Novel Type of Android Application Proposal: Ephemeral Authority Shar ing using NFC

Shi-in Chang and Yucheng Liu
{Electrical & Computer Engineering ,Information Security}
Georgia Institute of Technology, Atlanta

15 December 2012

Abstract

Owing to the advantages of NFC features like simplicity, ease of use, and low-cost, a large number of applications using NFC technology are widely spreading and providing convenience in everyday life. As an extended form of RFID, NFC can be used as a kind of access control method. However, NFC holdsthe same problem as RFID does in that there are always only two entities (i.e., Reader & Tag) in this access control model. In this paper, we propose a new type of multi-peer NFC communication mechanism, which allows a userto share the authority of accessing his/her own privilege-protected resources to a third party. We implemented the application on Android platform by designing mutual authentication protocols in peer-to-peer communication and encryption scheme in NFC application layer. We can build a secure authority sharing mechanism with various levels of restrictions among multiple entities and meanwhile insure that the third party will not be able to forge the authority or play replay attack.

1 Introduction

Near Field Communication (NFC) is a short-range communication, allowing two devices to exchange digital contents via one directional or bi-directional way by touching devices together. NFC's function is divided into three common modes namely file reading/writing, card emulating, and peer-to-peer (p2p) modes. Similar to the RFID system, NFC consists of readers and tags featuring an identification, but more commonly refered as active mode devices and passive mode devices[1]. Among the prevalent applications are making payments, ticketing, and use as a proximity card. This emerging technology is being integrated into mobile devices and is propelling a new paradigm, *mobiquity*, which means that every device all around us has been integrated into an interactive world with both mobility and ubiquity[2].

A large number of applications using the NFC are widely spreading during the last decade. Typically, incorporation of this technology into mobile phone is leading a new trend for an interactive world in a wealth of use cases in the past a few years; this is leveraged by several advantages of such an integration (e.g., comprehensive tools centralizing a user's daily life, high-level of functions, and powerful connectivity to the Internet).

According to the recent survey, 48% of US consumers are interested in mobile wallet system, and about 60% of them are expecting mobile-related companies would be the service providers rather than traditional banks[3]. With starting Nokia implementing NFC (i.e., Nokia 6131 was the first NFC phone in 2006), many of mobile vendors have provided the emerging trendy technology through their mobile products including Google Nexus series, Samsung Galaxy series, HTC mobile phones and LG mobile phones, etc. Recently, Motorola has launched a new phone, Motorola Droid Razr supporting NFC, (i.e., its first NFC phone) since the company was acquired by Google in earlier 2012.

Nevertheless, there are still remaining massive barriers to entry for NFC-based mobile commerce. One of the barriers to market realization is a security

issue. In a survey[4], the 33.77% of the respondents does not see any advantage in adopting Mobile Proximity Payment services, and they pointed out *a lack of security* as the first reason (i.e., 37.66%). This concern reflected a new iPhone5 launched recently. Apple decided not to include NFC because they believe it is not clear technology yet[5]. Moreover, the current NFC system is not scalable, but limited to only one pair-entity communication just as the RFID system does.

However, if we project such a communication model into an extended scenario (i.e., more than two entities can be taken into consideration at the same time), more application prospects and potential demands of NFC can be created. For instance, if father needs to lend his car (e.g., this car supports NFC unlocking system) to his son, we can simply come up with an idea; through tapping father's mobile phone and son's mobile phone together, they can simply complete the process of sharing the car key without any risk of losing the original key.

In this paper, we investigate security holes on the current specifications of NFC and propose a novel type of NFC application entitled with an ephemeral authority sharing application. The proposed application allows the owner of resources (e.g. laptop, tablet, or even car) to use a previously authenticated key securely stored in any of the owner's mobile devices. It also allows the owner to share the key via NFC technology with another person who wants to borrow the resources, temporarily and with limited authority. To the best of our knowledge, this is a new concept of authority sharing via NFC communication while previous research were focused on preventing authority leakage.

The vulnerabilities on NFC security concepts are also important and urgent issues, which should be taken into consideration. Link layer of NFC does not support any encryption scheme, and is susceptible to eavesdropping based on the current link layer control protocol[6]. Hence, we need to concern encryption scheme when designing application layer to countermeasure eavesdropping. Furthermore, the standardized NFC Data Exchange Format (NDEF) does not guarantee integrity and authenticity, even in the presence of a digital signature[7].

The remaining of this paper is organized as follows: Section 2 investigates previous related work in both RFID and NFC areas. Section 3 mentions NFC backgrounds, and Section 4 details our methodology with respect to security aspects. Section 5 presents the real implementation on Android Jelly Bean platform and a performance evaluation part will be followed in Section 6. Finally we provide a conclusion in the ending section.

2 Related Works

A growing number of applications using NFC technology built in mobile devices are being discussed and proposed in diverse areas, such as a card emulating mode, a passive tag read/write mode, and a p2p mode. Any of these modes is apparently associated with the others although only one mode can be selected at a time (i.e., for card emulating application, it should take account into writing/reading the debit and remaining balance consequently in an application).

Amongst them, Proximity Payment research (i.e., a card emulating mode) has been emphasized during the last decade, and specifically the use in a smartphone in the past few years. Ceipidor et al.'s survey on mobile payment[4] shows the adoption of new type of this payment is deeply depending on a level of user acceptance. Although the survey is within Italy, 75.03% of non-willingness to pay by drawing up the mobile phone instead of classic mode of payments is mainly concerning on the lack of security as a disadvantage of Mobile Proximity Payments. In [8], the authors indicate six factors influencing consumers in user-centric acceptance model such as ease of use, usefulness, mobility, cost, trust, and expressiveness. From these perspectives, many of NFC applications on mobile devices are proposed and realized. A core application standard was successfully applied to the development of an interoperable NFC ticketing system, and the system was fully integrated with the existing system architecture of the transportation association of Upper Austria[9].

Since the first mobile phone supporting NFC on the Android platform (e.g., Google Nexus S on Gingerbread, version of 2.3) was launched in the late of 2010, a research focus on NFC is expected to move fast onto the powerful platform from types of applications. It derives many of mobile vendors to pro-

duce diverse applications with more usefulness, mobility, and trust based on connectivity leading a new paradigm,*mobiquity*. Urien et al. propose a new MobileSE model for using resources hosted by Secure Elements (SE) embedded in the Android platform (i.e., NFC controller). A third party running in IP object establishes a secure channel with a mobile device, and providing shared secret. The authors clarify three functional steps for the deployment named smartphone registration, IP object enrollment, and secure channel set up and use [10]. The SE by itself is a hot topic in the current research on security on the mobile devices.

The secure element (SE) is a combination of hardware, software, interfaces and protocols embedded in a mobile handset that enable secure storage[11][12]. Selected by the user during card emulation mode[13], the SE provides a secure area for the execution of the applications and protection of the payment assets (e.g., payment data, keys, the payment application code)[14]. Besides being used for payment applications, the SE can also be involved in authentication process and for storing applications not related to payment, which require security mechanisms. As a consequence, the operating system running on the SE must be able to install, personalize and manage multiple applications issued by various providers preferably Over The Air[12].

In this reasons, a consideration of SE alternatives is one of the main issues on NFC research. Although many SE types of alternatives are currently analyzed by the NFC forum community, Reveilhac et.al. categorized four alternatives in [15];first classified by their ability of being removable from the handset or not, and then they are categorized by reusability or standardization progress.

Even if the stability of handling SE is guaranteed on a mobile device, any small portion of malfunction (i.e., unintentional bugs or intentional malware) of Android OS discourages users keeping trust when using NFC based applications, typically handling sensitivity data. In [16], vulnerability affecting Nexus S Android phones has already discovered. It could cause incorrectly formatted NFC transactions and lead to battery exhaustion and even denial of service. This is one of reasons that researchers should take into account operating system level of security when designing applications.

In other paper [17][18], such a backend server IP-connected to the mobile devices plays an important role to strengthen secure key sharing in the RFID lock systems and authentication process for access control in the smart poster applications, respectively.

Compared to the first two modes, a card emulation mode and a reader/writer mode, research on a p2p mode is still remaining unexplored. Lotio and Mazzocchi have just tried to implement an open source library supporting NDEF Push Protocol (NPP) facilitating the p2p over NFC [19].

3 Background: NFC

NFC technology is a short-range communication (limited to about 10 cm), which is useful for the development of various types of applications on the mobile devices ranging from a contactless payment to a smart poster. There are three different modes namely reading/writing, card emulating, and peer-to-peer (p2p) modes as shown in Figure 1[1].

Depending on the application types, a NFC-enabled device is being switched among the modes according to the procedures. NFC Forum specifications are based on existing and recognized standards like ISO/IEC 18092 and ISO/IEC 14443-2,3,4, as well as JIS X6319-4. They are implementation specifications that describe the parts of those standards that are relevant for NFC Forum devices. Therefore, compliant devices behave in the most consistent way, and the evolution of existing infrastructure toward full NFC is facilitated

3.1 Three Common Modes

3.1.1 Reader / Writer Mode: Touch & Go

In this mode, NFC device interoperates a RFID tag to write data in or to read data from. There are many use cases for the reader in advertisement, mobile coupons, and smart poster allowing the user to retrieve additional information by reading the tag. This configuration is defined in a pair of active device and passive device.

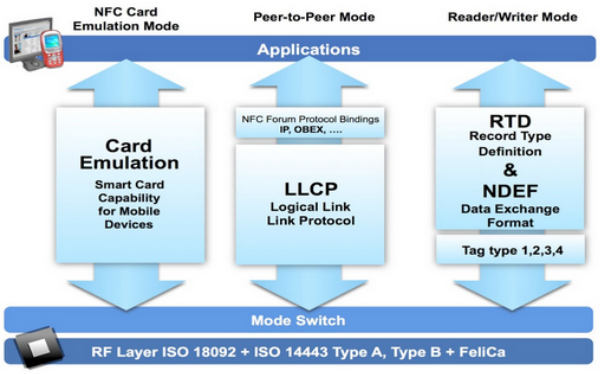

Figure 1: NFC and Interoperability - Three Common Modes

3.1.2 Card Emulator Mode: Touch & Confirm

This mode enables NFC device to act an existing contactless card. At this time, NFC device acts as transponder against an external card reader. It has many types of applications ranging from mobile ticketing to mobile payment. This feature is categorized in a pair of passive device vs. active device.

3.1.3 Peer-to-Peer Mode: Touch & Connect

P2P mode enables two NFC devices to communicate with each other by generating RF and exchange information. ISO 18092 defines Logical Link Control Protocol (LLCP) to establish the physical link. This mode is useful for exchange mobile address books, mobile configurations (e.g., Bluetooth, Wifi), or digital contents. In this mode, one of a pair of devices initiates any type of services to lead the other emulating a target. When the target responds to the communication, it also generates RF signal by its own poser supply whereas a passive device responds with coupled magnetic flux in the previous two modes.

3.2 Mode Changes

In the context of mode switch, whenever an NFC device sees another device in the radio field, it initially finds out whether it is a reader/writer, contactless card, an RFID tag or another NFC device. All these are made feasible by the mode switch design. It makes sure that an NFC device enters into a status in which it is able to communicate with the other device in the radio field. Additionally, it classifies the responses whenever a lot of cards are found in the radio field concurrently.

3.3 NFC Data Exchange Format (NDEF)

An NDEF is a data format classified by the NFC forum in connection with the exchange of information between two devices (i.e., an NFC-enabled device and an NFC tag). It presents rules in relation to the structure of a matching message, without limiting the types of information it contains. This permits the encapsulation of a large amount of varied data, such as images, URLs or XML files. It nonetheless, does not include any NDEF transmission protocol. For this reason, the type of channel for the transmission of messages is also liberally selectable, similar to the sort of information it contains.

An NDEF message is made of a series of NDEF records. Accordingly, the actual encapsulation of the data takes place in the individual NDEF records. Defined data formats that are commonly used, e.g. Uniform Resource Identifier (URI), Smart Poster, and Text are standardized by the NFC-Forum as Record Type Definitions (RTD) to allow interoperation of products coming from different vendors. The size and type of data transmitted can be recognized by means of the header. This allows a resourceful analysis of the information enclosed in the records to be carried out. With the help of the NFC Forum, a number of various types of information have been identified.

4 Methodology

4.1 General Concept of Authority Sharing System using NFC

In the near future, NFC technology will be employed for getting access to restricted areas or properties using mobile devices like RFID-based locking system. Using the current implementation of NFC, the key exchange and mutual authentication are great hassles. These open issues can be resolved by accompanying a concept like the 3rd guaranteeing party (e.g., Authentication Server) used in Kerberos Protocol as a part of the NFC functionality. The main concept of

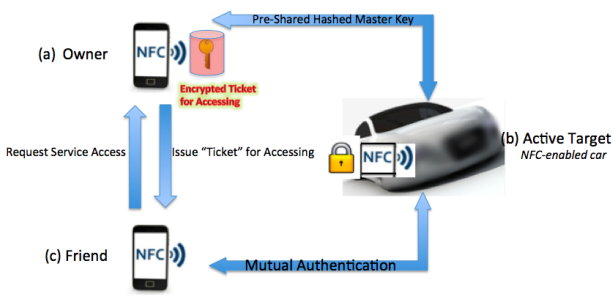

Figure 2: Authority Sharing System using NFC

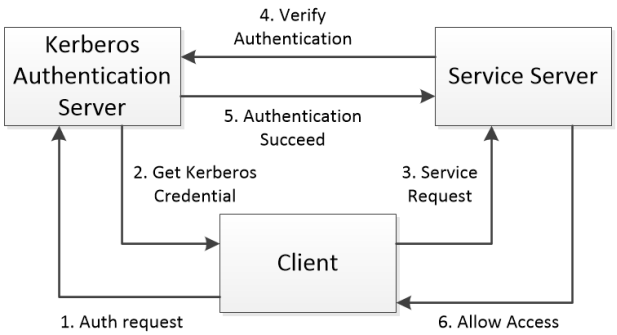

Figure 3: General Structure of Kerberos

our proposed application system is shown in Figure 2.

4.2 Starting from Kerberos

4.2.1 What is Kerberos

The figure 3 below shows the common structure of Kerberos protocol.

Kerberos is a distributed authentication system that many organizations use to handle domain-wide password security. One of the main concerns of Kerberos is that it prevent users password from being stored in unsecure (unencrypted, maybe), even in the authentication server database. Instead of using the password, Kerberos uses hashed value in encryption procedure (e.g., string2key function). The string2key is a type of a hash function, meaning that it is irreversible: given that an encryption key cannot determine the password, which generated it (unless by brute force).

4.2.2 Why Kerberos is good

We adopted Kerberos Protocol to our multi-peer application owing to several similarities in that Kerberos is an authentication protocol for trusted hosts on untrusted networks. This model is fitting on sharing authority through NFC communication in that a legitimate procedure is based on the relationship between trusted devices (i.e., by touching together).

Also, three entities in Kerberos system - Kerberos Authentication Server (AS), Service Server (SS) and Client map to Owner (O), Active Target (AT) and Friend (F) in our NFC authentication system, respectively. When F wants to get the authority of accessing AT from O, F needs to request for the authentication to O. If O trusts F and offers service ticket, then F gets the authority of accessing AT.

In addition, key sharing or key theft can allow impersonation attacks. If intruders somehow steal a principal's key, they will be able to masquerade as the user or service. To limit this threat, this system prohibits users from sharing their keys.

Here are some similarities between Kerberos and NFC Authority Sharing Protocol.

Firstly, Kerberos is an authentication protocol for trusted hosts on untrusted networks[20]. This model is fitting on sharing authority through NFC communication in that a legitimate procedure is based on the relationship between trusted devices (i.e., by touching together). However, when 3rd party joins to share the authority from an original holder, the topology of communication networks cannot be guaranteed as trustworthy.

Secondly, three entities in Kerberos system - Kerberos Authentication Server (AS), Service Server (SS) and Client map to Owner (O), Active Target (AT) and Friend (F) in our NFC authentication system, respectively. When F wants to get the authority of accessing AT from O, F needs to request for the authentication to O. If O trusts F and offers service ticket, then F gets the authority of accessing AT.

Thirdly, key sharing or key theft can allow impersonation attacks. If intruders somehow steal a principal's key, they will be able to masquerade as the user or service. To limit this threat, this system prohibits users from sharing their keys.

4.2.3 Why Kerberos is not good enough

There are big differences between our authority sharing model and Kerberos model. The Kerberos protocol assumes that all data exchanges occur in a circumstance where packets can be inserted, changed, or intercepted by intention. When 3rd party joins to share the authority from an original holder, the topology of communication networks cannot be guaranteed as trustworthy.

In our assuming scenario, the data packets are not so vulnerable as Kerberos because communication through NFC is not exposed broad widely based on its physical nature. Thus a required security level can be diminished.

Key Distribution Center (KDC) is a trusted third party with which every entity shares a secret key; this key is called the entity's master key. The KDC also maintains a centralized authentication database containing a copy of every user's master key. Based on the fact that it is not realistic for the owner to rent his/her properties to so many people that will increase the amount of key pairs significantly. Hence a standalone KDC in our system is not necessary.

4.3 Building a protocol

4.3.1 Assumption

A pre-shared hashed master key is generated at the first time for an Owner to register his device in the Active Target.

The NFC data will not be transferred unless another NFC device is detected within supported communication range (less than 10cm). With this reason, we assume all sensitive data including encryption keys and device IDs are kept secret unless NFC device owner intents to transfer any data.

To perfectly prevent leakage of service ticket (e.g., the Copyright Protection Mechanism against totally cloned impersonation), hardware secure element should be taken into consideration, which will make this design complicated. To simplify this design at this time, we assume that F is trustworthy and will not leak sensitive data to others intentionally.

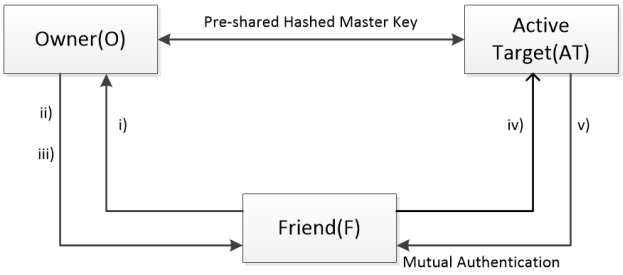

Figure 4: Authority Sharing System Protocol Structure

4.3.2 Protocol Structure

Upon the assumption of a secure key sharing between O and AT, the authority sharing protocol is followed as the sequences in Figure 4. In a pre-authenticated period,they agree any passwords to be used, which generates a hashed value for the master key (i.e., Key between O and AT, K_{O-AT})

Request Service Access in i) when F wants to access AT, F requests Service Access Granting with its own ID (e.g., Universal Integrated Circuit Card (UICC1) or an Secure Digital (SD) memory card) in plaintext. In person, they agree any passwords to be used, which generates a hashed value (i.e., Key between F and O, K_{F-O}).

Generate Service Session Key Kss in ii) this key is encrypted by K_{F-O}.

Issue Ticket in iii) the ticket for F to access AT with a limited validity is encrypted by the pre-shared hashed master key, K_{O-AT}.

This ticket includes O's ID, F's ID, Service Session Key, and Validity. F does NOT know what the ticket is composed of; it cannot be forged.

Forward the encrypted ticket in iv) F forwards the encrypted message of the step iii) without its knowledge.

Encrypt an authenticator in iv) authenticator is a timestamp, which is encrypted by Kss. AT can decrypt the ticket with K_{O-AT}.

The authenticator with Kss encrypted in the ticket, and then AT authenticates F (*reverse authentication*).

Reply for the authenticator in v) once AT decrypts the authenticator, AT encrypts authenticator plus one with *Kss* and replies to F, which can decrypt the reply with *Kss*, and then F authenticates AT (*forward authentication*).

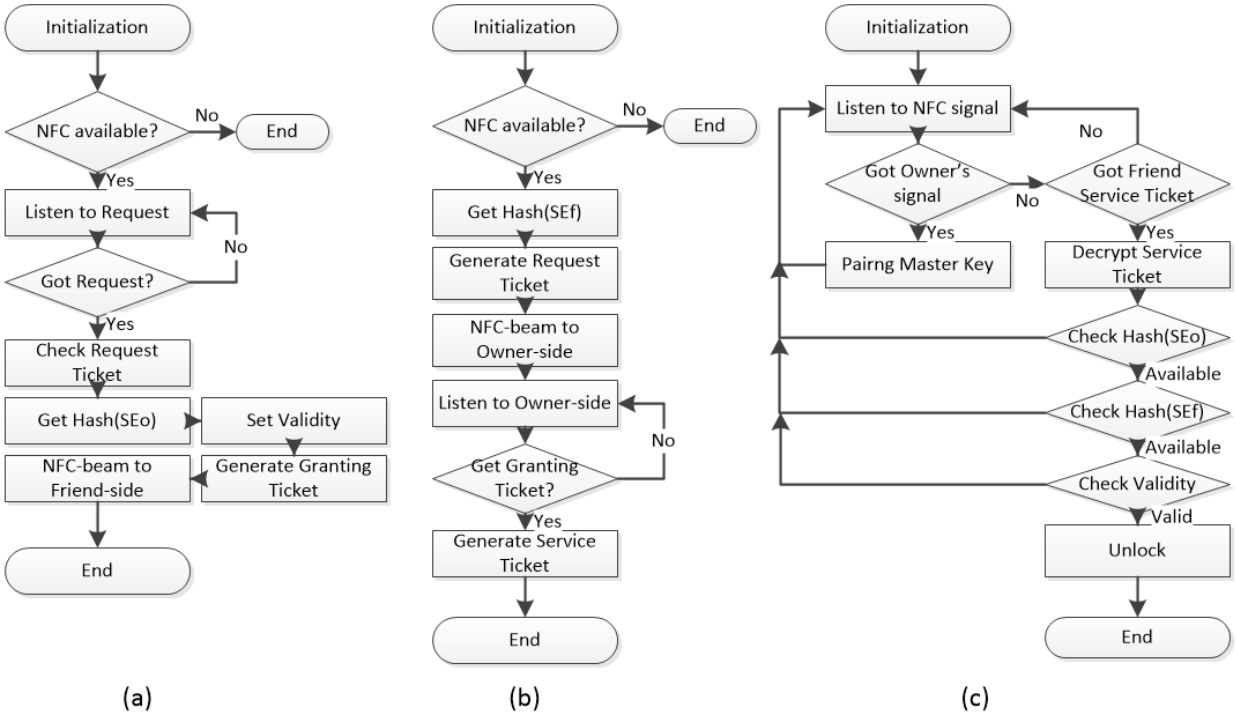

Figure 5: Architeture Design (a)Owner-Side (b)Friend-Side (c)ActiveTarget-Side

5 Application Study

5.1 Design Consideration

5.1.1 Assumption

This implementation is built based on the following assumptions:

1) The user should have a basic knowledge in accessing the application in Android and be experienced in a Android application usages and interaction.

2) This application is targeted to run on an Android version 2.3.3 and above and the user accessing the application should have permission to write data to the SQLite database on which this application is running.

3) This application that runs in different devices and plays totally different roles in each device. They will need to communicate each other through NFC. The operating system, namely, Android should allow this application to be installed and access the required resources such as the system calls and persistence layers of the target device. The application is not expected to run if there are any permissions are denied(for example, permission to use NFC sensor and get device ID).

4) There are no other apps that match the package name and variable names used by our application.

5.1.2 Constraints

This NFC Authority Sharing is an implementation of our design based on Android. The functionality still exists for newer versions but visually, the components are not where they are intended to be. This will be worked on future prototypes. The major design constraint is that since the software is itself is an interactive application, the capability of gracefully exiting the software in case of an error while accessing the operating system resources is minimum since the application accidentally exits before even the user can realize it.

Another constraint is that as internal leakage is always a tough threat to handle. To highlight our main ideas, we assume that Friend is always a trustworthy entity. We will discuss about possible resolution of dealing with internal leakage of authority sharing system.

5.1.3 Development Environment

The required system is a NFC-capable mobile device running Android version 2.3.3 and above. This is an application that runs on multiple(at least three) Android mobile devices and the user must have the permissions to install the application onto their device.

5.2 Architecture Design

The NFC authority sharing system is an application that runs on Android 2.3.3 as its minimum version. It allows the user to share authority with other devices with limited validity and proper controls

In Figure 5, there are three different flow charts in architectural design such as Owner-side, Friend-side, and ActiveTarget-Side.

In Figure 5(a), the Owners device will first check availability of NFC chip, if successfully detected NFC chip then it will be listening to NFC signal always until there comes a request for granting ticket. In the case that owners device detects a request, we assume that owner allows friend to get service and will generate granting ticket with data like Hash(SEo), validity, etc.

Once the friend wants to get service from Active Target, the device will first check the availability of NFC chip, if successfully detected NFC chip then it will use Hash(SEf) to generate a request ticket and transmit(through NFC-beam) to owner-side as shown in Figure 5(b). After successful transmitting this request, friends device will be listening to data backward from owner-side. Once it detects the granting ticket, it will get service ticket with padding its own Hash(SEf).

Finally, for the ActiveTarget-Side in Figure 5(c), the device will always be listening to NFC signal. Once it gets any NFC signal, it will first judge the source of this signal; if it comes from the owner, this Active Target will regenerate a new master key and pairing it to owner. If it comes from others, it will encrypt and check validity of service ticket by comparing the Hash(SEo) with such data stored internally, comparing the Hash(SEf) with the padding data and checking the left time of validity. If all pass, the friend will be permitted to get access.

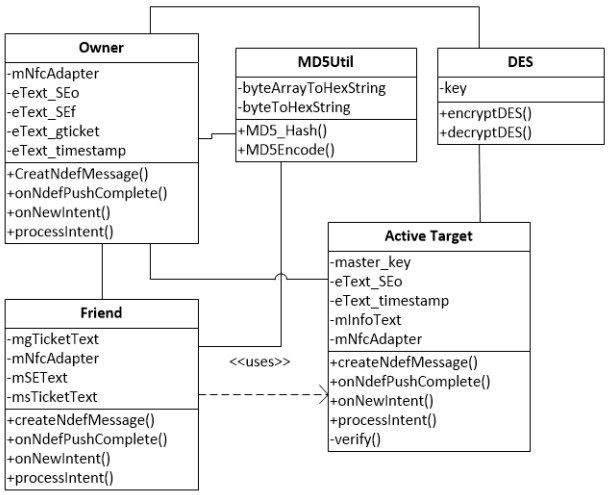

Figure 6: Class Diagram

5.3 Low Level Design

5.3.1 Class Diagram

MD5Util.java: This class aids in hashing the secure element of both owner's and friend's devices. The function MD5_Hash() will output MD5 value of secure element and the function MD5Encode() will standardize the format of data bits such that the data can be transferred using NFC API.
DES.java: The two functions encryptDES() and decryptDES() will encrypt and decrypt input using symmetric key which is generated through the pairing procedure with an Owner's device.
Owner.java: eText_gticket is an encrypted granting ticket generated with eText_SEo, eText_SEf, eText_timestamp. The function CreateNdefMessage() will generate the NDEF format data with eText_gticket. Functions onNewIntent() and processIntent() will get request from friend-side through NFC signal.
Friend.java: mgTicketText and msTicketText are granting tickets from owner-side and service ticket to active target-side. The function createNdefMessage() will generate NDEF format data. The function onNewIntent() and processIntent() will handle the NDEF data from owner-side.
ActiveTarget.java: master_key is generated at the time when owner's device is pairing with active target. The function verify() will compare the Hash(SEo) derived from service ticket and

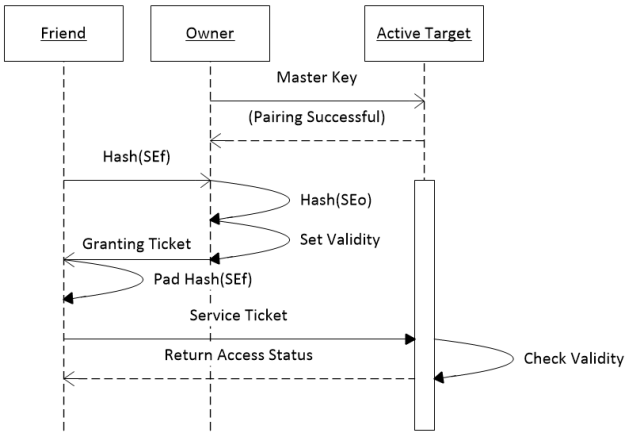

Figure 7: Sequence Diagram of Three Objects

Hash(SEo) stored in local database. Also it will compare Hash(SEf) decrypted from service ticket and from the padding part.

5.3.2 Sequence Diagram

Figure 7 above represents a Sequence Diagram for the three components in our system. This figure is a recap of interactions among three entities.

5.4 User Interface Design

Since this is an Android application, a user interacts with the application through a GUI. The navigation happens through the application screen which is fairly intuitive enough to address the user about the functionalities. As this is just a prototype for our design, the user should have some basic knowledge of the constructure of our system. The Figure 8 below shows an example of application UI demo. For example, the shared car key of descendent (i.e., in red) is limited to access full resources of the car.

6 Performance Evaluation

Eavesdropping: All NFC communications in our proposed application are encrypted with a symmetric cryptography. Here are two types of keys referred as a pre-shared master key (between an Active Target and its original owner) and a service session key Kss (between the owner and a borrower). The Kss is randomly generated by the owner, and uniquely on every service request base. Both of the keys are stored as a hash value, and are not open to the communication; a master key is not communicated through the channel whereas a service session key is encrypted with the master key in communication. With an assumption of the secure establishment of sharing the master key, all NFC transactions are protected from eavesdropping although LLCP does not support any encryption.

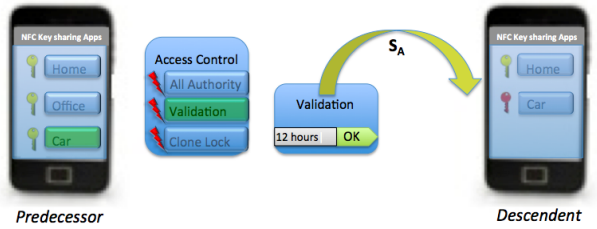

Figure 8: An example of User Interfaces

Replay Attack: In general, the encryption by itself is not secure against a type of replay attacks. Especially when a malicious opponent does not have to decrypt a message, it can use the encrypted message as a means of authentication or useless traffic exhaustion. In our proposed application, an authenticator, which consists mainly of timestamp is built from the owner and encrypted with the master key, will be used for a countermeasure of such a replay attack.

Access Control: With a definition of a ticket including various control features, the owner can handle the transferred authority to a borrower. The field of validity is one of them. This perspective inspires us to go further in developing the novel type of this application.

7 Conclusions

This paper demonstrates a novel type of a mobile NFC application based on the Android platform. This emerging technology is being integrated into mobile devices and is propelling a new trend of an interactive world with both mobility and ubiquity. In this paper, we propose a new type of peer-to-peer NFC application to share sensitive data (e.g., access-authority encryption keys) with various lev-

els of restrictions to another user to access privilege-protected resources. The proposed application allows the resource owner to use a previously authenticated key securely stored in any of the owners mobile devices. It also allows the owner to share the key via NFC technology with another person who wants to borrow the resource (e.g. laptop, tablet, or even car), temporarily and with limited authority.

Throughout developing and implementing the application, we find that NFC technology is secure to sharing sensitive data owing to the human intention driven event and a mutual authentication protocol. Moreover, encrypted type of ticket countermeasures an eavesdropping and a replay attack with protecting various level of authority control.

References

[1] NFC Forum homepage. [Online]. Available: http://www.nfc-forum.org

[2] J. Fischer, "NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]," *Communications Magazine, IEEE*, vol. 47, no. 6, pp.22–28, June 2009.

[3] S. Clark, "48 percent of Americans ready to use a mobile wallet," [Online] Available: http://www.nfcworld.com, June 2012.

[4] U.B. Ceipidor, C.M. Medaglia, A. Opromolla, V. Volpi, A. Moroni, S. Sposato, "A Survey about User Experience Improvement in Mobile Proximity Payment," in Near Field Communication (NFC), 2012 4th International Workshop on , vol., no., pp.51–56, March 2012.

[5] C. Arthur, "iPhone 5 shows that Apple still considers NFC as Not For Commerce," [Online] Available: http://www.guardian.co.uk/technology/apple-iphone-5-near-field-communication-nfc, Sep. 2012.

[6] Technical Specification, "Logical Link Control Protocol," NFCForum-TS-LLCP-1.1, June 2011.

[7] M. Roland, J. Langer, J. Scharinger, "Security vulnerabilities of the ndef signature record type," in 2011 Third International Workshop on Near Field Communication (NFC). pp. 65–70, Feb. 2011

[8] A. Zmijewska, E. Lawrence, R. Steele, "Towards understanding of ?factors influencing user acceptance of Mobile Payment systems," Proceedings of the IADIS WWW/Internet, Madrid, Spain, October 6-9, 2004.

[9] Widmann, R.; Grunberger, S.; Stadlmann, B.; Langer, J.; , "System Integration of NFC Ticketing into an Existing Public Transport Infrastructure," Near Field Communication (NFC), 2012 4th International Workshop on , vol., no., pp.13-18, 13-13 March 2012.

[10] Urien, Pascal; Kiennert, Christophe; , "A new cooperative architecture for sharing services managed by secure elements controlled by android phones with IP objects," Collaboration Technologies and Systems (CTS), 2012 International Conference on , vol., no., pp.404-409, 21-25 May 2012.

[11] J. Guaus, P. K. Liisa Kanniainen, P. Laaksonen, K. Murphy, J. Remes, N. Taylor, and O. Welin, "Best Practice for Mobile Financial Services," Mobey Forum.

[12] B. Choudhary and J. Risikko,"Mobile Financial Services Business Ecosystem Scenarios Consequences," Mobey Forum, Helsinki/Finland, April 2006.

[13] G. Association, Mobile nfc technical guidelines, GSM Association, 1st Floor, Mid City Place, 71 High Holborn, London WC1V 6EA, United Kingdom, Tech. Rep., Nov 2007.

[14] EMVCo, Emv mobile contactless payment technical issues and position paper, EMVCo, Tech. Rep., 2007.

[15] Reveilhac, M.; Pasquet, M.; , "Promising Secure Element Alternatives for NFC Technology," Near Field Communication, 2009. NFC '09. First International Workshop on , vol., no., pp.75-80, 24-24 Feb. 2009

[16] Urien, P.; Kiennert, C.; , "A new keying system for RFID lock based on SSL dual interface NFC chips and android mobiles," Consumer Communications and Networking Conference (CCNC), 2012 IEEE , vol., no., pp.42-43, 14-17 Jan. 2012.

[17] Wu, J.; Lin Qi; Kumar, R.S.S.; Kumar, N.; Tague, P.; , "S-SPAN: Secure smart posters in Android using NFC," World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a , vol., no., pp.1-3, 25-28 June 2012.

[18] Roberts, Paul; "Android NFC bug could be the first of many," [Online] Available:http://threatpost.com/en_us/blogs/android-nfc-bug-could-be-first-many-062111

[19] Lotito, A.; Mazzocchi, D.; , "OPEN-NPP: An Open Source Library to Enable P2P over NFC," Near Field Communication (NFC), 2012 4th International Workshop on , vol., no., pp.57-62, 13-13 March 2012

[20] "Kerberos: The Network Authentication Protocol," [Online]. Available: http://web.mit.edu/kerberos/