Emulation of Multi-application Contactless Smartcard in Near Field Communication Enabled Smart Phones

Awais Ahmad[1], Razi Iqbal[*2], Dawer Saeed[3]

[1-3]Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, G.T. Road, Lahore, Pakistan

[1]awais.ahmad@ozu.edu.tr; [*2]razi.iqbal@kics.edu.pk; [3]dawer.saeed@kics.edu.pk

***Abstract*-Near Field Communication (NFC) is a short-range wireless technology used to transmit data between two devices. In order to transmit data, NFC has three working modes. The card emulation mode has the greatest potential because it allows a smart phone to emulate passive NFC tags and contactless smart cards. Hence, it enables a smart phone to instantly share data with an NFC-enabled by a single tap on the NFC reader. Major applications of the card emulation mode include financial transactions such as purchasing a ticket at bus stop or a cinema hall, etc., so a secure storage space is required to locally store the data. The card emulation mode utilizes a Secure Element (SE) for the secure storage of data, and has no provisions for third party developers to write on this storage. This paper discusses the challenges of implementing card emulation in the Android Operating System for smart phones. Furthermore, emulation of a smart card defined by ISO 7816-4 for multiple application scenarios is discussed. The paper will go into detail regarding the implementation of card emulation mode and how it can help third party applications to read and write to and from SE while maintaining reasonable security.**

Keywords- Near Field Communication; Secure Element; Smart Card; Software Card Emulation; Two-way Communication; Short Range Wireless Communication

I. INTRODUCTION

Near Field Communication (NFC) is a short-range wireless technology which enables two-way communication. NFC is an emerging technology; due to its short range, it performs relatively better than other long and short range wireless technologies, e.g., Bluetooth, RFID, GSM and Wi-Fi for secure transactions, and identification [1, 2]. NFC has the ability to instantly transfer small amounts of data over small distances. Its ecosystem in smart phones is designed in such a way that it requires no pairing, unlike Bluetooth and Wi-Fi, hence making it an instant tap-and-go interaction. This instant action provides NFC with immense potential, including possible applications in various fields such as HealthCare [3], mobile payments, education, entertainment and enterprise [4].

NFC is a descendant technology of RFID and can sometimes be used in old infrastructures built for RFID. NFC was first launched in 2004 and standardized by the European Computer Manufacturers Association (ECMA) [5]. Later, the standardization was adapted to ISO/IEC-18092, ISO/IEC-21481. NFC is one of several advancements in radio frequency identification technology which follows smart card operating standards [6]. Application-oriented standards of data formats and protocols are handled by the NFC Forum, which is a non-profit organization [7].

Currently, NFC has been implemented in several systems and has proven its reliability and efficiency. For example, Razi, et al. [8] proved that NFC is better than barcode readers when tracking inventory items. Special orientation is necessary to read the barcode, and the barcode length increases as data size increases; however, with NFC, more data can be stored on chips of identical sizes. The integration of NFC into smart phones has made its scope even broader. Jeffrey Fischer has described it very well [9]; when equipped with Secure Element, NFC can also be used as a virtual wallet. NFC-enabled smart phones are highly interactive and user-friendly, combining the physical and digital world with the tap of a screen. NFC has three basic working modes based on the intended application:

- Reader Writer
- Peer to Peer
- Card Emulation

In Reader/Writer mode, NFC has two major components: an active device and a passive NFC tag. An NFC reader/writer is an active device that can read data stored on a tag and overwrite the tag with new data. Tags are passive devices that harvest energy from the radio waves of the active device via mutual induction. A passive tag can be placed anywhere, e.g., a wall, table, clothes, etc. As soon as an active device nears the tag, a magnetic field is generated and information is transferred. Active devices are usually embedded in smart phones or standalone systems that work separately or with a computer. In Peer-to-Peer mode, the NFC is used to transfer data between two active NFC devices. If communication occurs between two NFC-

DOI: 10.5963/IJCSAI0501002

enabled smart phones, one phone will act as an active NFC tag while the other behaves as an NFC Reader/Writer. Applications of active tags include contact/file sharing between two NFC-enabled smart phones.

This paper investigates the third operating mode: the card emulation mode. In the card emulation mode, an NFC-enabled smart phone emulates a passive NFC tag. There are two participants in this scenario: a smart phone and a reader device that is connected to a system. The smart phone appears to the reader as a dynamic passive tag. This mode has the highest potential in consumer applications because it eliminates extra cards and tags and makes use of the already ubiquitous smart phones. The mechanism and challenges of card emulation in Android phones will be discussed in this paper to investigate available options to develop a card emulation mode application.

II. CARD EMULATION AND ITS CHALLENGES

The card emulation mode of an NFC enables a smart phone to participate in scenarios in which a smart phone interacts with another NFC-enabled system. Such a system can be an automated boarding/ticketing system, secured payment systems or even access control systems. Fig. 1 shows a smart phone interacting with a reader in a smart card infrastructure. In most of these applications, a third party such as a mobile network operator must be involved to enable the overall system to function. This complicates the ecosystem of such models [10].

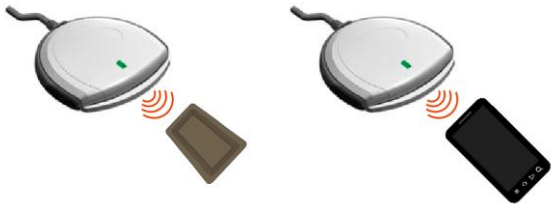

Fig. 1 A smart phone interacting with a reading device

A. Smart Card and Secure Element

For security purposes, a secure element must be involved in the application of the card emulation mode. Most phones that have NFC capability also have some form of a secure element. A Secure Element (SE) is a highly-secured smart card chip capable of running smart card applications. A smart card is a card with an embedded computer chip which has a microprocessor, ROM, EEPROM, RAM and I/O ports integrated into it, as demonstrated in Fig. 2. The yellow patches represent the pins which communicate with the smart card. VCC represents the power supply pin; the driving device of this chip sources power to the chip via this pin. RST is the reset signal and is used to reset the communications of the card. CLK is the clock signal which should be provided to the chip by the driving device; every digital device works on a clock signal, as does this chip. GND is a common reference voltage for this chip. VPP is typically used when the chip is required to save data to its non-volatile memory, e.g., EEPROM, and mostly used for proprietary purposes. I/O is the serial input and output pin used for the purpose of communication. Two pins, C4 and C8, are used for the USB interface and additional purposes which depend upon the scenario [11]. Some smart cards also contain a contactless interface, such as NFC. Their communication is accomplished without any physical contact with the driving device. Smart cards use various techniques to implement tamper resistance, making it quite difficult to extract data from the chip [12].

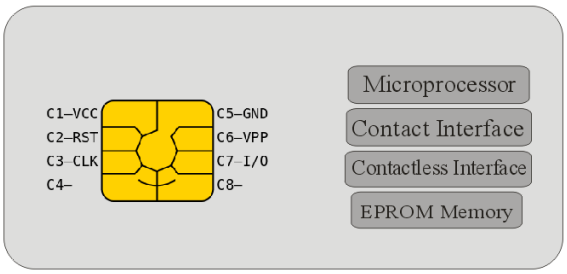

Fig. 2 Smart card components

Smart cards also come with a pre-installed operating system, which manages the hardware of the chip and on which it runs applets. In a card emulation application, the SE program is located at the centre of two devices: the NFC reader and a smart mobile device. The NFC reader communicates with the SE to perform a transaction (for example, a credit card transaction) via the NFC interface [10]. The Android application communicates with the SE to provide graphical interface to the user. In order to program the SE, the applet should be written and installed on SE. Java Card instructions are required to write applets in SEs, since SE itself is similar to a Java Card.

Access to security elements are closed to third party developers, in order to ensure security. The access to the secure element is limited to the mobile manufacturer or a network operator, or in some cases the financial body, depending on the

DOI: 10.5963/IJCSAI0501002

business model. Therefore, a common developer cannot create and deploy his own application into the market. There are many scenarios in which a secure element can exist, e.g., if a phone or smart device contains NFC and is able to perform card emulation, but all scenarios require access permissions which a common developer lacks, or they require tweaks to the OS (firmware) of the phone, hence making it impossible to launch a consumer application.

III. METHODOLOGY

The SE can be found in many forms; it can be embedded in the handset, connected to an SD card (Secure Digital non-volatile memory card) slot or integrated in an SIM (Subscriber Identity Module)/UICC (Universal Integrated Circuit Card). Most devices such as smart phones and smart watches, which support NFC, have a secure element embedded in the NFC chip, making it possible to access the chip wirelessly via NFC. In order to access the SE in UICC, permissions from the Mobile Network Operator are required. Thus, installing the applet on an SE can be expensive. Similarly, the manufacturer of the phone or the operating system controls the embedded SE. The SD Card Secure Elements are not yet very popular because they still require the addition of patches to certain Android OS, files which makes it tedious for use in a commercial application for a common user. For example, SEEK (Secure Element Evaluation Kit) is a smartcard API (Application Program Interface) which provides flexible access to the SE [13]. A person must add and compile the patches given in the SEEK to communicate with the SE.

There is another way to make use of the card emulation mode in Android, which is to emulate a software tag. The host processor of a smart phone has access to the embedded SE, and an applet (*Soft-SE*) can be written in the application to send and receive Application Protocol Data Unit commands on a low level [14].

A. Software Card Emulation

Research in Motion (RIM) introduced a functionality called “virtual target emulation”, which was intended to become available on the BlackBerry Bold 9900 device through the BB7 operating system; this was the origin of software card emulation [15]. This mode could be used to exchange data with another NFC device that operates in reader/writer mode. An application can emulate a full ISO/IEC 14443-4 smart card. Emulation is possible for both ISO/IEC 14443 Type A and Type B protocol variants. Software emulation allows the creation of a unique identifier UID of the tag for security reasons; hence, it cannot be used for applications which require low level UID referral, e.g., access control or financial transactions. This mode allows the exchange of protocol data units in addition to the block transmission protocol defined by ISO-14443-4.

Applications that emulate an ISO-7816-4 PKI smart card have been developed [3]. Previously, there were no public APIs (application program interfaces) that supported software card emulation; this application was implemented on an aftermarket firmware called CyanogenMod [16] which added patches [17] into its firmware. These patches enable this type of card emulation on Android devices with an NXP PN544 NFC controller.

Most recently, Google has introduced host card emulation [18], an API which allows a developer to emulate an NFC tag in an Android application. This API was introduced in Android 4.4 KitKat, and may require some time to capture a significant market share. Hence, CyanogenMod seems to be the only current option for implementing the software card emulation mode.

B. Implementation

Smart cards are basically cards with intelligent chips or memory chips within them. Smart cards have been available for decades and have evolved in many aspects. The best feature of smart cards is their enhanced security. Smart cards are excellent solutions for security applications, and have been used in payment and identification applications for years, e.g. employee time cards, ATM cards, and credit cards.

Smart cards are a combination of hardware and software. Normally, a smart card contains a built-in operating system known as firmware. The smart card operating systems have evolved in many ways over time, increasing their security and functionality [19]. They have been used as the core of intelligent memory cards, logical cards, microprocessor cards and contactless smart cards. Damien [20] discussed the multiple applications of smart cards. The evolution of smart card operating systems has enabled smart cards to run multiple applications through the same card, enabling endless applications in the daily life of a consumer.

The SE in a smart phone has the same capabilities as a smart card. Therefore, SE can emulate a smart card and can be used in already-existing smart card systems. Since an SE can host multiple applications such as smart cards, smart phones can replace most cards in a wallet. As shown in Fig. 3, SE in a smart phone can be accessed by the phone processor. In Fig. 3, the processor is represented by the host controller. The application running on the host controller can control the SE, allowing the proper structure of the security protocol which is defined by the ISO/IEC 7816 standard. The NFC controller begins to emulate the data from the SE, which is eventually controlled by the host controller [10].

DOI: 10.5963/IJCSAI0501002

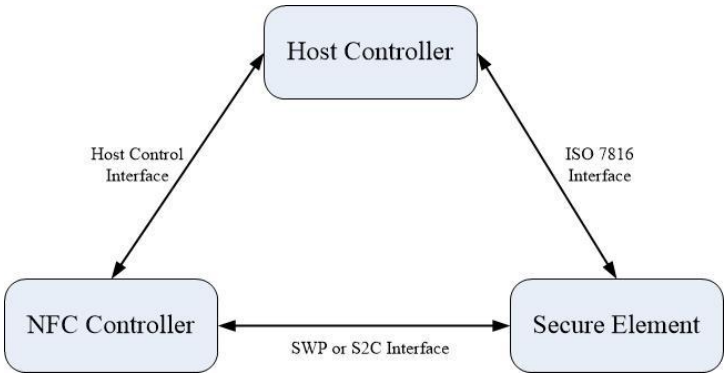

Fig. 3 Interface between host controller, NFC chip and the secure element in a smart phone

1) File System Management:

Every software card emulation in CyanogenMod (the aftermarket firmware) is restricted to ISO 14443-4 because it typically communicates via APDU (Application Protocol Data Unit) commands. ISO 7816-4 defines these APDU command structures and the OS. Based on [21], the operating system manages the application protocols and writes the commands in the host application.

The file organization supports two categories:

- Dedicated Files (DFs)
- Elementary Files (EFs)

There must be at least one dedicated file at the root, designated the Master File (MF). The file referencing system allows the creation of and access to any file at any hierarchy, as does a PC. As shown in Fig. 4, this file management and referencing system allows the developer to run a multi-application smart card on the host processor, using the SE to emulate the smart card running APDUs.

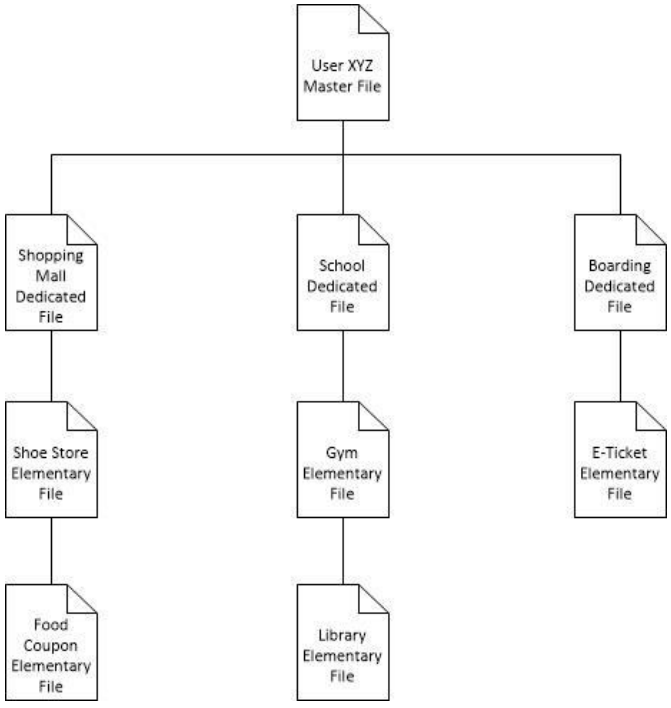

Fig. 4 File Management in a multi-application ISO 7816 smart card

2) Developing for Android:

Android works with NFC by registering apps for NFC intent filters. The OS has a foreground dispatch system which launches the activity registered for a specific intent filter as soon as that type of NFC event is detected. These NFC events are ACTION_NDEF_DISCOVERED, ACTION_TAG_DISCOVERED and ACTION_TECH_DISCOVERED. The first two catch the generic tag technologies supported by Android, while the third event catches the events that register a technology that is neither NDEF-formatted nor defined as a tag, but registers simply as an NFC technology [22]. To register these events, they must be defined in the "filter_nfc.xml" file of the application in Fig. 5.

DOI: 10.5963/IJCSAI0501002

```
<tech-list>
<tech>
android.nfc.tech.IsoPcdA
</tech>
</tech-list>
```

Fig. 5 Commands for developing in Android

The patches in CyanogenMod add support two new technologies: *IsoPcdA* and *IsoPcdB*. Both of these definte the majority of available Proximity Coupling Device readers. Hence, when an NFC reader is located in the field of the smart phone, the *ACTION_TECH_DISCOVERED* is registered and the app which filters this intent is launched. The host is then ready to communicate with the reader via a simple exchange of APDUs.

A typical card emulation operates differently, in which the phone actually emulates a passive tag. In software card emulation, both the NFC reader and the smart phone appear to each other as passive devices, and they both read and write one other through the exchange of APDU commands. Once the reader in registered as a tag in the Android application, all API methods for tag technology using reflection can be described as follows: the transceive() is the most important method, which sends raw APDU data over the NFC chip and receives the response.

The desired command according to ISO 7816-4 is stored in an array of bytes and transferred to the applet emulating the SE; it uses the NFC chip to communicate with the reader and registers its responses. A typical command sequence in both directions is described in Fig. 6.

```
Class<?> cls = Class.forName(tech);
```

→ [CLA] [INS] [P1-P2] [Lc – Data – Le]

(Data sent by phone or reader)

← [Data][SW1 – SW2]

(Response data)

Fig. 6 Command sequence for request and response

CLA is a class byte of a command which selects a command category. ***INS*** is the instruction byte, which selects a specific instruction out of a class. P1 and P2 are optional parameter bytes. **Lc** is the length of the actual data byte, and **Le** is the expected number of bytes in response to this specific set of commands. The response data consists of two fields: the optional Data field and the status of operation conducted on previously received commands.

IV. APPLICATIONS

Due to security reasons (as described in section III-B), software card emulation cannot be used in typical security and credit card applications. Fortunately, numerous applications exist solely for the purpose of improving the interaction of the digital and physical worlds. Some potential applications with significant impact on daily life are discussed here.

A. Loyalty Cards

NFC card emulation can be used in applications such as gift cards purchased for friends and family; people can digitally send cash or products as gifts. These cards will be saved in the recipient phone, to be swiped during checkout so that the gifts are stored digitally. Similarly, hotels and stores can offer loyalty or discount cards to customers online so that when they visit the store physically, relevant offers are downloaded to their phones.

The information in these cards can be sent to anyone around the world using the Internet or SMS. Businesses can deploy their promotional offers in the form of smart posters around the city, allowing consumers to store offers in their cell phones. The potential interesting applications are endless in this domain.

B. Boarding Passes

Boarding passes and e-tickets are emailed to the customers, or they downloaded from the internet, and customers must then bring the printed tickets at the boarding gates. These tickets contain barcodes, that the traveller must scan in order to acquire

DOI: 10.5963/IJCSAI0501002

their boarding pass. NFC card emulation can transfer the digital data to another device with a single tap. Thus, it eliminates the need to print e-tickets and scan them at the airport or a railway station. The e-ticket is downloaded and stored in the application and when the phone is placed in proximity to the reader at the gates, the data is transferred directly to the boarding system and a pass is generated.

C. Library Cards

A smart phone emulating a smart card can also be used as student ID card for issuing books from the library or sports equipment from the gym. As has already been discussed, a smart phone in emulation mode cannot generate a unique UID, making it impossible for applications with high security risks. However, the exchange of APDU commands can be used to construct an encrypted infrastructure for handling unique IDs and managing data. The record of books and utilities issued by a student can easily be synchronized both on the server and the smart phone of the student with just a single tap of the phone.

V. CONCLUSIONS

This paper evaluates the challenges of card emulation for a developer and describes software card emulation as a possible alternative for the card emulation applications. Card emulation is more accessible to developers as it bypasses the secure element. Software card emulation uses standards identical to those of smart cards; hence, smart phones with NFC and card emulation capabilities will be able to find immediate applications in already-existing solutions without any necessary hardware upgrades.

Software card emulation compromises the security of the system, as it exists solely in the code and lacks sophisticated hardware protection. Therefore, it must be avoided for payments or secure access control applications. The data is prone to relay attacks; hence, some common and simple interactive applications are suggested.

This paper evaluated the experimented results of an aftermarket firmware called CyanogenMod, which requires the user to perform tedious actions in order to install it on their phone. Fortunately, Google has provided API support for host card emulation with the recent release of the Android Kitkat 4.4. Solutions based on NFC are being implemented around the world. For now, these systems require passive tags involved in the form of tickets or tokens. We hope that with the passage of time, the technology will evolve to overcome the security problems and that smart phones may one day replace credit cards, ID cards, tickets and tokens, making smart phones virtual wallets or dynamic NFC tags.

REFERENCES

[1] F. Resatsch, S. Karpischek, U. Sandner, and S. Hamacher, "Mobile sales assistant: NFC for retailers," in *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services, ACM*, pp. 313-316, 2007.

[2] M. L. McKelvin Jr., M. L. Williams, and N. M. Berry, "Integrated Radio Frequency Identification and Wireless Sensor Network Architecture for Automated Inventory Management and Tracking Applications," in *Proceedings of the 2005 Conference on Diversity in Computing, ACM*, pp. 44-47, 2005.

[3] E. Strömmer, J. Kaartinen, J. Pärkkä, A. Ylisaukko-oja, and I. Korhonen, "Application of Near Field Communication for Health Monitoring in Daily Life," in *28th IEEE EMBS Annual International Conference*, pp. 3246-3249, 2006.

[4] K. Curran, A. Millar, and C. Mc Garvey, "Near Field Communication," *International Journal of Electrical and Computer Engineering*, vol. 2(3), pp. 371-382, 2012.

[5] https://en.wikipedia.org/wiki/Near_field_communication.

[6] M. Roland, "Software card emulation in NFC-enabled mobile phones: Great advantage or security nightmare," in *4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use*, pp. 6-12, 2012.

[7] (2015) NFC. [Online]. Available: http://nfc-forum.org/.

[8] R. Iqbal, A. Ahmad, and A. Gilani, "NFC based inventory control system for secure and efficient communication," *Computer Engineering and Applications Journal*, vol. 3(1), pp. 23-33, 2014.

[9] J. Fischer, "NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]," *Communications Magazine, IEEE*, vol. 47(6), pp. 22-28, 2009.

[10] V. Coskun and K. Ok, *Professional NFC application development for Android*, Wiley, 2013.

[11] https://en.wikipedia.org/wiki/Smart_card.

[12] (2012) Nelenkov, "Android Explorations: Emulating a PKI smart card with CyanogenMod 9.1." [Online]. Available: http://nelenkov.blogspot.com/2012/10/emulating-pki-smart-card-with-cm91.html.

[13] http://seek-for-android.github.io/.

[14] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *6th International Conference on Radio Frequency Identification: Security and Privacy Issues*, pp. 35-49, 2010.

[15] https://en.wikipedia.org/wiki/Host_card_emulation.

[16] https://en.wikipedia.org/wiki/CyanogenMod.

[17] (2014) D. Yeager, "Added NFC Reader support for two new tag types: ISO PCD type A and ISO PCD type B." [Online]. Available: https://github.com/CyanogenMod/android_packages_apps_Nfc/commit/d41edfd794d4d0fedd91d561114308f0d5f83878.

DOI: 10.5963/IJCSAI0501002

[18] http://developer.android.com/guide/topics/connectivity/nfc/hce.html.

[19] K. Markantonakis, M. Tunstall, G. Hancke, I. Askoxylakis, and K. Mayes, "Attacking smart card systems: Theory and practice," *Information Security Technical Report*, vol. 14(2), pp. 46-56, 2009.

[20] D. Deville, A. Galland, G. Grimaud, and S. Jean, "Smart Card Operating Systems: Past, Present and Future," in *5th NORDU/USENIX Conference*, pp. 12-22, 2003.

[21] D. Sauveron, "Multiapplication smart card: Towards an open smart card," *Information Security Technical Report*, vol. 14(2), pp. 70-78, 2009.

[22] http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-4.aspx.

A. Ahmad was born in Lahore, Pakistan on 4th June, 1991. He did his matriculation in 2007 from Latif Education High School, Lahore and then completed Pre-Engineering from Punjab College of Science, Lahore in 2009. He then got into University of Engineering and Technology, Lahore and completed his Bachelors in Mechatronics and Control Engineering in 2013. During his Bachelors he also went to University of Idaho, Moscow ID, USA for one semester as an exchange student in fall 2011. After completing his Bachelors he joined Al-Khawarizmi Institute of Computer Science as a Research Officer. There he worked on research projects related to Near Field Communication till August 2014. Currently he is pursuing his Masters in Robotics at Ozyegin University, Istanbul, Turkey. His research there is mainly about minimally invasive surgical biopsy robots.

R. Iqbal was born in Lahore, Pakistan on 17th July, 1982. Iqbal completed his early childhood education from Stirling School, Stirling, Scotland in 1988. Iqbal completed in secondary school from Divisional Public School Lahore in 1998. After completing his secondary school education he completed his Intermediate from Crescent Model Higher Secondary School, Lahore in 2000. He completed his Bachelors in Computer Science from University of Central Punjab, Lahore in 2004. After completing his Bachelors he went abroad for higher educations. He completed his Masters in Computer Science and Engineering from Akita University, Japan in 2008. He continued his education and completed his PhD in Computer Science and Engineering from Akita University, Japan in 2011Currently he is working as Chairman, department of Computer Science and IT and Director Office of Research, Innovation and Commercialization, Lahore Leads University. He is also working as a Research Scientist at Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan. He has published several research papers in renowned international journals.

D. Saeed was born in Lahore, Pakistan on 25th June, 1989. Saeed started his early schooling from KAPCO Secondary School, KAPCO, Kot Adu and after he moved to Lahore in 2002 he completed his Secondary education from board of Intermediate and Secondary education Lahore in 2005. After completing his Secondary education he completed his intermediate from Govt. Islamia College Civil Lines, Lahore in 2008. He completed his Bachelors in Mechatronics and Control Engineering from University of Engineering & Technology, Lahore in 2012. After completing his bachelors he started working for Khawaja Electronics Pvt. Limited as Production Engineer and then he switched to Pakistan Cycle Industrial Co-operative society as Assistant Manager Production in June 2013. He is now working as a Research Officer at Al-Khwarizmi Institute of Computer Science, University of Engineering and Technology, Lahore Pakistan since August 2014. He has worked on different projects majorly on Near Field Communication and has written different papers. He has also presented his paper at International Conference on Open Source Systems and Technologies (ICOSST) held at UET Lahore on 17-20th December 2014.

DOI: 10.5963/IJCSAI0501002