A Security Framework Model with Communication Protocol Translator Interface for Enhancing NFC Transactions

Lishoy Francis, Gerhard Hancke, Keith Mayes and Konstantinos Markantonakis

Information Security Group, Smart Card Centre
Royal Holloway University of London

Egham Hill, TW20 0EX, Surrey, United Kingdom
{L.Francis, Gerhard.Hancke, Keith.Mayes, K.Markantonakis}@rhul.ac.uk

Abstract—With the recent technological advances of Near Field Communication (NFC) enabled mobile phones it is now possible to introduce additional transactions of value, including those originating from contact-based security tokens within the existing infrastructure. We propose a low cost security framework including a PKI based security protocol, which can be used to integrate transactions involving external contact-based smart cards, for the purposes of e-identification, e-payment, e-ticketing, and communication services. We then designed and implemented a secure Communications Protocol Translator Interface (CPTI), which allows an NFC enabled mobile phone to access and use, over a contactless interface, any additional smart cards (or secure elements (SE)) which are externally available on a contact based interface and vice-versa. By using CPTI, it is now possible to have communication and interaction between passive security tokens as well as to use external contact based security tokens in the NFC environment, such as a contact based payment smart card.

Keywords—security, near-field-communication, protocol-translator, passive tokens, payment transactions, token, smart card.

I. INTRODUCTION

Mobile based transactions and related commercial opportunities have been rapidly growing over the past decade. For instance, [1] indicates that mobile based transactions in the money market in western Europe alone are set to grow between 4 and 5 billion Euros by 2013.

Traditionally, when a consumer buys merchandise using the mobile phone, he/she places the order via SMS (Short Service Message) or WAP (Wireless Application Protocol) or mobile Internet and pays the merchant post-purchase as part of the phone bill or pre-pays using an e-purse. The transaction is later settled by the merchant’s acquiring bank. These methods of purchase were found to be not so time-efficient, compared to more traditional methods.

The inconvenience of slow transactions has been overcome to a certain extent with the adoption of contactless technology, where the consumer can walk into a store and just need to wave the contactless token at the Point-of-Sale (POS) to complete a purchase. By realising the potential increase in such transactions, the banks and merchants have been encouraging micro-payments enabling fast offline transactions of low value, adding to the convenience of merchant and consumer. The security tokens used in these proximity transactions are typically contactless based smart cards and increasingly mobile phones. In this paper, we use the terms tokens, security tokens, smart cards, secure elements and security elements interchangeably. The mobile technologies have seen evolutionary changes with new capabilities such as NFC (Near Field Communication) [2]. NFC technology brings the user experience, convenience and security of contactless technology to the mobile devices, and is enabling quick transactions and service accesses in our day-to-day lives. The growth in NFC based transactions has been phenomenal and is expected to exceed $30 billion globally by 2012 [3]. However, the contactless security tokens and NFC enabled mobile phones are not expected to replace transactions originating from security tokens based on contact interfaces. These tokens include, payment cards [4], e-identification documents such as e-visa [5], employee ID, e-driving license [6], e-ticketing tokens [7], and Subscriber Identity Applications (SIA) such as SIM [8], USIM [9].

Currently, there exists a problem when applications need to use the contact based smart cards within the NFC infrastructure. For example, high value transactions are only supported by contact based smart cards (Chip & Pin cards). We envisage that the current transaction model (for example, payments, ticketing, access control, and identification) using NFC enabled mobile phones can be enhanced by incorporating contact based security tokens and also, by allowing peer-to-peer (P2P) transactions between passive tokens. With P2P transactions, a consumer could also act as a merchant. This new model would include and enable the participation of contact based passive security tokens, thus supplementing the NFC environment. With the proposed model, offline or online transactions of value can be made, in a convenient and flexible manner.

In this paper, we describe an interface that translates between contactless communication and contact based communication and propose a generic security protocol for cryptographically binding, authenticating and establishing a secure channel between the passive security tokens involved, for instance, the NFC enabled mobile phone and the external
II. BACKGROUND

NFC is a short range and standardised (ISO 18092) [2] wireless communication technology that adds contactless functionality to mobile devices including mobile phones and PDAs (Personal Digital Assistants). Such devices can act both as a “contactless card” (based on its secure element) and as a “contactless reader” and also operate in P2P mode with peer devices. These devices support various contactless communication standards, such as ISO 14443 [10], ISO 15693 [11], FeliCa [12] and Mifare Standard [13]. NFC finds application in ticketing, banking, access control, identification, health care, etc. Further details on the potential of NFC technology can be found in [14]. Figure 1 illustrates the interactions of an NFC enabled mobile phone with internal and external tokens. Using the contactless interface such as ISO 14443, the applications in NFC enabled phones (in the ‘reader’ mode) are able to communicate with external contactless smart cards. The JSR 257 API [15] also allows communication with such external smart cards in different messaging formats, for example, NDEF (NFC Data Exchange Format) and APDUs (Application Protocol Data Unit).

In the literature, one can find many examples of NFC based payment systems such as [16], [17], and [18]. Presently, the external interactions of a NFC device are obviously limited to only contactless smart cards and not to any ISO 7816 T=0/T=1 [19] based smart cards. In general, mobile phones do not have the means to communicate with external smart cards that support contact interfaces, which might be needed to communicate with external legacy tokens, such as currently deployed Chip & Pin cards used in payment transactions.

The NFC driven payment model has the potential to evolve from the traditional payment model (where the consumer pays the merchant for the goods using mobile phone) into a new model where the consumer can also act as a merchant. Similarly, an NFC enabled mobile phone user can take the role of an identity verifier or ticket validator without limiting the type and form-factor of the security token. There are several interesting innovations made on the mobile phone platform that are fast becoming available in the real world. For example, [20] provides a magnetic card reader accessory to the iPhone [21]. Similar examples include, [22] and [23] which provide support for ISO 7816 and ID-1 form-factor based smart card readers in addition to the readers currently available for NFC based smart cards.

Figure 1. Our proposed security framework model showing CPTI and its interaction with security tokens, including a schematic view of the NFC enabled Mobile Phone and relevant APIs.
available (the default plug-in/ID-000 reader for SIM and contactless reader) on a mobile phone. Our contributions in this paper include, a security framework model for using external security tokens in an NFC ecosystem, including a secure channel protocol for securing the transactions, and a protocol translator (contactless to contact and vice-versa) interface for security tokens. The aforementioned also enables secure P2P transactions between passive security tokens. We also present the practical proof of concept implementations, on a PC and on an NFC enabled mobile phone.

III. SECURITY MODEL FOR TRANSACTIONS INVOLVING EXTERNAL SECURE ELEMENTS IN NFC

In this section, we propose a security framework model (as illustrated in Figure 1), which can be used to enhance the NFC ecosystem at a low cost and requiring little additional infrastructure. The proposed security model permits NFC applications to securely interact with externally available smart cards over contact and contactless interfaces. The above capability would enable the NFC applications to perform security functions such as authenticating using electronic identification documents; making payment transactions using contactless and contact based payment cards and peer devices; utilising external authentication functions (for mobile network access) that are only available and implemented on contact based smart cards such as (U)SIM. The notation used in this paper is listed in Table I.

A. Communication Protocol Translator Interface (CPTI)

The proposed Communication Protocol Translator Interface (as illustrated in Figure 2) would allow communication between passive tokens (for example, contactless and contact based SEs). The CPTI runs a translator engine that translates and relays messages between SEs available on multiple communication interfaces. The options available to implement CPTI are as follows, (1) CPTI on a mobile phone, (a) on the phone’s J2ME software layer and/or (b) on the internal SE’s software layer; (2) CPTI on a personal computer with multiple card readers with contactless and contact interfaces; and (3) CPTI on a specialised standalone hardware module (for example, on a Field Programmable Gate Array (FPGA)) interfaced with contactless and contact based card readers. The CPTI is explained in detail in Section V.

B. Security Protocol for Transactions

In order to secure the transactions involved, we propose a public key based protocol (illustrated in Figure 3) where authentication, integrity, and confidentiality, are achieved by the sender signing, \( \text{Signature} = \{M\} \text{Sign}_{\text{Priv}} \), the message \( M \) with its private key, then encrypting the message and signature, \( \{M, \text{Signature}\} \text{Enc}_{\text{Pub}} \), using the recipient’s public key. Non-repudiation is also achieved by the signing process. The secure channel thus established protects the messages between the communicating parties. The
A

\[ \{ID_A, \{ID_A, \text{Cert}_A, \text{Rand}, \text{Loc}_A, T_{s_1}\} \text{Sign}_{\text{Priv}_{K_A}}\}\text{Enc}_{\text{Pub}_{K_B}} \]

Decrypts using \text{Priv}_{K_A}
Verifies \text{Cert}_A \& \text{Loc}_A
Records \text{Rand} \& T_{s_1}

\{(ID_B, \text{Cert}_B, \text{Rand}, \text{Loc}_B, T_{s_1}) \text{Sign}_{\text{Priv}_{K_B}}\}\text{Enc}_{\text{Pub}_{K_A}}

Decrypts to obtain \text{Data}_i\text{, using } \text{Priv}_{K_A}

\{(ID_A)[B, \text{Rand}, \text{Loc}_A][B, T_{s_1}, \text{Data}_i] \text{Sign}_{\text{Priv}_{K_A} \text{Priv}_{K_B}}\}\text{Enc}_{\text{Pub}_{K_A} \text{Pub}_{K_B}}

Decrypts to obtain \text{Data}_i\text{, using } \text{Priv}_{K_B}

Figure 3. Asymmetric, mutually authenticated, secure channel protocol using certificates.

certificate (Cert_A) is issued to the subject, for instance to the entity A, by the Certificate Authority (CA), which is signed by CA’s private key. It contains unique identifier of CA, public key of the subject, and is bound with unique identifier ID_A of the subject. The certificate Cert_A of entity A, is represented as \{ID_A, ID_{CA}, \text{Pub}_{K_{CA}}\}\text{Sign}_{\text{Priv}_{K_{CA}}}, is card verifiable [24], and is issued for a time period. The certificate chain can be verified by checking the corresponding CA’s public key and until the root CA. The security is dependent upon the secrecy of the private key and strength of the underlying asymmetric algorithm. The timestamp T_{s_1}, and random number, Rand, are used in the transaction to resist any replaying [25] attacks. If there was only Rand present then anyone listening to one transaction could potentially masquerade as A. The T_{s_1} introduces a secondary ‘randomness’ to the message exchange. The T_{s_1} would also ensure that the transaction messages occur in a timely manner (e.g. to detect and to resist an attacker from introducing any time-delay for some malicious purpose, such as relaying). The location information is used as an additional metric for non-repudiation of the transaction between two parties. It ensures that there is an audit trail showing both parties were present as we have signature from both parties agreeing that they were at the same position at the same time, as indicated by the T_{s_1}. The use of location information in our protocol is motivated from [26]. Symmetric protocols have been proposed such as [27] may alternatively be used in the above scenario, but the hardware on the available mobile devices and tokens are advanced and is capable of running asymmetric schemes. An asymmetric-symmetric hybrid protocol establishing a secure channel based on symmetric cryptography is described in [28]. Although, one could find other such proposals in the literature for establishing a secure channel, we proposed a full asymmetric protocol for our solution as it allows easy key management in the mobile devices and tokens. We note that the recently proposed mutual authentication and secure channel protocol ‘SCP10’ by Global Platform [29] is not suitable in this environment due to the following reason, H and T (as illustrated in Figure 1) would need to be supporting SCP10, whereas they are operating at the mobile layer and to include the participation of the SE layer, would require significant communication overhead. Furthermore, SCP10 uses asymmetric methods only for mutual authentication and relies on symmetric cryptography for realising a secure channel. The complete asymmetric protocol using card verifiable certificates, in order to secure the communication between H, E, and T as presented in Figure 1, is illustrated in Figure 3.

The protocol starts with A sending to B, ID_A, Cert_A, Rand, Loc_A, and T_{s_1}, signed with its private key, Priv_{K_A}, and encrypted with B’s public key, Pub_{K_B}. Then decrypts the message received using its private key, Priv_{K_B}, and looks up A’s public key (Pub_{K_A}) using ID_A and verifies the certificate of A, Cert_A, and the location information, Loc_A, if present. B then records the random number, Rand, and the timestamp, T_{s_1}. B replies A with message, ID_B, Cert_B, Rand, Loc_B, T_{s_1}, signed with private key, Priv_{K_B}, and encrypted with A’s public key, Pub_{K_A}. Upon receiving this message A decrypts it with private key, Priv_{K_A}, and looks up B’s public key, Pub_{K_B}, using ID_B and verifies the certificate of B, Cert_B, and the location information of B, Loc_B, if present. Now, the mutually authenticated secure channel is established between A and B, and the messages of the underlying application protocol can be encapsulated within the data field, Data_{i}, of the subsequent protocol messages. Thus, A and B can communicate securely with each other using the proposed protocol.

C. Authenticating External Contactless Security Elements

As illustrated in Figure 1, any NFC enabled mobile device is capable of interacting with external contactless targets. These targets include contactless smart cards and peer NFC
mobile devices. The users can utilise these contactless targets to perform transactions over the contactless interface. These transactions include, electronic micro-payments; authentication to services using electronic identification documents such as electronic passports, contactless identification or access control cards. Now let us examine the proposed security model for authenticating NFC enabled mobile devices using external SEs for the purposes such as electronic card payment and electronic card identification over contactless interface.

A contactless transaction, $T_{iS_j}$, would involve $CL_{SM}$ with $Agent_{AID_{SE,i}}$ (refer to E of Figure 1) interacting or exchanging messages with $Agent_{AID_{M,i}}$ (refer to H of Figure 1) and with the $S_i$ hosted on $N_i$ via $CL$. The security of $T_{iS_j}$ is dependent upon the underlying protocol used, for example, EMV [30], or ICAO [31]. Our proposed approach would not require changes to the underlying protocol and our model provides a secure channel between the local entities. In our model, $T_{iS_j}$ is secured by providing security services, of mutual authentication, confidentiality, integrity, non-repudiation and auditability. The mutual authentication, non-repudiation and data integrity between the communicating parties, $Agent_{AID_{SE,i}}$ and $Agent_{AID_{M,i}}$, are achieved by means of the public key based protocol presented in the previous section. The APDU messages are secured by using the same protocol or by establishing secure channel as specified in [29] or optionally, encrypting the data field using encrypted APDUs as specified in [32].

D. Authenticating External Contact Based Security Elements

Using our proposed Communication Protocol Translator Interface (CPTI) which is introduced in this Section III-A and elaborated in Section V, we show how an NFC enabled mobile device is made capable of interacting with external contact based tokens (illustrated in Figure 1). The practical implementation of CPTI on a PC based environment, on a mobile phone, and on a specialised hardware, are discussed later in Section V. The external contact based tokens include smart cards operating on ISO 7816 contact based interface. The users can use these contact based tokens to perform electronic transactions such as macro-payments using EMV [4] based smart cards, and to authenticate to services using electronic identification documents that are available on contact based smart cards. Now let us examine the proposed security model for authenticating NFC enabled mobile devices using external SEs for the purposes such as electronic card payment and electronic card identification over contact based interface.

A contact based transaction, $T_{iS_j}$, would involve $CT_{SM}$ with $Agent_{AID_{SE,i}}$ interacting or exchanging messages with $Agent_{AID_{M,i}}$ and with the $S_i$ hosted on $N_i$ via $CT$. The security of $T_{iS_j}$ is again dependent upon the underlying protocol used. The security model is similar to the one described in Section III-C providing the security services including the cryptographic binding described. The additional features here include an inherent encapsulated security offered by CPTI which acts a secure channel between the communicating parties. This would deter any similar attacks presented in [33].

IV. USE CASES

In order to demonstrate the NFC authentication and transactions with external tokens, we present examples of electronic payment scenarios, identification using electronic documents, SIM based services, and access control (as illustrated in Figure 4). By implementing the proposed framework model within the existing NFC infrastructure, the service providers would be able to grant new privileges to the NFC enabled mobile phone user, such as being a merchant and accept payment cards, to verify identity, to validate tickets, to enable secure P2P transactions, etc. We assume that the service provider is able to deploy applets (agents) [34] and security key pairs, in order to realise end-to-end transaction security using the proposed protocol and CPTI, onto the fielded security tokens and devices. In scenarios where deploying the agents is not fully possible, it is assumed that the security would rely on the strength of the underlying protocol and the operating interface. Our proposed framework provides a model for communication-interface independent transactions between passive tokens and devices. Now let us examine the potential use cases in detail.
A. Payments

The potential payment scenarios (as illustrated in Figure 4) include, exchanging electronic money or vouchers of value between users, and using external payment token with the NFC enabled mobile phone in order to pay for the services offered; and involving passive tokens with contact or contactless interfaces. Now let us say that two users want to exchange money, one of them assumes the role of the ‘merchant’, accepting payment from the other user. With the tokens being a NFC enabled phone in “card” emulation mode or contact/contactless based payment cards, the users have the choice of using CPTI implemented on a PC or on a NFC enabled mobile phone (detailed in Section V). In the ‘untrusted’ PC environment, the transaction is secured with end-to-end security between the NFC enabled mobile phone and the smart card or between two smart cards. In the scenario where only the NFC enabled mobile phone (host) is used, the transaction participants can be, two contact/contactless based payment cards or the host and another payment card. For high value transactions (for instance, an EMV based macro-payment originating from contact based smart card requiring high transaction security), the transactions may need to go online and be authorised. The host can then connect to the online server, using the supported data bearer. The transactions can be recorded and sent to the payment acquirer and settled at a later stage, for example, when the token is presented at the ATM (Automated Teller Machine).

Alternatively, a 3rd user or a 3rd party can assume the role of an arbitrator, and provide CPTI linked via two NFC mobile phones, \( NFC_H \) and \( NFC_T \), using IEEE 802.15 (Bluetooth) interface (using NFC to pair the phones), in order to exchange value between two users with passive tokens. In this scenario, the user presents the payment card \( CT_{SM} \) in \( NFC_T \) which is processed over \( CL_{H,E} \) by the payment application running on \( NFC_H \) interacting with another payment card. In the scenario where the user wants to pay using an external payment token, the CPTI implemented mobile phone processes the contact or contactless payment tokens, over \( CL_{H,E} \) or \( CT_{H,E} \) interfaces. The protocol presented in Figure 3 would establish the secure channel and all transaction messages encapsulated at the APDU level, and guarantees end-to-end security.

B. Identification, Access Control, and E-Ticketing

The users may need to be identified before given access to services. For instance, the merchant may want to identify the payer or vice-versa. The payment application is able to extract the stored personal details such as a picture from the token presented, to help with the verification of identity. However, additional forms of electronic identity may be used, such as an e-passport/e-visa or an e-driving license or an employee identity card. Users with a contactless electronic ID, \( CL_{SM} \), implemented on the token (as illustrated in Figure 4) would be verified over \( CL_{H,E} \) by CPTI, in order to access the desired service. The user with a contact based electronic ID, \( CT_{SM} \), would identify him/her over \( CT_{H,E} \) or over \( CT_{H,T} \) using CPTI. This use case can be extended to a corporate environment, where CPTI would enable the services to accept additional identification tokens without placing any limitation to the communication interface. For instance, an employee may use his/her additional electronic card identification, even those available on contact based interface, to enhance, to distribute or to upgrade the privileges, in a hierarchical privilege based access system.

The CPTI enables the user tokens to interact with external reader tokens such as a SAM (Secure Access Module) [35], [36]. A SAM can be a smart card component which is placed inside the internal contact reader slot of a ticket turnstile or a payment terminal. The SAM identifies the token that is presented to the system, checks its authenticity, and protects the transfer of data, usually by encryption, between the terminal and the back-end system, thus validating the transactions. Using CPTI, the NFC mobile phone can act as a portal between such passive tokens, for example, the token available on the phones’ contact interface is used to authenticate to the SAM.

C. Accessing Services using External SIA (U)SIM

With multiple reader interfaces becoming available, an NFC enabled mobile phone is now able to interact with external (U)SIMs. The (U)SIM implements the security protocols and algorithms necessary to authenticate to UMTS and GSM network access systems.

Presently, the user needs to pay for network services using the conventional methods such as walk into a store to purchase top-up vouchers or send SMS or use website to make an online purchase or make a bank transfer or even pay at end of the month upon receiving the bill. By using CPTI, it is now possible to have applications that would allow the users to pay instantly with the payment token and credit the (U)SIM, across contact or contactless interface (as illustrated in Figure 4). The CPTI thus allows an external payment card to be used with a (U)SIM.

With NFC becoming fast available on advanced mobile phones (for example, Windows For Mobile based smart phones [23]) that can start without an SIA (for example, (U)SIM) being present (or allow the user to choose between SIAs), CPTI finds application in providing access to externally available (U)SIMs. Elaborating this, let us say that \( Agent_{AID_SM} \) implemented on \( NFC_H \) needs to initialise and authenticate to the \( N_{U\text{MTS},SM} \), using the external (U)SIM, \( CT_{SM} \), present on the ISO 7816 T=0/T=1 interface that is implemented with \( Agent_{AID_SM} \). \( CT_{SM} \) can also be present on another mobile phone, \( NFC_T \). Both \( NFC_H \) and \( CT_{SM} \) are pre-loaded with signed certificates \( Cert_H \) and \( Cert_{SM} \); and the key pair \( (Priv_K_H, Pub_K_H) \) and \( (Priv_K_{SM}, Pub_K_{SM}) \) respectively. The initial two message
Figure 5. CPTI implemented on a PC and on a NFC enabled Mobile Phone.

exchanges of the protocol presented in Figure 3 is run to establish a mutually authenticated secure channel. The subsequent protocol messages would encapsulate the APDUs command-response messages needed for the session established between Agent AID M_H and Agent AID S_Ei. These APDU messages are of type SELECT, READ RECORD, READ BINARY, VERIFY PIN, MANAGE CHANNEL, STATUS, etc, comprising the full sequence of (U)SIM initialisation and authentication functions, which are then relayed to the (U)SIM application(s) by Agent AID S_Ei using an inter-applet communication mechanism. Alternatively, the (U)SIM application(s) may integrate and implement the agent (Agent AID S_Ei). The user would then be able to use the resulting authentication for accessing any associated network service. We also note that the CPTI would also be useful in any office or corporate environment, for example, the company would be in a position to offer a dedicated (U)SIM for its employees to make business calls and the private calls would use an external (U)SIM accessed via CPTI.

V. PRACTICAL PROOF OF CONCEPT IMPLEMENTATION

There already exist implementations in which an SIA is interfaced with contactless technology via the mobile phone (for example, (U)SIM with contactless interface) such as Giesecke & Devrient UniverSIM Proximity Card, and modified mobile phone with an extended antenna [37] supporting contactless transactions. The above solution would need modifications to the mobile phone, such as additional RF (Radio Frequency) interface, adding connections from the (U)SIM to an external antenna. In our solution, we provide a software interface which allows SIAs such as (U)SIM in NFC mobile phones to authenticate to externally existing contactless and contact based smart cards without having to use any additional antenna. Now let us look into the proof of concept implementations of the proposed solution, on a PC and on a NFC enabled mobile phone. Furthermore, we propose a stand-alone hardware module with dual interface and firmware with translator interface, to realise the above.

A. CPTI on a PC Environment

The proof of concept for Communication Protocol Translator Interface (CPTI) on a PC environment has been implemented (illustrated in Figure 5) as a Java application which used the Opencard Framework [38] API (Application Programming Interface). The contactless reader and contact based reader were both registered to the “CardTerminalRegistry” service. The SE listener service for CPTI has been implemented using the “CardRequest” API, which listened for any contactless and contact based SEs. The APDUs were sent and received using the “PassThruCardService” API. The contactless and contact based targets were loaded and
installed with Java Card [39] applets that were developed. Let us call them Applet_CL and Applet_CT (bytecode of applets were 3 kilobytes each in size). The CPTI realised communication between Applet_CL and Applet_CT in the same session. When the contactless SE was presented to the contactless reader connected to the PC, CPTI selected Applet_CL and send an APDU initiating the session. The APDU response from Applet_CL to CPTI was passed to the Applet_CT on contact based SE present on the contact based smart card reader. The APDU response from Applet_CT was then forwarded to Applet_CL to which the applet program logic responded with the final status word ‘9000’ indicating the execution success and completing the communication cycle between Applet_CL and Applet_CT. Thus, CPTI translated the communication between contactless and contact based SEs. The Java application was developed using freely available Eclipse IDE (Integrated Development Environment) [40] on a PC (x86 family based PC) interfaced with two readers, a contactless ($65) and a contact based reader ($25).

B. CPTI on a Mobile Phone Environment

The proof of concept for Communication Protocol Translator Interface (CPTI) on a NFC enabled mobile phone environment has been implemented and is illustrated in Figure 5. The CPTI exists as a MIDP 2.1 application (MIDlet) executing on the J2ME software layer, and uses JSR 257 APIs for interacting with external contactless based SEs via phones’ contactless interface and SATSA (JSR 177) for communicating with contact based SE available on the phone’s contact reader interface. The CPTI MIDlet was 10.3 kilobytes in size. The external SEs, described in Section V-A, with Applet_CL and Applet_CT were presented to the NFC enabled mobile phone. In this instance, Applet_CT was implemented on a SIM based smart card. The CPTI on the mobile phone was signed [41] and given application privileges to access the smart card on the SIM reader interface. The SIM functions or the internal embedded SE can be utilised to further secure the CPTI interface in a mobile phone environment. The CPTI listened for any contactless SEs and when found, it initiated the communication session with Applet_CL implemented on a contactless SE. The response was then forwarded to the Applet_CT implemented on the contact based SE which is present on the contact interface (for example, (U)SIM reader) of the mobile phone, whose response was then returned to Applet_CL completing the full application protocol cycle. In reality, any contactless SE such as a contactless smart card, on SE on the NFC enabled mobile phone, is able to interact with the contact SE such as a SIM based smart card, a EMV based smart card using the NFC enabled mobile phone with CPTI implemented. The MIDlet implementing CPTI was developed, using freely available Nokia NFC SDK (Software Development Kit) [42] and was installed on a NFC enabled mobile phone, Nokia 6212 Classic [43], ($230). An alternate approach is to implement the CPTI on the SE of the NFC enabled mobile phone, but this would involve overhead in realising the communication and synchronisation between the embedded SE on the mobile phone and that on the contact based reader. The embedded SE is found to be more suitable for secure storage.

C. CPTI on a Dedicated Hardware

We envisage that CPTI could also exist as a special custom built hardware module with multiple contact and contactless reader interfaces, and running CPTI software interface on the module firmware. There is a similar solution presented in [44] which physically connects contactless and contact based reader interfaces with Single Wire Protocol (SWP) [45] and forming a special extended “adapter”. We note that this solution does not provide an opportunity to secure the communication (between a contactless SE and a contact based SE), but our solution achieved this with the help of secure channel protocol. The existing solution has inherent weakness in the RF link between the contactless interface on the “adapter” and contactless targets (for example, an NFC enabled mobile phone). It also relies on the security of the internal physical communication link. In our solution, CPTI implemented in the firmware on tamper resistant hardware would securely drive the module, and translate the communication between contactless and contact based SEs.

VI. CONCLUSION

In this paper, we investigated and found that there exists a gap within the NFC infrastructure in co-working with external contact based smart cards. With the evolution of mobile phones supporting multiple reader interfaces, it is now possible to include the transactions originating from such smart cards. We have improved the interaction of NFC devices with external smart cards especially those with contact based interface by providing a secure communication protocol translator interface, CPTI. We presented a security framework model and a public key based security protocol which would allow any NFC enabled mobile phone to access and use, over contactless interface, any additional smart cards which are externally available on contact based interface. The results provide a secure interface, for communication between internal and external SEs, using the mobile phone as a portal, over contactless and contact interfaces; and which also securely binds the disjoint passive tokens available over contactless and contact interfaces. This solution also provides support for secure P2P transaction between passive security tokens. We demonstrated the feasibility of our proposal by showing how to practically implement CPTI in a PC environment and also on an NFC enabled mobile phone. We illustrated the use cases of our proposal in electronic card identification, in access control, in mobile telecommunication, and in electronic card payments. Some
novel use cases of this work were identified and described, such as P2P payment using passive security tokens, using external smart cards for the purposes of identification and access control with mobile phones, crediting (U)SIM with a payment card, and accessing an external (U)SIM. In future work, we would further examine any potential security issues that may arise in the practical deployment of the proposed solution. We also plan to identify further use cases and implement practical identity and payment applications based on this contactless to contact (and vice-versa) communication protocol translation concept using CPTI.

REFERENCES


