section V introduces our proposed transaction protocol in detail followed by its analysis. Finally, Section VII places our solution in context, summarises how it operates, and draws some conclusions.

II. MANAGEMENT OF THE 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 that can be found in a smart card [4]. It provides storage within the mobile phone and 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

 ISBN: 978-1-61208-353-7

can be used to store other applications which require security mechanisms and it is 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, preferably over-the-air. Still, the ownership and control of the SE within the NFC ecosystem may result in a commercial and strategic advantage, as well as reluctance to participate from other providers. However, some solutions are already in place [4] and researchers are developing further models to overcome this problem.

A. Advantages of the 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 retrieved from a remote virtual SE which is stored within the cloud environment. The mobile phone SE provides temporary storage and authentication assets for the transaction to take place, and all communication between the cloud provider and the vendor terminal is established through the NFC phone.

An issue with SEs is that companies have to meet the requirements of organisations such as EMVco [5] to provide high level security to store personal data. This makes the SE expensive for companies. However, a cloud-based approach would transfer this cost. Then the SE in the NFC phone need only be responsible for user/device authentication and not for storing personal data. This improves the cost efficiency of the SE compared with the present, enabling many more secure applications to be supported because of the reduced space. Also, the NFC controller chips could 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 many more applications as they are no longer stored 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 single, shared, secure, storage space. Moreover, fraud detection would be instantaneous 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". The communication between the mobile phone and the PoS is carried out through NFC technology that transmits the payment details to the merchant's PoS. The Google Wallet is in the form of an Android application with a Secure Element (SE) on the customer's mobile phone and an SE in the cloud. The customer will have an account with Google Wallet which includes the relevant registered credit/debit cards whose details are stored in the online SE using secure servers. The transaction takes place in the form of a virtual prepaid credit card which is transfered to the merchant's PoS when the customer taps his phone on the PoS. For this, the Google Wallet app initializes the SE on the mobile device and a secure channel is established between the SE of the device and the SE in the cloud [6].

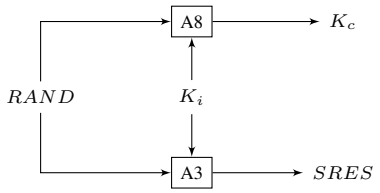

FIGURE 1. GENERATION OF K_c AND $SRES$ FROM $RAND$.

Our model differs from Google Wallet in the context of the location of the SE. We use the SIM as the SE in the mobile device whereas, the Google wallet requires an embedded SE. If a customer changes his handset, our approach still works as only the SIM needs to be replaced in the new device. This makes the approach more flexible.

Gerald Madlmayr and Josef Langer presented a purse-based micro-payment system [7]. They designed a pre-paid wallet where the money is stored in the Secure Element in the mobile device. The user can top-up his account Over-The-Air (OTA), anywhere and at any time.

Other solutions include "MasterPass" [8]. This service was developed by MasterCard as an extended version of the PayPass Wallet Services [9] that provides a digital wallet service for safe and easy online shopping.

IV. 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 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 $RAND$. This is used to generate two responses: a signed response $SRES$ and a key K_c as shown in Figure 1, where $SRES = E_{K_i}(RAND)$ uses the A3 encryption algorithm and $K_c = E_{K_i}(RAND)$ uses the A8 encryption algorithm [10].

$(RAND, SRES, K_c)$ is known as the *Authentication triplet* generated by the AuC. The AuC sends this triplet to the MSC. On receiving a triplet, the MSC forwards $RAND$ to the mobile device. The SIM computes the expected response $SRES$ from $RAND$, using A3 and the key K_i which is stored in the SIM. The mobile device transmits $SRES$ to the MSC. If this $SRES$ matches the $SRES$ in the triplet, then the mobile is authenticated. K_c is then used for communication encryption between the mobile device and the Base Station (BS).

 ISBN: 978-1-61208-353-7

TABLE I. ABBREVIATIONS

Acc_{ID}	Account ID of the customer
App_{ID}	Transaction approval message for customer account ID
AuC	Authentication Center (subsystem of MNO)
BS	Base Station
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	$E_{K_i}(RAND)$ 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
MD	Mobile Device
MNO	Mobile Network Operator
MSC	Mobile Switching Centre
MTD	MNO Transaction Department
PI	Payment Information
$RAND$	Random Number (128 bits) generated by MNO
R_s	Random number generated by SIM (128 bits)
SBAD	Shop Bank Account Detail
SE	Secure Element
SRES	Expected Response
TEM_u	Transaction Execution Message for user
TEM_s	Transaction Execution Message for shop
TMSI	Temporary Mobile Subscriber Identity
TP	Total Price
T_{SID}	Temporary Shop Identifier
TS_a	Approval Time Stamp
TS_s	Shop Time Stamp
TS_{tr}	Transaction Time Stamp
TSN	Transaction Serial Number

V. THE PROPOSED PROTOCOL

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

Our proposal is based on a cloud architecture where the cloud is being managed by the MNO. The cloud is used to store sensitive information about customers. A customer, who is a user of a cell phone, opens up a payment account with the respective MNO prior to use of the proposed payment feature. Each account is identified by a unique identity, the *Account ID* or Acc_{ID}. The account is either a *pre-paid* or a *pay-as-you-go* account. In the former type of account, the customer needs to top-up his account by either pre-paid vouchers or cash. This feature is more suitable for customers who do not have bank accounts. The amount a customer has in his pre-paid account is stored in the cloud against respective Acc_{ID}. In the *pay-as-you-go* type of account, the customer provides his banking credentials, like credit/debit card details to the MNO. The banking credentials are verified in the registration process by the MNO from respective bank and are stored against the respective Acc_{ID} in the cloud. A customer can have one or more accounts of either type and has the option to select one account while payment.

We suggest a dedicated department, the MNO Transaction Department (MTD), to manage the monetary transactions. A virtual secure tunnel is established between the mobile device and the MTD to ensure the security of the messages. The virtual tunnel is of special significance when the BS of some other network is used for the transaction as, in such scenario, the MNO responsible for monetary transaction does not want to reveal any sensitive information to the BS.

The mobile device communicates with the MNO over the standard GSM link. Communication over the GSM link between the mobile device and the BS is encrypted as specified in the GSM standard. Otherwise, communication between different entities of the GSM network is not considered to be secure and so encryption needs to be added where appropriate.

The MNO may be linked to the customer through its own BS or through a BS of some other network. Especially in the latter case, the proposed protocol should not disclose any sensitive information to the other network. The shop communicates with the MNO through the customer's mobile device using NFC and the GSM link. The shop PoS terminal does not require to be registered with the MNO. This makes the protocol more flexible and can also be used for monetary transactions between two individuals (the payer and the payee are analogous to the customer and the shop respectively).

The mobile device is connected to the shop terminal over an NFC link, but note that, although the NFC link is generally regarded as secure because of its short range of operation, yet it can be eavesdropped [11] or vulnerable to relay attacks [12]. A recent study suggested that any metallic object in the vicinity of an NFC link, even a shopping trolley, can act as 'rogue' antenna to eavesdrop the communication [13].

The shop does not use any direct link with the MTD for transactions. However, it needs to trust the MTD, i.e., a message digitally signed by it should be considered authentic and its contents trusted.

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.

Before any transaction can take place, the customer must have registered at least one account with his MTD and the merchant must have downloaded an (MNO-independent) application for performing our protocol. The merchant's application must be able to form several messages, such as PI (detailed below), in a format acceptable to the MTD of any MNO. In addition, the merchant needs to obtain and store trusted certificates for the keys of the MTDs he is willing to trust.

The protocol executes in three different phases: customer identification and credit check, customer authentication, and transaction execution. The steps of the protocol are illustrated in Figure 2, with numbering as in the text. Table I describes the abbreviations used in the proposed protocol.

A. Phase I: Customer Identification and Credit Check

This phase is initiated when the store owner sends the payment request to his NFC reader and the customer places his MD on the shop's NFC enabled point.

Step 1: The MD and shop terminal establish an NFC connection.

 ISBN: 978-1-61208-353-7

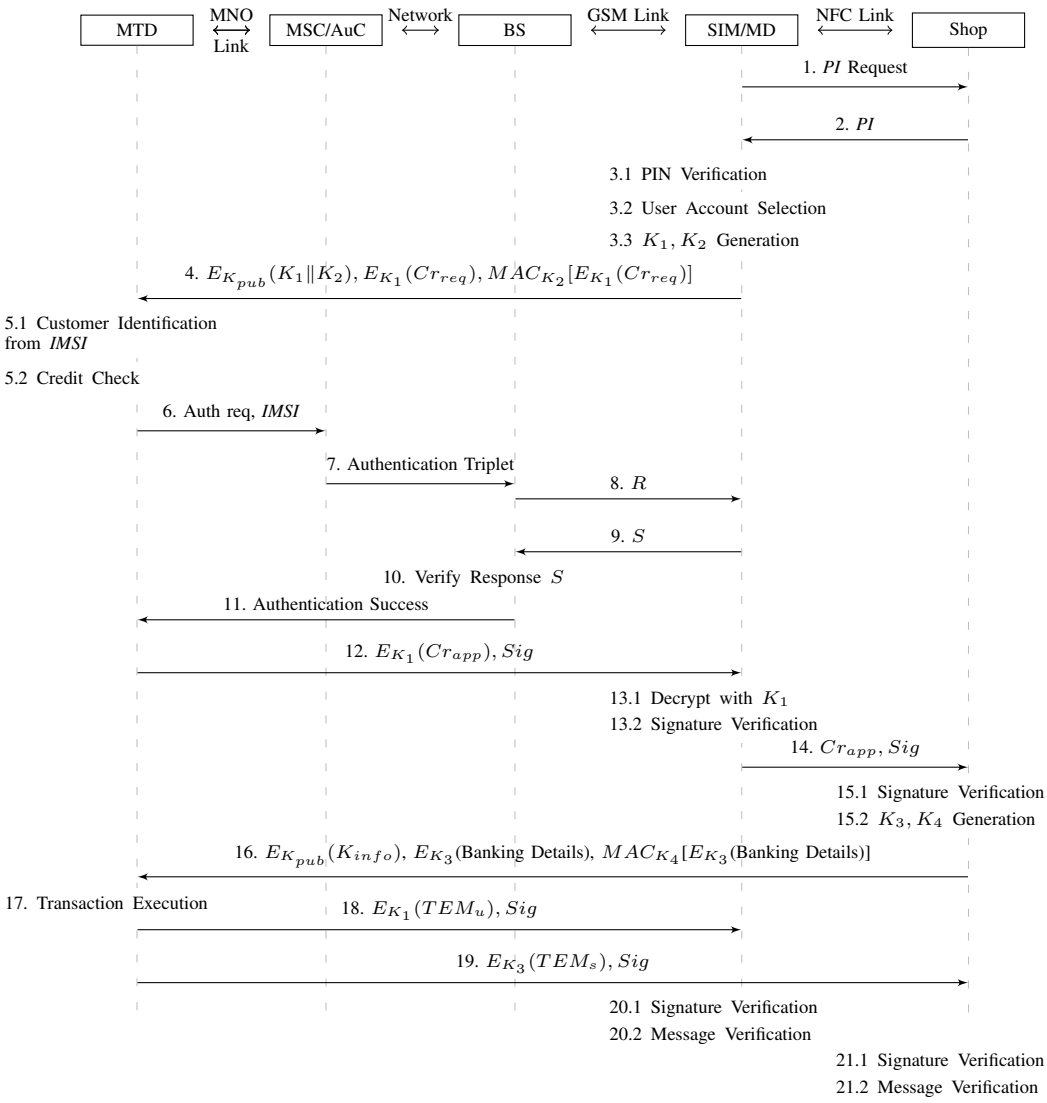

FIGURE 2. THE PROPOSED CUSTOMER AUTHENTICATION & PAYMENT PROTOCOL

Step 2: The shop terminal forms the Payment Information message *PI* containing at least the Total Price TP, a temporary shop identity T_{SID}, and the shop's Time Stamp TS_s, and sends it to the MD:

$$PI = TP \| T_{SID} \| TS_s \tag{1}$$

The T_{SID} acts as one time identifier used by the shop to identify the transaction. It is updated and fresh for each transaction. Optionally, PI may also contain a description of the shop and the goods which would appear on the customer's credit/debit card account statement.

Steps 3-4: Once the payment information is received from the shop, the application installed on the MD displays the transaction amount TP to the user and asks him to select a payment account and provide PIN authentication. This is for assurance that the customer is the legal owner of the mobile device, and therefore also the owner of the account which will be used for payment. It also provides confirmation that the amount and account details are accepted by the user.

After successful PIN verification, the mobile device needs to obtain a credit approval certificate for the shop from the respective MTD indicating that the customer has sufficient funds in his account and has agreed to pay the required amount. The information in this exchange should not be accessible to the BS or any other entity of GSM network other than the MTD. To provide a secure connection for this exchange between the MD and the MTD, the former generates two keys K_1, K_2 for symmetric encryption and MAC respectively. The actual encryption process used here is irrelevant, but will most likely be specified by the card provider and EMV requirements. It should not depend only on quantities known to either the BS or the MNO since only the MTD should be able to perform the decryption. The mobile device forms a credit request message Cr_{req} for credit approval from the MTD, namely,

 ISBN: 978-1-61208-353-7

$$Cr_{req} = PI \| IMSI \| Acc_{ID} \quad (2)$$

This is encrypted with K_1 and a MAC is computed on the ciphertext using K_2 to provide data integrity. Then the keys, K_1 and K_2, are encrypted with the MTD's public key K_{pub}. The entire message, consisting of the encrypted keys, the encrypted credit request and the MAC value, is sent to the MTD as in message 4 of Figure 2.

Step 5: Upon receipt of this message, the MTD starts by decrypting the first part of the message with its private key K_{pr} to extract the encryption and MAC keys, K_1 and K_2. It then verifies the MAC and in case of successful verification, it decrypts the second part of the message, containing Cr_{req}, and checks the freshness of the shop's time stamp in PI. The MTD identifies the customer from the IMSI in Cr_{req} and performs a credit check against the named account Acc_{ID}.

B. Phase II: Customer Authentication

Steps 6-11: Whether or not the credit and freshness checks are successful, the MTD sends an authentication request message to the MSC/AuC to authenticate the MD. The MD has already been identified by its IMSI. However, since the IMSI is not a secret, it may be used by a malicious party. To counter such threat, the MD needs to be authenticated under the IMSI claimed in Cr_{req} prior to any monetary transaction. With this IMSI, the MSC follows the usual procedure to authenticate an MD and it does not required further user interaction. So, in the case of successful authentication, the usual success message is sent from the BS to the MTD.

Step 12: If the credit check fails or the authentication success message is not received, the protocol is terminated with the sending of a fail message from the MTD to the MD. Termination does not occur before the authentication in order to hide the result of the credit check from an unauthenticated attacker. Otherwise, when both the credit check and the authentication are successful, a credit approval identifier App_{ID} is generated by the MTD. This acts as an index to a table in which the MTD stores information about the debit account, the amount to be transferred, the destination shop identity, a time stamp and the MD identity (IMSI). This identifier helps in resolving any disputes in the future but the details of the transaction are not contained therein.

The MTD now forms a new string Cr_{app} indicating credit approval for the Payment Information PI, namely,

$$Cr_{app} = PI \| TS_a \| App_{ID} \quad (3)$$

The MTD computes a signature with its signing key K_{sign} over the hashed 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. The former cannot be decrypted in transit as the encryption key K_1 is unavailable, nor is Cr_{app} revealed by applying the verification key to the signature because of the hashing. Moreover, because of TS_s, TS_a or App_{ID}, the message differs each time even if the user buys the same goods on successive occasions.

Steps 13-16: The mobile device decrypts the message with the encryption key K_1 to obtain Cr_{app} and forwards it to the shop along with the corresponding signature. The shop verifies the signature using K_{ver} and compares the PI content in the Cr_{app} message to the one it initially sent in message 2. In the case of an invalid signature or a mis-match with PI, the shop discards the message, rejects the payment, and withholds the goods or services from the customer. 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 (the MTD), trusted by both other parties, provides assurance that the other party is willing to pay the specified price.

The shop now needs to send its banking details to the MTD to complete the transaction. The banking details may include the account name and number, the bank and branch codes, etc. This is sensitive information and should not be disclosed to any entity other than the MTD, not even to the MD. The shop therefore generates encryption and MAC keys, K_3 and K_4 to secure its banking details. It encrypts the banking details with the key K_3, and computes a MAC over the ciphertext with the key K_4. It also form a string, K_{info}, containing the information about the keys as follows:

$$K_{info} = K_3 \| K_4 \| App_{ID} \quad (4)$$

The role of the approval identifier App_{ID} in this step is to enable the MTD to connect the authentication phase to the transaction execution phase. The shop encrypts the string K_{info} with the public key K_{pub} of the MTD and sends it to the MTD via the MD. This forms a virtual tunnel between the shop and the MTD through the MD, as the latter cannot decrypt the message content. Note, however, that the shop needs to be certain it has K_{pub} correctly from the MTD, and not a key substituted by an attacker.

C. Phase III: Transaction Execution

Step 17: The MTD associates the App_{ID} received in the step 16 with the already stored App_{ID} (step 12). It decrypts the banking details of the shop with keys K_3, K_4 and transfers the approved amount, stored against corresponding App_{ID}, to the shop account. The MTD flags the App_{ID} indicating that the transaction has been executed to ensure that the same App_{ID} could not be used again.

Step 18-21 After a successful transaction, the MTD generates a Transaction Serial Number (TSN) and forms Transaction Execution Messages, TEM_u and TEM_s for the MD and the shop respectively.

$$\begin{aligned} TEM_u &= PI \| TSN \| TS_{tr} \| Acc_{ID} \\ TEM_s &= PI \| TSN \| TS_{tr} \| SBAD \end{aligned} \quad (5)$$

The MTD computes a signature on the hashed plaintext, encrypts TEM_u with the key K_1, and sends it to the MD. The MD decrypts the message and verifies the signature. An invalid signature indicates that the transaction confirmation has been accidentally or deliberately corrupted en route. In such a case, the MD enquires about the transaction from the MTD. If the transaction has already been executed, the MD asks for a fresh confirmation message. Otherwise, it is obvious that message 16 has not been delivered to the MTD. This may happen if a malicious party has blocked the message

 ISBN: 978-1-61208-353-7

from reaching the MTD and has instead transmitted a fake transaction confirmation message. Of course, such a fabricated message cannot go undetected as it is signed by the MTD. In such scenario, the MD asks the shop to resend message 16.

The MTD also forms TEM_s for the shop by appending the Shop's Banking Details as shown in Eq (5). The MTD computes a signature over the hashed plaintext and encrypts TEM_s with the key K_3. The MTD sends this encrypted message along with its signature to the customer MD which relays it to the shop. The customer's MD 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 decrypts the message, verify its contents and the signature, thereby confirming that his account (rather than an attacker's) has been credited correctly. The contents consist of important transaction information exchanged during the transaction. Hence, if the shop wants any subsequent clarification, it can approach the MNO quoting the TSN and the App_{ID} received in step 14. Finally, if the shop is satisfied, it produces a receipt together with the goods or services for the customer.

VI. Analysis

In this section, we analyze the protocol from multiple perspectives to ascertain the strength of our protocol. This analysis encompasses the authentication and security of the messages. We assume that the MNO is trustworthy, whereas the customer or the shop can be dishonest, and there may be an active attacker listening to any of the NFC or other messages.

A. Dishonest Customer

Scenario 1. A dishonest customer plans to buy some products, making the payment from someone else's account. The PIN requirement in step 3 should force the customer to use his own mobile device to enact the protocol. Indeed the protocol depends on the strength of this PIN, just as is the case with credit card withdrawals. However, rogue applications on the MD could have already sniffed the PIN.

Assume that the attacker uses his own mobile and knows the IMSI and account numbers ($IMSI'$, Acc'_{ID}) of the target victim. He must fabricate Eq (2) as:

$$Cr'_{req} = PI \| IMSI' \| Acc'_{ID} \tag{6}$$

As this message can be decrypted only by the MTD, the malicious contents remain undetected by all other entities. The MTD decrypts the message and identifies the customer from $IMSI'$. Assuming the protocol does not fail here because the target victim is not a legitimate customer or the account has insufficient funds, the MTD proceeds to the fresh authentication of $IMSI'$. So the MSC/AuC provides the authentication triplet in step 7 corresponding to $IMSI'$. However, the attacker cannot compute the valid response S' as his mobile device lacks the necessary key K'_i. So, the authentication check fails and the protocol terminates. Thus, an incorrect identity cannot be successfully used in the protocol.

Scenario 2. Suppose a dishonest customer plans to buy goods without payment. He could accomplish this by providing his own banking details, instead of the shop's, to the MTD for the payment recipient. He then blocks the legitimate message 16, and replaces it as follows. Using his own keys K'_3 and K'_4, he fabricates message 16 with own banking details and sends it to the MTD. The MTD performs the transaction against this information, deducting the amount from the customer's account but paying it back into the same or another account of the customer. (These may be distinct in an attempt to avoid detection). After executing the transaction, the MTD sends 'receipts' in messages 18 and 19. The MD must block message 19 as this message contains the substituted bank details which the shop checks. So the dishonest customer needs to replace the banking details in this message with the shop's banking details. He can decrypt message 19 as it is encrypted with his own malicious key K'_3. However, he must now change the banking details and encrypt them with the shop's key K_3. As he lacks this key, he cannot generate a valid ciphertext. Moreover, the original message is protected by the digital signature. If the customer were to make any alteration to the banking details, it would void the signature which the shop verifies next. In neither case is the shop able to verify the transaction and a failure message is reported to the shopkeeper. Hence, the 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 MTD since he could not succeed in the way described above; rather, he masquerades as the MTD to the shop. The target of the customer is to send fake but acceptable receipts to the shop at the end of the protocol by replaying old legitimate, messages or fabricating new messages. Since the customer is not communicating with the MTD, his account will not be debited. In the original protocol, the shop receives three messages from the MD: messages 1, 14 and 19. Message 1 originates from the MD, whereas messages 14 and 19 actually originate from the MTD but are relayed by the MD to the shop. The dishonest customer needs to construct 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. They contain the Shop Identitifier T_{SID} and Time Stamp TS_s. T_{SID} is a random value generated by the shop every time at the start of the protocol. This value not only serves as a shop identifier during the protocol, but it also 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 to succeed. Hence, the dishonest customer is again unsuccessful. Of course, as usual in PKI, the shop should check the digital certificates of the MTD keys to justify its trust in them.

Scenario 3. Assume now that the dishonest customer plans to pay less than the required amount but claim payment of the full amount. To accomplish this, the MD sends TP' in the Credit Request message Cr_{req} of step 4 to the MTD, where $TP' < TP$. The MD receives the Credit Approval message, Cr_{app}, in step 12 from the MTD confirming that the initially requested amount TP' has been approved for transaction. But the MD needs to confirm to the shop in step 14 that the original amount, TP, is approved for transaction. Since the approved price is digitally signed by the MTD, it cannot be amended by the MD. So the actual price that is approved by the MTD is transmitted to the shop. As the shop application checks the

 ISBN: 978-1-61208-353-7

approved amount against that requested, this attack also fails.

Scenario 4. Here, 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 and the credit/debit cards have not been cancelled, it can still be used for transactions. After the device receives the payment information *PI* from the shop in step 2, the application installed on the mobile device requires PIN verification from the customer. Since the customer does not own the mobile device, he should not have knowledge of the PIN. So the protocol does not proceed further. Additionally, the application can be designed to be blocked in the MD and by the MTD after a limited number of failed attempts at PIN verification. This provides an assurance to the customers that their lost mobile device could not be used for any monetary transactions even while the SIM remains active.

B. A 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 sent to the MTD by the MD in the Credit Request message, Cr_{req}, in step 4. A mobile device cannot send more than the price contained in PI and approved by the user in step 3 unless the device itself is compromised. Therefore, a shop cannot obtain more than the agreed amount if, as requested, the customer checks the amount before entering his PIN.

Scenario 6. The shop is dishonest and denies receipt of the transaction execution message in step 19. In this way, the shop decides not to deliver the goods or services despite receiving the required amount. However, the MD has the signed receipt from the MTD with the TSN from Eq (5). This is linked to the approval App_{ID} generated in step 12. As both are digitally signed by the MTD, the customer can approach the MTD regarding any dispute. With knowledge of the account credited during the transaction and the shop receipt from the customer, the MTD can take action to identify the criminal and refund the customer.

C. Message Security

Apart from the above-mentioned scenarios, we also analyzed our protocols from various other angles. The data over the GSM link (between the MD and the BS) is encrypted according to the GSM specification. The data sent over the NFC link in steps 1, 2 and 14 are sent in the clear. This data does not contain any particularly sensitive information except perhaps for the TP. However, the range within which this data can be captured is very limited, and it is occupied by the shop keeper and the customer, at least one of whom should notice unwelcome devices (such as other NFC capable mobile phones) in the vicinity. The read range of the price displayed on both the shop till and the user's MD is much more than the range of the NFC link. Therefore, we considered PI as not sufficiently sensitive to need protection over the NFC link. Nevertheless, we should consider this in a little more detail.

Other information that is sent in clear over the NFC link includes the App_{ID} in the Cr_{app} message. At this point the attacker can hi-jack the protocol by blocking the communication of message 16, replacing it with his own forged message which contains his own bank details. There is no relevant data which is not known to the attacker. This results in a successful transfer of funds to the criminal and also a successful acknowledgment in step 18 to the legitimate customer. However, the shop owner will either not receive the transfer message in step 20, or will receive one which fails his verification. Thus, although the shop keeper will not then release the goods, the attacker will have obtained the funds. The solution is to include a means for the MTD to verify that message 18 comes from the same source as message 2.

We therefore propose the inclusion of a Diffie-Hellman key agreement (DH) between the MD and Shop during messages 1 & 2 in situations where the NFC link may be compromised. Then step 18 can include a proof of origin. Step 1 would include the public parameters for DH, and the MD's exponentiated value, while step 2 would include the shop's response of the other DH exponentiated value. As message 16 contains a MAC of the other components of message 16 using the DH shared key, the MD can check the authenticity of message 16, ensuring that the protocol has not been hi-jacked. However, an attacker who can hi-jack the protocol at step 18 could equally easily hi-jack it at step 1. This requires blocking the legitimate message PI and replacing it as necessary with PI' so that the MD agrees a shared key with the attacker instead of the shop and, later, the forged message 16 is authenticated by the MD. Since TP is not known to the attacker until PI is transmitted, the attacker needs to collect the legitimate PI first in order to include TP in PI', this being necessary to obtain the customer's agreement over the price. However, for this to succeed, the attacker must prevent the correct PI from reaching the MD. Consequently, the success of Diffie-Hellman key exchange between shop and MD cannot be prevented unless the attacker can guess TP correctly or the customer fails to check the amount carefully. An attacker may use a hidden camera which can read the shop's till display, then his NFC hi-jack device can know TP in advance and so determine a value for PI which the customer will accept. He can therefore block the legitimate message 2 and replace it with his own. The threat from this is similar to that of a camera capturing PIN values. Payment Card Industry Security Standard Council (PCI SSC) prohibits use of cameras near a PIN entering device to avoid monitoring of displays, PIN pads, etc., [14]. Moreover, there are two methods to block RF communication on the NFC link and neither of the methods can easily be adopted in our scenario. The first is to cover the transmitter or receiver with some shielding material. The other method is to produce a high noise on the same operating frequency resulting in a significant decrease in the signal-to-noise ratio. For the former the attacker must shield the MD or the shop terminal. The latter requires noise generating hardware in close proximity to the MD and the shop sales terminal. Both approaches are visibly detectable. This means there is little scope for a successful attack when the MD also verifies the authenticity of message 16. It should therefore be an acceptably small risk.

App_{ID}, which is sent in the clear over the NFC link, is a random string generated by the credit approval authority. From an attacker's perspective, its only significance is its assurance that the customer had, at least before the transaction, the amount TP in his account. This assurance can also be achieved if a customer successfully pays for some goods. Therefore, App_{ID} is not sensitive information in this scenario.

 ISBN: 978-1-61208-353-7

The Role of the Approval Identifier in message 16. App_{ID} acts as a bridge between phase II and III. It also adds freshness to message 16, so it cannot be replayed in future. Any alteration in the K_{info} results in invalid keys and an invalid App_{ID}. Hence it is detectable.

Non-repudiation of Transaction Execution Messages. TEM_u and TEM_s are digitally signed by the MTD. In case of any dispute over payment, the MTD has to honour both messages. So, both the customer and the shop are completely assured of the transaction payment taking place.

Disclosure of Relevant Information. The Cr_{req} containing price information is not disclosed to the base station or any other GSM entity apart from the MTD. The SBAD is sensitive information. It is encrypted not only over the GSM links but also over the NFC link. It is transmitted through the mobile device to the MTD, yet the former cannot decrypt this information. The Acc_{ID} of the customer is not disclosed to the shop. The MNO does not need to know the shopping details of the customer. Therefore, only the total amount is communicated to the MNO for transaction.

New Keys for every Transaction. The encryption and MAC keys for the message Cr_{req}, namely K_1 and K_2, are freshly generated by the mobile device in each round. Similarly, the keys K_3 and K_4, generated by the shop, are fresh for each transaction. Of course, these should not be predictable, especially if previous such keys become known.

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 [15]. Apart from its cryptographic value, 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.

D. Monetary Transaction Between Two Individuals

The proposed protocol can be used for monetary transactions between two individuals. The payee acts as a shop PoS terminal, and uses his own mobile phone for this. The added advantage in our proposal is that the payee does not need to register himself with the payer's (= customer's) MNO. This eliminates dependency of both parties to be on the same mobile network for monetary transactions. The payee needs only to provide his banking details in step 16 of the protocol.

VII. CONCLUSION

In this paper, we have proposed a transaction protocol for providing a secure and trusted communication channel for payment of goods using mobile devices. The proposed protocol was based on the NFC Cloud Wallet model [16] and the NFC payment application [1] on secure cloud-based NFC transactions. We considered a cloud-based approach for managing sensitive data, ensuring the security of NFC transactions by means of a virtual SE within the cloud environment, as well as considering and simplifying the role of the physical SE within the NFC phone architecture. The operations performed by the vendor's reader, by an NFC enabled phone and by the cloud provider (in this paper the MNO) are detailed. Such operations are possible using current technology as most of the functionality is already implemented to support other mechanisms. We considered the detailed execution of the protocol and showed our protocol performs reliably and securely in a cloud-based NFC transaction architecture. The main advantage of this paper is to demonstrate another payment method for those who do not have bank accounts in the normal sense. This way of making payments eases the process of purchasing as participants only have to top up their MNO account without having to follow all the usual banking procedures.

REFERENCES

[1] P. Pourghomi, M. Saeed, and G. Ghinea, "A Proposed NFC Payment Application," in International Journal of Advanced Computer Science and Applications (IJACSA), vol. 4, no. 8. SAI, 2013, pp. 173–181.

[2] G. Cattaneo, G. Maio, P. Faruolo, and U. Petrillo, "A Review of Security Attacks on the GSM Standard," in Information and Communication Technology, ser. Lecture Notes in Computer Science, vol. 7804. Springer, 2013, pp. 507–512.

[3] P. Pourghomi, M. Saeed, and G. Ghinea, "Trusted Integration of Cloud-based NFC Transaction Players," in 9th International Conference on Information Assurance and Security. IEEE, 2013, pp. 6–12.

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

[5] J. Paillès, C. Gaber, V. Alimi, and M. Pasquet, "Payment and Privacy: A Key for the Development of NFC Mobile," in International Symposium on Collaborative Technologies and Systems. IEEE, 2010, pp. 378–385.

[6] M. Roland, J. Langer, and J. Scharinger, "Applying Relay Attacks to Google Wallet," in The 5th International Workshop on Near Field Communication, Zurich, Switzerland. IEEE, 2013, pp. 1–6.

[7] G. Madlmayr and J. Langer, "Near Field Communication Based Payment System," in Konferenz Mobile und Ubiquitäre Informationssysteme (MMS), Germany, ser. LNI, vol. 123. GI, 2008, pp. 81–93.

[8] "MasterPass," 2013, URL: https://masterpass.com/Wallet/Home [accessed: 2014-06-04].

[9] Sarah Clark, "MasterCard Enters the Mobile Wallet Market," in NFC World, 9 May 2012, URL: http://www.nfcworld.com/2012/05/09/315600/mastercard-enters-the-mobile-wallet-market/ [accessed: 2014-06-04].

[10] 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.

[11] G. P. Hancke, "Practical Eavesdropping and Skimming Attacks on High-Frequency RFID Tokens," in J. Comp. Sec., vol. 19, no. 2. IOS Press, 2011, pp. 259–288.

[12] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones," in IACR Cryptology ePrint Archive, 2011, pp. 618:1–618:16.

[13] T. W. C. Brown and T. Diakos, "On the Design of NFC Antennas for Contactless Payment Applications," in 5th European Conference on Antennas and Propagation (EUCAP). IEEE, April 2011, pp. 44–47.

[14] PIN Transaction Security (PTS) Point of Interaction (POI), Payment Card Industry (PCI) Std. Version 3, June 2013, URL: https://www.pcisecuritystandards.org [accessed: 2014-06-04].

[15] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in Journal of Cryptology, vol. 21, no. 4. Springer, 2008, pp. 469–491.

[16] 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.

 ISBN: 978-1-61208-353-7