View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Royal Holloway - Pure

2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications

Secure Mobile Payment on NFC-Enabled Mobile Phones Formally Analysed Using CasperFDR

Sarah Abughazalah, Kostantinos Markantonakis, Keith Mayes
Smart Card Centre-Information Security Group (SCC-ISG)
Royal Holloway, University of London
Email: {Sarah.AbuGhazalah.2012, K.Markantonakis, Keith.Mayes}@rhul.ac.uk

***Abstract*—Near Field Communication (NFC) mobile phones can be used as payment devices and can emulate credit cards. Although NFC mobile services promise a fruitful future, several issues have been raised by academics and researchers. Among the main concerns for the use and deployment of NFC-enabled mobile phones is the potential loss of security and privacy. More specifically, mobile phone users involved in a payment transaction conducted over a mobile handset require that such a system does not reveal their identity or any sensitive data. Furthermore, that all entities participating in the transaction are legitimate. To this end, we proposed a protocol that meets the mobile user's requirements. The proposed protocol attempts to address the main security concerns and protects the customer privacy from any third party involved in the transaction. We formally analysed the protocol using CasperFDR and did not find any feasible attacks.**

***Index Terms*—Security protocol, Privacy, NFC-enabled mobile phone, Mobile payment**

I. Introduction

An NFC-enabled mobile phone is increasingly becoming a ubiquitous device allowing access to a wide variety of services other than making calls and sending texts. NFC is a communication link that uses Radio Frequency (RF) signals to exchange data between devices that are normally less than 10 cm [1]. It can be used in mobile payment transactions just by waving the NFC-enabled phone at the point of sale (POS) terminal. The NFC-enabled mobile phone has a secure element, which can be defined as the main element for securely hosting mobile applications and their confidential and cryptographic data. [2].

In this paper we focus on a mobile payment, which can be defined as any payment that involves a mobile phone to initiate, authorise and confirm an exchange of financial value in return for goods and services [3].

A number of mobile payment studies have been proposed in recent years, for example utilising Short Message Service (SMS) [4], bluetooth [5], [6], or NFC technology. However, using SMS for making a mobile payment might be vulnerable to exploits of SMS latency [8]. In addition, using a bluetooth as a communication mode can be compromised by a new attack known as snarfing, which allows the intruder to exploit a security flaw in the wireless protocol [7]. One of the most well-known NFC mobile payment applications among Android devices is Google Wallet. Google Wallet is a container for bank cards, gift cards, reward cards and special offers. The user has to provide a PIN number to unlock the application, choose the payment card and then place the phone on the POS terminal to make the payment. However, Ronald et al. shows that Google Wallet can be vulnerable to relay attack; a detailed attack can be found in [9].

There is less literature related to authenticating all of the parties involved in a shop-based NFC mobile payment transaction, which will be the focus of this paper. Moreover, this study will also concentrate on maintaining privacy of the customer's identity and sensitive data such as bank account.

A simple scenario which this paper is based on is shown in Fig. I. It works as follows: 1- The customer needs to scan the products' tags by using his NFC-enabled mobile phone. 2- The customer presents his phone to the merchant's reader (POS). 3- The reader contacts the issuer bank for making a payment of the purchased products.

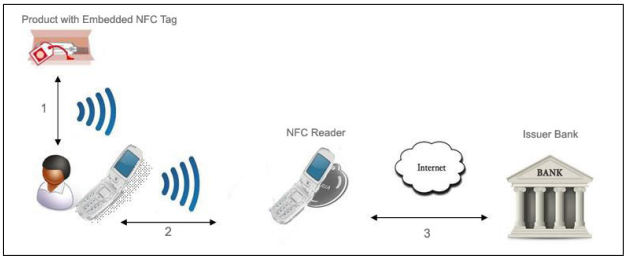

Fig. 1. System model

Mobile phone users involved in a payment transaction conducted over a mobile handset require that such a system does not reveal their identity or any sensitive data. Furthermore, that all entities participating in the transaction are legitimate. To achieve this, the proposed mobile payment in this paper involves three stages:

1) In the first stage, we propose to authenticate the products' tags to the merchant's reader (POS).
2) In the second stage, we propose to provide a mutual authentication between the customer's phone and the merchant's reader.
3) In the third stage, we propose to authenticate the customer, customer's phone and merchant's reader to the bank, and vice versa.

Moreover, to achieve privacy to the customer's sensitive data, our proposed protocol exhibits the following:

- *Customer privacy:* The customers who wish to use their

978-1-4799-6513-7/14 $31.00 © 2014 IEEE
DOI 10.1109/TrustCom.2014.55

IEEE computer society

mobile phones for making a payment, are not required to submit any card details. They just need to know the bank application PIN number and be in possession of the mobile phone. Also, the merchant never receives any bank card data; instead, they receive an encrypted and signed credit card certificate.

We propose to employ a one-time password (OTP) mechanism within the NFC-enabled mobile phone for the authentication purpose. OTP is a single use password, hence an intercepted password will not be reusable. One way to generate an OTP in the NFC-enabled mobile phone is via a smart card such as the Universal Integrated Circuit Card (UICC). An UICC smart card is used as the secure element that provides a tamper-resistant storage of cryptographic keys and performs cryptographic operations [10].

The rest of the paper is structured as follows: Section II introduces the main motivation for proposing our protocol. Section III shows the scenario for making a mobile payment in a shop. In Section IV, the protocol's main goals and requirements are discussed. In Section V, we explain the proposed protocol in detail. In Section VI, we show some diverse scenarios and analyse the proposed protocol with respect to informal analysis. In Section VII, we formally analyse the proposed protocol using CasperFDR. Finally, in Section VIII, we provide the concluding remarks.

II. Motivation

We proposed this scheme mainly to solve the problems with using a smart card that utilises the Europay-Mastercard-Visa (EMV) standards for making a payment. The EMV standards were developed to overcome weaknesses found in the magnetic stripe card and to provide more security to the payment applications [11]. The so-called "EMV Chip and PIN" protects smart card transactions by authenticating the card, cardholder and transaction through a combination of cryptographic functions, and the entry of a PIN [12]. However, it has been found that the security provided by the EMV in the smart card can be vulnerable to bypassing, downgrading or cloning attacks as listed in [13]–[15].

In [13], the authors show that the PIN verification process can be compromised by an attacker, who can use a genuine stolen card to make a payment without knowing the card's PIN, and the transaction will be successful only if the card supports the "offline plaintext PIN cardholder verification" option. This happens because in this mode, the terminal asks the cardholder to enter the PIN, and then the PIN is sent from the terminal to the card in plaintext to find a match with the card's stored PIN.

Moreover, the authors in [14] show that an attacker can downgrade the Cardholder Verification Method (CVM) list from "encrypted PIN cardholder verification" to "plaintext PIN cardholder verification" using a dedicated skimming hardware that can capture the card data and record the entered PIN. Also, an attacker could modify the Issuer Action Code, which affects how the terminal should react in case of an exception. The attacker can change the action code to be "in case of a failed data authentication, authorise the transaction online". Therefore, although the transaction can be detected and denied at the back-end, the PIN will be transferred to the card in plaintext, which can then be intercepted by the attacker.

Furthermore, Bond et al. in [15] show that an attacker can predict the random number generated by the terminal and can resend the authorisation request cryptogram (ARQC) to the issuer to appears as if the card is alive, present, and engaged in the transaction. ARQC is a description of the transaction details, such as transaction amount, currency, type, a nonce generated by the terminal, etc. Due to the predictability of the random number, the attacker could calculate a series of ARQCs and compare them with a real credit card in advance. Later, the attacker can clone a card using these pre-calculated ARQCs to perform actual payment transactions.

One of the main problems that exists with the EMV standard is that the attacker can tamper with the terminal, such as the POS or the ATM, in order to steal the card's PIN or to clone a legitimate card. This motivated us to propose a new protocol that authenticates the terminal to the other parties involved in the communication session before accomplishing subsequent tasks.

Instead of using a physical card for making a payment, a contactless payment system, where the card data is stored in the mobile's secure element, is introduced. One of the most global contactless payment card standards is the EMV Contactless Specifications for Payment Systems [16], which is based on the ISO/IEC14443 standard. However, using a wireless NFC technology may simplify some attacks, such as eavesdropping and relay attacks [9]. For example, Francis et al. [17] showed that NFC-enabled mobile phones can be used as platforms for attacks against ISO/IEC 14443-based smart card systems. The authors showed that an attacker with an NFC phone can perform two attacks, namely token cloning and contactless skimming. In addition, Francis et al. [18] demonstrated that an attacker can cause a relay attack in the peer-to-peer NFC communication mode via installing suitable MIDlets on the attacker's own NFC-enabled phones.

III. Scenario

The proposed approach can be used in a shop, such as a clothes shop, as follows:

1) A group of brands or products share the same NFC tag. This tag can be placed on the shelf edge in front of the products. The customer uses his NFC-enabled mobile phone to scan the product's NFC tag.
2) The phone displays the product's information, such as its ID and price. If the customer wants to buy the product, he clicks on the Approve button, otherwise he clicks on the Cancel button. The customer does the same process with all the products he wants to purchase.
3) If the customer wants to delete an item after approving it, from the list of selected products he just clicks on the unwanted item and clicks on the Delete button.

4) When the customer finishes scanning the products' tags, he places his phone close to the NFC reader (POS).
5) The reader displays the total price to the customer.
6) The customer clicks on the OK button to confirm.
7) The bank application in the phone asks the customer to enter the application's PIN on the mobile phone.
8) The customer enters the bank application's PIN.
9) The customer brings his phone close to the reader.
10) The transaction result is displayed on the reader, stored in the merchant database, and the bill is sent to the mobile phone (in both successful or failed transaction).

IV. GOALS

In this section we consider the main requirements for every party involved in the mobile payment process, namely a bank, a merchant and a customer.

1) The issuer bank requires the following:
   - Customer and mobile phone authentication: The bank needs to authenticate the customer and his mobile phone before debiting his account.
   - Merchant authentication: Before the bank debits the customer's account, it needs to have a proof that a legitimate merchant has requested this transaction.
2) The merchant requires the following:
   - Scanned products authentication: The merchant should authenticate all the products scanned by the customer before proceeding with the transaction.
   - Customer's phone authentication: The merchant needs to ensure that it deals with a mobile phone belongs to a legitimate customer, not an attacker impersonating the customer.
   - Bank authentication: The issuer bank should authenticate itself to the merchant, so that the merchant knows it is dealing with a legitimate bank.
   - Transaction authentication: The merchant should receive a confirmation from the bank about the status of the transaction.
3) The customer requires the following:
   - Customer data privacy: The protocol should protect the customer's identity, such as the phone number and/or sensitive data, such as the customer's bank account from being revealed by the merchant or intruders.
   - Bank authentication: The customer's phone should confirm that it is dealing with a legitimate issuer bank, who issued the stored customer's card.
   - Merchant authentication: The customer's phone should confirm that it is dealing with a legitimate merchant's reader, not a spoofed reader.
   - Transaction authentication: The customer should receive a bill which shows if the transaction is successful or not.

V. THE PROPOSED PROTOCOL

In this section, we explain the proposed protocol in detail.

A. Notation

T_i: Denotes the i^{th} product's tag
P: Denotes a mobile phone
M: Denotes a merchant
B: Denotes an issuer bank
AC: Denotes an acquiring bank
K_{Px}: A public key of entity x of L bits (e.g. L=128 bits), used for encryption
K_{Sx}: A private key of entity x of L bits (e.g. L=128 bits), used for decryption
K_{SGx}: A private key of entity x of L bits (e.g. L=128 bits), used for signing the data
K_{xy}: A shared session secret key between two entities x and y of L bits (e.g. L=128 bits)
$E_{K_{xy}}(M)$: A message M encrypted with a shared session secret key (symmetric encryption)
$E_{K_{Px}}(M)$: A message M encrypted with a public key (asymmetric encryption)
$Sig_{K_{SGx}}(Z)$: A signature on data Z, signed using x private key
MSISDN: The Mobile Subscriber Integrated Services Digital Network Number (mobile phone number)
$Cert_x$: A digital certificate of an entity x, i.e. $Cert_x=Sig_{K_{SGCA}}(K_{Px}$, expiry date, $ID_x)$, where K_{SGCA} is the Certification Authority private key.
Card-Cer_P: A bank card digital certificate calculated by the issuer bank, i.e. Card-$Cer_P=E_{K_{PP}}(Sig_{K_{SGB}}$(customer name, account number, MSISDN, K_{PP}, expiry date))
HPIN: The result of applying hash function on the mobile phone banking application PIN number
TMSI: The Temporary Mobile Subscriber Identity
ID_x: The ID that uniquely identifies entity x
R_x: A fresh random number generated by entity x
H(Z) : The result of generating a hash of data Z, where H: $\{0,1\}^* \rightarrow \{0,1\}^L$
d: The number of scanned products
n: The number of products in the shop
Tprice: The total price of the purchased items
TREF: The transaction reference generated by the merchant

B. Assumption

In designing the proposed protocol we assume the following:

- The issuer bank calculates Card-Cer_P for a customer's mobile phone and stores it in the bank.
- The customer installed the mobile phone banking application. The application is stored in the secure element. The customer cannot tamper with the application as it is installed in the secure element, which is isolated from the phone's main operating system and hardware.
- Only the legitimate issuer bank can access the mobile phone banking application and store Card-Cer_P and ID_B via the Mobile Network Operator (MNO), which provides the bank with a unique security key to manage its application within the secure element.

- The mobile phone banking application asks the customer during installation to enter a new PIN for the application. Once entered, the UICC calculates the hash of PIN denoted as (HPIN) and stores it in the secure element. The UICC calculates Password=$E_{K_{PB}}(Sig_{K_{SGP}}(HPIN))$ using bank public key (K_{PB}) and phone's private key used for signature (K_{SGP}), and transmits it to the issuer bank along with phone's certificate ($Cert_P$) via the MNO. The bank decrypts Password using the bank's private key, verifies the signature using the phone's public key and stores (HPIN) and $Cert_P$ associated with the customer's stored data.
- The communication channel between the merchant and issuer bank is secure (for example using HTTPs).
- The mobile phone communicates with the product's tag and the merchant's reader through a wireless channel that is vulnerable to attacks.
- The product's NFC tags in the proposed scheme is applicable to NFC Type-4 tags that supports ISO 7816-4 and therefore contain a cryptographic processor, which can compute symmetric key encryption [19].
- The i^{th} product's NFC tag (T_i) is encoded by the merchant with some data, such as ID_M, ID_{T_i} and $price_i$. These data are also stored in the merchant database.
- The i^{th} product's NFC tag (T_i) also stores a shared session secret key with the merchant (M) denoted as (K_{T_iM}).
- The issuer bank contacts the acquiring bank via existing architectures and is generally considered to be secure.
- The best practices should be used to ensure that the mobile phone is a trustworthy device; for instance, take the advantages of using a trusted secure element and a secure processor.
- Each product embeds a Radio Frequency IDentification (RFID) UHF tag as shown in Section VI.

C. Protocol Description

The proposed protocol contains three stages, namely issuing phase, authentication phase and payment phase.

- *Issuing Phase*

  This stage occurs when the customer scans the products he wants to buy through the NFC-enabled phone, which acts as a reader. This stage is shown in Table I. It involves the following steps:

  1) The customer should make sure that his device has the NFC function turned ON. The customer scans the product's NFC tag with his mobile phone. The product's tag sends the price to the phone.
  2) The mobile phone displays the price to the customer. If the customer wants to buy the product, he clicks on the Approve button, otherwise he clicks on the Cancel button.
  3) If the customer clicked the Approve button, the secure element generates a random number R_{T_i}.
  4) The phone sends R_{T_i} to the tag.
  5) The i^{th} tag calculates $D_i = E_{K_{T_iM}}(ID_{T_i}, ID_M, R_{T_i})$.
  6) The tag sends D_i, ID_{T_i}, and $price_i$ to the phone.
  7) The phone displays the product's ID and price to the customer. Simultaneously, the phone stores D_i, $price_i$ and R_{T_i} in the secure element to be used in the authentication phase.

  This process is repeated for each product the customer wishes to buy. Finally, the secure element calculates the total price (Tprice) of the scanned products and stores it.

- *Authentication Phase*

  Given a successful scanning process from the previous phase, the merchant's reader needs to authenticate the products' tags and mutually authenticate the customer's phone. This phase is shown in Table II. It involves the following steps:

  1) The customer presents his phone close to the merchant's reader (POS).
  2) The phone sends its certificate ($Cert_P$) along with the TMSI to the reader. TMSI is a temporary identification number to ensure the privacy of the mobile subscriber [20]. It remains fixed until the device connects to a base station controlled by a different Visitor Location Register (VLR). TMSI serves as a temporary phone ID.
  3) The reader verifies the phone's certificate ($Cert_P$).
  4) The reader obtains the phone's public key (K_{PP}) from $Cert_P$.
  5) There is no shared secret between the reader and the mobile device until this stage. Thus, the reader generates a fresh random number as a secret session key between the reader and the phone, denoted as (K_{MP}). This key is used for subsequent encryption between the merchant's reader and the phone.
  6) The reader signs the session key with the merchant's signature private key, and then encrypts the signature with the phone's public key, i.e. $M1=E_{K_{PP}}(Sig_{K_{SGM}}(K_{MP}, ID_M, TMSI))$. The inclusion of ID_M and TMSI ensures that both principals have knowledge of each other's identity.
  7) The reader sends M1, and ID_M to the phone along with the merchant's certificate ($Cert_M$).
  8) The secure element verifies the merchant's certificate ($Cert_M$).
  9) The secure element obtains the merchant's public key (K_{PM}) from $Cert_M$.
  10) The secure element decrypts M1 using its secret key (K_{SP}). Then, it checks the signature using the obtained merchant's public key from $Cert_M$ to confirm that the message comes from a legitimate merchant. Then, the secure element verifies the authenticity of the the data received from the reader, such as ID_M and TMSI. Moreover, the secure element verifies

TABLE I
ISSUING PHASE

Mobile phone		NFC product
1- Scan the product		
2- Display the price		
3- If Approve, generate R_{T_i}		
	4-R_{T_i} $\longrightarrow$	
		5- Calculate D_i =$E_{K_{T_iM}}$(ID_{T_i}, ID_M, R_{T_i})
	6-D_i,ID_{T_i},$price_i$ $\longleftarrow$	
7- Display ID_{T_i} and $price_i$, and store D_i and R_{T_i} in UICC		

that the session secret key (K_{MP}) is fresh if the TMSI has not been changed in order to prevent replay attack. If this process is successful, the secure element authenticates the reader.

11) The secure element sends all the data received from the products' tags to the reader to be authenticated by calculating M2=$E_{K_{MP}}$($Sig_{K_{SGP}}$(ID_M, ID_B, TMSI, D_1 ... D_d)) using the secure element's signature private key, and the shared session key. The inclusion of ID_B is to inform the reader about the identity of the issuer bank.
12) The phone sends M2 and R_{T_1} ... R_{T_d} (from Step 3 in the issuing phase) to the reader.
13) The reader decrypts M2 using the shared session key, then checks the signature using the obtained phone's public key to confirm that the message comes from a legitimate phone. If this process is successful, the reader authenticates the mobile phone.
14) Now, the reader attempts to authenticate the products' tags by decrypting D_1 ... D_d using the shared secret keys (K_{T_1M} ... K_{T_dM}) previously known to the reader. Then. the reader compares the data within D_1 ... D_d with the data previously stored in the database during the encoding phase. If this process is successful, the reader authenticates the scanned products.
15) The reader retrieves the price values from the database associated with the received ID_1 ... ID_d, and then calculates the total price (Tprice).
16) The reader sends Tprice to the customer's phone.
17) The secure element compares the received value of Tprice from the reader with the stored value of Tprice calculated by the phone in the issuing phase. If there is a match, the phone displays the total price value to the customer.
18) The customer reviews the price and clicks the OK button to confirm. If the customer wants to cancel a purchased item, he goes back to the products list on the phone and presses the Delete button. The phone recalculates M2 by omitting the deleted item (D_i) and resends it to the reader.

- *Payment Phase*

If the authentication between the merchant, mobile phone and products' tags is achieved in the previous phase, the customer phone is now ready to perform a mobile payment process. This phase is shown in Table III. It involves the following:

1) Based on the received value of ID_B, the reader starts the connection with the issuer bank. The reader generates a transaction reference number (TREF) that should be unique for each transaction, then calculates M3=$E_{K_{PB}}$($Sig_{K_{SGM}}$(TREF, Tprice, ID_M, ID_B, $Cert_P$)). The inclusion of ID_M and ID_B ensures that both principals have knowledge of each other's identity.
2) The reader sends $Cert_M$, and M3 to the issuer bank via the Internet.
3) The bank verifies the merchant's certificate ($Cert_M$) and obtains the merchant's public key (K_{PM}) and ID_M.
4) The bank decrypts M3 using its secret key (K_{SB}), then checks the signature using the obtained merchant's public key from $Cert_M$ to confirm that the message comes from a legitimate merchant. Then, the bank verifies the authenticity of the data received from the reader, such as ID_M and ID_B. The bank checks that TREF is unique and has never been received before from this merchant. If this process is successful, the bank authenticates the reader and stores TREF and Tprice.
5) The bank compares the received phone's certificate ($Cert_P$) with the stored certificate and obtains the phone's public key (K_{PP}).
6) The bank generates a fresh random number as a challenge between the bank and the phone to be used in the calculation of the session key.
7) The bank signs the challenge with its private key, and then encrypts the signature with the phone's public key, i.e. M4=$E_{K_{PP}}$($Sig_{K_{SGB}}$(ID_B, challenge).
8) The bank calculates M5= $E_{K_{PM}}$($Sig_{K_{SGB}}$(ID_B, ID_M, TREF) to be authenticated by the merchant.
9) The bank sends M4, M5 and ID_B to the merchant.
10) The merchant decrypts M5 using its private key (K_{SM}) and verifies the bank's signature using the bank's public key (K_{PB}). The merchant verifies

TABLE II
AUTHENTICATION PHASE

Mobile phone		NFC reader
1- The phone is presented to the reader		
	2-Cert$_P$,TMSI $\longrightarrow$	
		3- Verify Cert$_P$ 4- Obtain K$_{PP}$ from Cert$_P$ 5- Generate K$_{MP}$ 6- Calculate M1=E$_{K_{PP}}$(Sig$_{K_{SGM}}$(K$_{MP}$, ID$_M$, TMSI))
	7-M1,ID$_M$,Cert$_M$ $\longleftarrow$	
8- Verify Cert$_M$ 9- Obtain K$_{PM}$ from Cert$_M$ 10-Decrypt M1, check the signature and verify the authenticity of the the received data 11- Calculate M2=E$_{K_{MP}}$(Sig$_{K_{SGP}}$(ID$_M$, ID$_B$, TMSI, D$_1$ ... D$_d$))		
	12-M2,R$_{T_1}$ ... R$_{T_d}$ $\longrightarrow$	
		13- Decrypt M2, then check the signature 14- Decrypt D$_1$ ... D$_d$, verify the signatures, then verify the authenticity of the the received data 15- Retrieve the price values from the database, then calculate Tprice
	16-Tprice $\longleftarrow$	
17- Compare the received value of Tprice with the stored value of Tprice 18- The customer reviews the total price and clicks the OK button to confirm		

the authenticity of the data (ID$_B$, ID$_M$, TREF) within M5. If this process is successful, the reader authenticates the bank.

11) The merchant sends M4 and Cert$_B$ to the phone.
12) The secure element verifies the bank's certificate (Cert$_B$) and retrieves the bank's public key (K$_{PB}$) and ID$_B$.
13) The secure element decrypts M4 using its private key, then checks the signature using the obtained bank's public key from Cert$_B$ to confirm that the message comes from the bank. The secure element authenticates the data (ID$_B$) within the signature with the stored one. If this process is successful, the phone authenticates the bank.Then, the secure element retrieves the challenge.
14) The phone asks the customer to enter the payment application PIN on the mobile phone. The customer enters the PIN on the payment application in the mobile phone, then clicks on the OK button.
15) The customer brings his phone close to the reader.
16) The secure element calculates the hash of the entered PIN and verifies it with the HPIN stored in the secure element. If the HPIN is verified, the secure element generates a one-time password, which serves as a session secret key, i.e. K$_{BP}$=H(HPIN $\oplus$ challenge). This step is necessary to authenticate the customer to the bank.
17) The secure element calculates M6=E$_{K_{BP}}$(Sig$_{K_{SGP}}$(MSISDN, ID$_B$, Tprice, Card-Cer$_P$)). The inclusion of both the ID$_B$ and the MSISDN ensures that both principals have knowledge of each other's identity. The calculation of M6 is necessary to authenticate the customer via the entered PIN and to verify the signature of the phone and hence authenticate them.
18) The phone sends M6 and TMSI to the merchant's reader.
19) The reader sends M6 and TMSI to the bank.
20) The bank retrieves the customer's HPIN from the database (which should be associated with the phone's certificate (Cert$_P$)) and calculates the session key, i.e. K$_{BP}$=H(HPIN $\oplus$ challenge), then decrypts M6 using (K$_{BP}$). If the bank successfully generated the correct session key, verified the signature, and authenticated the data, such as MSISDN and ID$_B$, then the bank authenticates the customer and phone. Later, the bank compares the Tprice values it received from the merchant and phone, and if there is a match, the bank withdraws the total price from the customer's account.
21) To inform the merchant that the transaction is completed, the bank calculates M7=E$_{K_{PM}}$(Sig$_{K_{SGB}}$(ID$_M$, ID$_B$, TMSI, Transaction-result)) i.e. Transaction-result= (Tprice, TREF, Transaction-status). Transaction-result is stored in the bank for further acknowledgment.
22) The bank sends M7 to the merchant reader.
23) The merchant decrypts M7 using its secret key (K$_{SM}$) and verifies the bank's signature. The merchant verifies the authenticity of the data within M7.
24) The reader displays Tprice and Transaction-status on the screen.
25) The reader sends the Bill to the phone.
26) The phone displays the Bill to the customer.

TABLE IV
DATA STORAGE

Product Tag	NFC Mobile Phone	Merchant	Bank
K_{T_iM}, ID_M, ID_{T_i}, $Price_i$	K_{SP}, K_{PP}, $Cert_P$, MSISDN, TMSI, Card-Cer_P, ID_B, HPIN	K_{SM}, K_{PM}, K_{T_iM} ... K_{T_nM}, $Cert_M$, ID_M, ID_1 ... ID_n, $price_1$...$price_n$	K_{SB}, K_{PB}, $Cert_B$, ID_B, Card-Cer_P, $Cert_P$, MSISDN, HPIN

27) The final step is to contact the acquiring bank. The merchant sends an authorisation request, i.e. M8= $E_{K_{PAC}}(Sig_{K_{SGM}}(ID_M, ID_{AC}, Transaction\text{-}result))$, to the acquiring bank. Once the acquiring bank authorises the merchant, it contacts the issuer bank and sends an authorisation request along with the Transaction-result. The authorisation and authentication between the issuer and acquiring banks is beyond the scope of this paper.
28) For further security, once the transaction is successful, the RFID reader should disable the RFID tags attached to each purchased product that are placed on the basket contactlessly by sending the Kill command as shown in the Specification [21]. The Kill command length should be long enough to prevent a brute force attack (for example, 128 bits). The details of this process can be found in [21].

The data stored in each party is shown in Table IV:

VI. INFORMAL ANALYSIS OF THE PROTOCOL

The protocol has been designed with respect to a number of possible attacks which are examined in this section.

1) *Stolen mobile phone:* If the customer presents a stolen mobile phone, the mobile phone banking application PIN would prevent this from being useful as the attacker would not know the application PIN. The attacker will not be able to obtain the application PIN during messages exchanged as the PIN is not transmitted in clear. Moreover, the bank application PIN is not stored in the secure element, only the hash of the PIN (HPIN) is stored in the secure element.
2) *Observed response:* An attacker may attempt to eavesdrop on a session between the mobile phone and merchant's reader and capture the exchanged messages. Then, in the next session, the attacker's phone resends the captured messages to charge purchases to the victim's account for example. However, the OTP mechanism for generating the session key between the phone and merchant (K_{MP}) and between the phone and bank (K_{BP}) is used within only one session, and they cannot be reused when the session ends. Best practices must be used for generating the session keys.
3) *Scanning a cheap product's tag instead of an expensive product's tag:* An attacker may choose an expensive item but scan a tag attached to a cheap item. However, the RFID reader placed near the gate should detect such an attack as the expensive item's RFID tag has not been disabled by the RFID reader, thus causing alarm. An attacker will not be able to disable the RFID tag as he needs to send Kill command which depends on a kill password only known by the reader and RFID tag.
4) *Fault scanning:* An attacker might attempt to purchase two identical items but scan the NFC tag only once to pay for only one item. However, the gate's reader will detect that there is an extra item that has not been disabled by the RFID reader.

Moreover, the proposed protocol meets the main requirements discussed in Section IV as follows:

- *Customer privacy:* The privacy of the customer is provided by not sending his identity, such as mobile phone number (MSISDN) or his bank account data, to a merchant. The merchant only receives the TMSI, which is a temporary number associated with the mobile phone. Moreover, the customer's bank account certificate cannot be disclosed to the merchant, as it is encrypted by the session key (K_{BP}) only known to the phone and bank.
- *Unilateral authentication between merchant and product:* Each tag associated with a group of products is programmed to encrypt its data using the shared session key between the tag and reader (K_{T_iM}), thus only a legal reader with the shared key can decrypt the tag's data.
- *Mutual authentication between the mobile phone and merchant's reader:* The merchant authenticates the mobile phone, and vice versa, via three entities:
  - Entity digital certificate ($Cert_p$, $Cert_M$)
  - Shared session key (K_{MP})
  - Entity digital signature (K_{SGx})
- *Mutual authentication between the merchant's reader and bank:* The merchant and bank are mutually authenticated via two entities:
  - Entity digital certificate ($Cert_M$, $Cert_B$)
  - Entity digital signature (K_{SGx})
- *Mutual authentication between the mobile phone and bank:* The phone and bank are mutually authenticated via three entities:
  - Entity digital certificate ($Cert_p$, $Cert_B$)
  - Shared session key (K_{BP}), which is based on the customer's HPIN and bank's challenge.
  - Entity digital signature (K_{SGx})
- *Mobile phone holder's authorisation:* In our protocol the customer is authorised for making a mobile payment by entering the bank's application PIN, which is checked by the phone's secure element and the bank database.
- *Merchant's reader impersonation attack:* The attacker (M') might be able to impersonate a legitimate reader, generate a session key ($K_{M'P}$) and send a forged message to the phone ex. M1'=$E_{K_{PP}}(Sig_{K_{SM'}}(K_{M'P}, ID_M, TMSI))$ with the real reader certificate ($Cert_M$). However, the secure element will detect such an attack by verifying the signature of the data as they are signed by the attacker not by the legitimate reader.
- *Man in the middle attack:* It is essentially difficult for an attacker operating between the sender and receiver to manipulate the exchanged messages, add new messages,

TABLE III
PAYMENT PHASE

Mobile phone	NFC reader	Issuer bank
	1- Generate TREF, then calculate M3=$E_{K_{PB}}$(Sig$_{K_{SGM}}$(TREF, Tprice, ID_M, ID_B, Cert$_P$))	
	2-Cert$_M$,M3 →	
		3- Decrypt M3 4- Generate a challenge 5- Calculate M4=$E_{K_{PP}}$(Sig$_{K_{SGB}}$(ID_B, challenge) 6- Calculate M5= $E_{K_{PM}}$(Sig$_{K_{SGB}}$(ID_B, ID_M, TREF)
	7- M4,M5,ID_B ←	
	8- Decrypt M5	
9-M4,Cert$_B$ ←		
10- Decrypt M4 11- The customer enters the PIN on the mobile phone, then clicks on the OK button 12- The customer brings his phone close to the reader 13- Calculate HPIN, and calculate K_{BP}=H(PIN $\oplus$ challenge) 14- Calculate M6=$E_{K_{BP}}$(Sig$_{K_{SGP}}$(MSISDN, ID_B, Tprice, Card-Cer$_P$))		
15-M6, TMSI →		
	16-M6, TMSI →	
		17- Calculate K_{BP}=H(HPIN $\oplus$ challenge), then decrypt M6 18- Calculate M7=$E_{K_{PM}}$(Sig$_{K_{SGB}}$(ID_M, ID_B, TMSI, Transaction-result))
	19-M7 ←	
	20- Decrypt M7 21- Display the transaction results on the terminal	
23-Bill ←	22- Prepare the Bill	
24- Display the Bill to the customer		

or modify legitimate messages due to the close distance between the sender and receiver [22]. Also, all the exchanged messages are signed and encrypted, which create no meaning to the attacker.

- *Relay attack:* According to [9], there are some potential solutions to avoid relay attacks, such as verifying the PIN within the application in the secure element and displaying the total amount to the customer for confirmation. In our protocol, the PIN is verified within the bank application installed on the secure element. Hence, the attacker will need to know the customer's PIN to conduct a successful relay attack. Also, the customer's mobile phone shows the total price of the transaction and allows the customers to detect if they are being charged for more than expected.
- *Replay attack:* The attacker will not be able to resend the previously exchanged messages as they incorporates a fresh random number (R_{T_i}) or session keys such as K_{MP} and K_{BP} that can be used only once. Moreover, there are some data for which their values will be changed after each session, such as TMSI, TREF, challenge and Transaction-result.
- *Predicting the random numbers*: The proposed scheme should deploy best practices to generate random numbers that are difficult to predict, such us utilising a hardware random number generator.
- *Non-repudiation:* In this paper, a digital signature is used to ensure that a message has been electronically signed by an entity x, and to ensure that entity x cannot later deny that he created the signature.
- *Usability:* The proposed solution suggests that the user only needs his phone to make the payment. Mobile phones are ubiquitous devices that can be hardly considered a burden to carry. The user has only to touch his phone to the reader and enter the application's PIN, which is rather easy and coherent from the user's point

of view.

VII. Formal analysis using CasperFDR

Gavin Lowe has developed the CasperFDR tool [23], a tool for checking the soundness of the protocol based on the specified requirements. It takes a high-level description of the protocol together with its security requirements and produces a Communicating Sequential Process (CSP) code checked and verified by Failure-Divergence Refinement (FDR).

To verify the protocol formally and provide indicative result, we prepared a CasperFDR input file, but due to the page limit we show only the payment phase protocol in Appendix A. CasperFDR is used to verify the authentication and secrecy requirements of the protocol. The main data, keys and functions are defined in the #Free variables Section. In the #Protocol description Section, we described the main steps for sending and receiving messages between the the phone, merchant's reader and bank. At the beginning of the #Protocol description Section, we assume that all the parties contact the Certification Authority server (s) to retrieve the digital certificate of each entity, namely digA, digB, and digM.

The requirements are defined in the #Specification Section. One form of authentication used in the script is Agreement. Agreement means that if Bob meets the Agreement specification, he confirms that Alice has run the same protocol, and agreed on the exchanged values. For example, Agreement (b,a,[HPIN, Challenge, IDB]) means that the phone (b) is authenticated to the bank (a) and both parties agreed on the data values (HPIN, Challenge, IDB).

Similarly, secrecy is specified using the requirement Secret, which checks whether the intruder could know the secret value at the end of the protocol. For instance, Secret (a, passwd(HPIN,Challenge), [b]) means that the secret session key (passwd(HPIN,Challenge)) should only be known to the bank (a) and phone (b).

According to the script in Appendix A, the following goals are achieved as follow:

- Mutual authentication: Before the merchant sends M3 and digM, it performs Running.m.a.[IDM, IDB, TREF, digB] event, which means the merchant starts a run of the protocol, apparently with the bank agreeing on data. Later, the bank will perform Commit.a.m.[IDM, IDB, TREF, digB] event at the end of its part of the protocol, which means the server has finished the protocol with the merchant agreeing on the received data. Similarly, before the bank sends M4 and M5, it performs Running.a.b.[IDB, HPIN, Challenge, MSISDN] and Running.a.m.[IDB, IDM, TREF] respectively, and when the tag receives M4, it performs Commit.b.a.[IDB, HPIN, Challenge, MSISDN], and the merchant performs Commit.m.a.[IDB, IDM, TREF].
- Tag anonymity is depicted as Claim_Secret.a.b.Challenge, Claim_Secret.a.b.passwd(HPIN,Challenge) and Claim_Secret.a.b.HPIN events, which means that the three values of Challenge, passwd(HPIN,Challenge) and HPIN should be kept secret between the mobile phone and the bank.
- Resistance to replay attack is illustrated as a scenario where the phone is engaging in the protocol twice. The phone firstly runs the protocol with the bank, and the intruder obtains M6. Then, the intruder runs the protocol with the same bank and resends M6 to the bank via the merchant's reader. The bank will not perform the Commit event as it received the same session key. Similarly, if the phone engages in the protocol run by receiving duplicate messages from the intruder or the bank, it will not perform the Running event.

In addition, in the #Intruder information section, the intruder is defined to be Mallory, who can take full control of the session: he can impersonate any entity in the protocol, read the messages transmitted in the network, intercept, analyse, and/or modify messages.

CasperFDR did not find any feasible attacks on the proposed protocol, which means that the proposed protocol successfully authenticates all the entities and at the same time ensures that the secret data are kept secure between the legitimate entities.

VIII. Conclusion

In this paper, we proposed to use an NFC-enabled mobile phone for making a secure mobile payment that tackles the issues such as card cloning, skimming, downgrading the terminal and relay attacks. The proposed work focuses on mutually authenticating the customer, mobile phone, merchant's reader and bank before proceeding with the secure payment. The protocol uses the OTP to generate secret session keys that encrypt the customer's data; hence preserve tag's data privacy and provide mutual authentication. We presented an initial informal analysis and we formally analysed the protocol to provide some indicative results. Based on the results of the analysis, we found that it relatively meets the security and privacy requirements that shape the customer and organisation demands. We are planning to implement the proposed protocol and verify its privacy and security features using another formal analysis tool such as Automated Validation of Internet Security Protocols and Applications (AVISPA).

IX. Acknowledgment

This research was supported by the Ministry of Higher Education and King Khaled University in Saudi Arabia.

References

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[2] "NFC Forum Technical Specifications." [Online]. Available: http://members.nfc-forum.org/specs/spec_list/

[3] Y. Au and R. Kauffman, "The economics of mobile payments: Understanding stakeholder issues for an emerging financial technology application," *Electronic Commerce Research and Applications*, vol. 7, no. 2, pp. 141–164, 2008.

[4] H. Harb, H. Farahat, and M. Ezz, "SecureSMSPay: Secure SMS mobile payment model," in *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, Aug 2008, pp. 11–17.

[5] S. Pradhan, E. Lawrence, and A. Zmijewska, "Bluetooth as an enabling technology in mobile transactions," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2, April 2005, pp. 53–58 Vol. 2.

[6] S. Manvi, L. Bhajantri, and M. Vijayakumar, "Secure mobile payment system in wireless environment," in *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*, April 2009, pp. 31–35.

[7] M. Alzomai and A. Jsang, "The mobile phone as a multi OTP device using trusted computing," in *Network and System Security (NSS), 2010 4th International Conference on*, Sept 2010, pp. 75–82.

[8] W.-D. Chen, M. Keith, Y.-H. Lien, and J.-H. Chiu, "Nfc mobile payment with citizen digital certificate," in *Next Generation Information Technology (ICNIT), 2011 The 2nd International Conference on*, June 2011, pp. 120–126.

[9] M. Roland, J. Langer, and J. Scharinger, "Applying relay attacks to Google Wallet," in *Near Field Communication (NFC), 2013 5th International Workshop on*. IEEE, 2013, pp. 1–6.

[10] "UICC-terminal interface; physical and logical characteristics," June 2009, technical Report. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102200_102299/102221/08.02.00_60/ts_102221v080200p.pdf

[11] "EMV specifications," Online. [Online]. Available: http://www.emvco.com/specifications.aspx

[12] K. Mayes and K. Markantonakis, *Smart cards, Tokens, Security and Applications*. Springer, 2008, iSBN: 978-0-387-72197-2.

[13] S. Murdoch, S. Drimer, R. Anderson, and M. Bond, "Chip and PIN is broken," in *IEEE Symposium on Security and Privacy*, 2010, pp. 433–446.

[14] A. Barisani, D. Bianco, A. Laurie, and Z. Franken, "Chip & PIN is definitely broken," in *Presentation at CanSecWest Applied Security Conference, Vancouver*, 2011.

[15] M. Bond, O. Choudary, S. Murdoch, S. Skorobogatov, and R. Anderson, "Chip and skim: cloning EMV cards with the pre-play attack," *CoRR*, vol. abs/1209.2531, 2012.

[16] "EMV contactless." [Online]. Available: http://www.emvco.com/specifications.aspx?id=21

[17] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Potential misuse of NFC enabled mobile phones with embedded security elements as contactless attack platforms," in *ICITST*, 2009, pp. 1–8.

[18] L. Francis, K. Mayes, G. Hancke, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *Radio Frequency Identification: Security and Privacy Issues*, ser. Lecture Notes in Computer Science, S. Ors Yalcin, Ed. Springer Berlin Heidelberg, 2010, vol. 6370, pp. 35–49. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16822-2_4

[19] *Identification Cards Integrated Circuit Cards*, International Standard Organization / International Electrotechnical Commission Std Std., 2005.

[20] "ETSI-GSM Technical Specification," Tech. Rep., Feburary 1992. [Online]. Available: http://www.etsi.org/deliver/etsi_gts/03/0320/03.03.02_60/gsmts_0320sv030302p.pdf

[21] "EPC radio frequency identification protocols Class 1 Generation 2 UHF RFID." [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_0_9-standard-20050126.pdf

[22] C. Adams, "Have money, will travel: A brief survey of the mobile payments landscape," School of Electrical Engineering and Computer Science, University of Ottawa, Tech. Rep., June 2013.

[23] G. Lowe, "Casper: A compiler for the analysis of security protocols," in *Computer Security Foundations Workshop, 1997. Proceedings., 10th*. IEEE, 1997, pp. 18–30.

APPENDIX A
APPENDIX A: CASPERFDR SCRIPT

```
-- NFC PAYMENT PHASE PROTOCOL
-- Not all the data are included in the script
 for simplicity
#Free variables
a, b, m : Agent
-- a= bank  b= phone  m=merchant
s : Server
Challenge, TREF : Nonce
IDB, IDM, HPIN, MSISDN : Data
pk : Agent -> PubKey
sk : Agent -> SecKey
pks : ServerPublicKey
sks : ServerSecretKey
passwd : Data x Nonce -> Password
InverseKeys = (passwd, passwd), (pk, sk),
(pks, sks)
#Processes
BANK(a,s,m, b, IDB, Challenge, pks, HPIN,
IDM, MSISDN) knows sk(a), pk(a), pk, passwd(HPIN,
Challenge)
PHONE(b, s, a, IDB, pks, HPIN, MSISDN) knows
 sk(b), pk(b), pk, passwd(HPIN,Challenge)
SERVER(s, a, b, m, pks, sks) knows pk(a),
pk(b), pk(m)
MERCHANT(m , s, a, IDB, IDM, TREF)
knows pk(m), sk(m), pk
#Protocol description
0.      -> m : a
1a.  s -> a : {a, pk(a)}{sks} % digA
1b.  s -> b : {b, pk(b)}{sks} % digB
1c.  s -> m : {a, pk(a)}{sks} % digA,
           {b, pk(b)}{sks} % digB,
           {m, pk(m)}{sks} % digM
2a.  m -> a : digM % {m, pk(m)}{sks}
2b.  m -> a:  { {IDM, IDB, TREF,
         digB % {b, pk(b)}{sks}}{sk(m)}}{pk(a)} -- M3
3a.  a -> m :{{ IDB,Challenge}{sk(a)}}{pk(b)}% M4 -- M4
3b.  a -> m: { {IDB, IDM, TREF}{sk(a)}}{pk(m)}  -- M5
4.       -> m: b
5a.  m -> b : digA % {a, pk(a)}{sks}
5b.  m -> b:M4%{{ IDB,Challenge}{sk(a)}}{pk(b)}
6.   b -> m : { {IDB, MSISDN}{sk(b)} }{passwd(HPIN,
                    Challenge)}  % M6 -- M6
7.   m -> a:M6%{{IDB}{sk(b)}}{passwd(HPIN,Challenge)}
8.   a -> m : {{IDM, IDB, TREF}{sk(a)}}{pk(m)} -- M7
#Specification
Secret(a, passwd(HPIN,Challenge), [b])
Secret(a, Challenge, [b])
Secret(a, HPIN, [b])
Agreement(b,a,[HPIN, Challenge, IDB, MSISDN])
Agreement(a,m,[IDB, IDM, TREF])
#Actual variables
Phone, Bank, Merchant, Mallory : Agent
Server1: Server
Challenge1, TREF1 : Nonce
IDB1, HPIN1, IDM1, MSISDN1: Data
pks1 : ServerPublicKey
sks1 : ServerSecretKey
InverseKeys = (pks1, sks1)
#Functions
symbolic passwd, pk, sk
#System
BANK(Bank,Server1,Merchant, Phone, IDB1,Challenge1,
       pks1, HPIN1, IDM1, MSISDN1)
PHONE(Phone, Server1, Bank, IDB1, pks1, HPIN1,MSISDN1)
SERVER(Server1, Bank, Phone, Merchant, pks1, sks1)
MERCHANT(Merchant,Server1, Bank, IDB1,IDM1,TREF1)
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Phone,Bank,Merchant,Mallory,
pk(Mallory), sk(Mallory)}
```