The State of Near-Field Communication (NFC) on the Android Platform

Jaromír Karmazín and Pavel Očenášek(✉)

Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
xkarma06@stud.fit.vutbr.cz, ocenaspa@fit.vutbr.cz

Abstract. We analyze the Android operating system as a platform for building NFC-enabled applications. First, we briefly examine the security of NFC and provide an overview of the three modes (reader/writer, peer-to-peer, card emulation) that are exposed to developers through Android's API. Furthermore, we present some existing Android applications using NFC, such as diagnostic tools, contactless tag manipulation tools, peer-to-peer NFC applications, as well as a few uncommon use cases. We conclude with an assessment of the completeness of Android's NFC API and suggest a novel use case.

Keywords: Near-Field Communication · Mobile device · Android · Application · Communication

1 Introduction

Near-Field Communication, or NFC for short, is a somewhat new technology that allows direct wireless communication between two mobile devices, or between a mobile device and a passive tag, over a short distance.

The number of devices supporting NFC has been growing recently, including models such as the mobile phones Apple iPhone 6, Microsoft Lumia 950, and Samsung Galaxy A, as well as tablets like the Google Nexus 10 [1].

In this article, we are going to focus on the Android platform, its NFC capabilities, and existing applications for Android that utilize NFC.

2 Security of NFC

Where using NFC is an option, the very nature of the technology makes it a harder target for attacks than other technologies such as Bluetooth or Internet-based services. In order for an attacker to eavesdrop on or interfere with NFC communication, they need to establish a physical presence near the two communicating devices, at a range shorter than conventional wireless technologies. There is also no infrastructure involved, so there is no trusted third party that the attacker can spoof.

As with any method of wireless communication, eavesdropping on the physical layer is possible if the reception is good. Per [2, Sect. 7.5.1], "the distance can generally be greater than the standard reading distance" in the case of RFID tags. Haselsteiner and

T. Tryfonas (Ed.): HAS 2016, LNCS 9750, pp. 247–254, 2016.
DOI: 10.1007/978-3-319-39381-0_22

Breitfuß state in [3, Sect. 3.1] that for NFC, eavesdropping can be done within about 10 meters of a device communicating in active mode and within about 1 m of a device communicating in passive mode. However, they give these numbers only as rough estimates, reasoning that a "huge number of parameters" determine the radius of possible eavesdropping.

It is possible for an attacker in the vicinity of communicating devices to corrupt the transmitted data by also transmitting at the same frequencies with the correct timing and modulation. This is a Denial of Service (DoS) attack [3, Sect. 3.2].

It is also possible for an attacker to modify the transmitted data by transmitting a specific signal at the same time. Except when 100 % amplitude-shift keying (or ASK for short) modulation with modified Miller encoding is used, all bits can be modified by an attacker [3, Sect. 3.3].

An attacker may insert messages into a legitimate communication, but only if there is a long enough pause between two legitimate messages [3, Sect. 3.4].

A Man-in-the-Middle attack is considered infeasible by [3, Sect. 3.5] because of the overlapping radio fields. However, we would like to not rule out the possibility of an attack being developed in the future. For example, a sticker with two antennae and a shield separating them, placed on an NFC device, could be devised for such an attack. Applications should always provide a mechanism for verifying the other party.

For certain security-sensitive applications, such as mobile payment, the mobile device needs the level of security similar to a smart card. The application's private data should be stored in a location where other applications, and perhaps even the user, cannot read it or tamper with it.

Secure NFC exists for these purposes [4, Sect. 11.6.1]. It relies on the presence of a hardware module called the *secure element*, which can be a SIM card, a secure memory card, or a smart card chip. In this setup, the security-sensitive applet runs in the secure element and communicates directly with the NFC device. An application controller, which may run in an insecure environment, serves only administrative purposes and cannot access the secure element's data directly.

3 Android's NFC Capabilities

Android's API allows developers to use the NFC capabilities of the host device. This is documented at [5]. This section serves as a summary of these documented possibilities.

3.1 Reader/Writer Mode

Android applications can "read and/or write passive NFC tags and stickers". This uses NFC in passive mode, with the Android device as the initiator and the tag or sticker as the target. The initiator creates a magnetic alternating field, which it uses to send both power and data (using ASK) to the target. In order to send data in response, the target uses load modulation on the initiator's magnetic field.

When the Android device's screen is unlocked and NFC is not disabled in its settings, the device scans for nearby NFC tags. Once a tag is discovered, Android creates an intent

(a specific type of object in Android's API), locates the application best suited to handle the given tag, and dispatches the intent to it.

Per the documentation, "Android has the most support for the NDEF standard, which is defined by the NFC Forum."

In addition, the API exposes interfaces which allow raw communication with other tag technologies. In these cases, the application must implement its own protocol for communicating with the tag.

The supported technologies are:

- NFC-A (ISO 14443-3A),
- NFC-B (ISO 14443-3B),
- NFC-F (JIS 6319-4),
- NFC-V (ISO 15693), and
- ISO-DEP (ISO 14443-4).

Optionally, Android devices may support these additional NFC tag technologies:

- MiFARE Classic and
- MiFARE Ultralight.

API level 9 (Android 2.3 Gingerbread) offers limited tag reading support. The reader/writer mode is supported comprehensively starting from API level 10 (Android 2.3.3 Gingerbread MR1). Android Application Records, which provide a stronger certainty that an NFC tag will be handled by an application, are supported starting from API level 14 (Android 4.0 Ice Cream Sandwich).

3.2 Peer-to-Peer Mode

Android offers functionality called Android Beam, which allows sending an NDEF message from one Android-powered device to another. This uses the active mode of NFC, which means that both devices take turns transmitting both power and data. In the same way as the initiator in NFC's passive mode, each device in active mode transmits data using ASK modulation.

To send data over Android Beam, the sending application must be running in the foreground of the sending device, and the receiving device must be unlocked and within a close range of the sending device. When these conditions are met, the sending device displays a "Touch to Beam" UI. If the user confirms the action using this UI, then the data is sent.

The payload of Android Beam is transmitted in one way only. The API does not provide any way to receive any payload as a recipient from the recipient.

Android Beam is supported starting from API level 14 (Android 4.0 Ice Cream Sandwich).

NFC is not suitable for the transfer of large files because of its low data rate, combined with the fact that the sender and receiver need to be in close proximity for the whole duration of the transfer. According to [7], the "sane upper bound" for data transferred using NFC is "about 1 KB (. . .), which can usually be exchanged within 300 ms".

Since API level 16 (Android 4.1 Jelly Bean), it is possible to transfer large files between Android devices using the Android Beam file transfer API [8]. Android overcomes the speed limitations of NFC by only using it for the initial setup, then silently enabling Bluetooth, temporarily pairing the two devices, and performing the actual file transfer over Bluetooth [9].

A similar approach to transferring large files was implemented in Sam- sung Galaxy S III, whose S Beam technology uses Wi-Fi Direct instead of Bluetooth to transfer large files after a connection is initiated over NFC [9].

3.3 Card-Emulation Mode

Android provides an API to implement host-based card emulation [7]. This allows Android applications to talk directly to the NFC reader without involving a secure element.

In this mode, the Android device acts as an NFC target in passive mode. This means that the Android device emulates a possibly unpowered tag and does not create a magnetic field. It only transfers data in response to an initiator (reader) using its magnetic field and load modulation.

Supported are emulated cards "based on the NFC-Forum ISO-DEP specification (based on ISO/IEC 14443-4) and process Application Protocol Data Units (APDUs) as defined in the ISO/IEC 7816-4 specification". The developer implements their own protocol stack for sending and receiving those APDUs.

Card emulation only works when NFC is enabled and the screen is on. Unlike the reader/writer mode, this can work from the device's lock screen and does not require any application to be in the foreground.

This functionality is available starting from API level 19 (Android 4.4).

3.4 Missing Features

Even though Android's API supports all modes of NFC operation, there is no public API for using NFC's secure element as of Android 6.0, so payment applications need to use host-based card emulation, which provides less protection for sensitive data.

In addition, NFC's peer-to-peer mode can only be used for Android Beam, which only allows one-shot, simplex data transfers. There is no way to obtain any payload in response to a peer-to-peer request, let alone develop a custom communication protocol, using only the public API.

4 Android Applications Using NFC

We searched for the term "NFC" on [6] in order to get an idea about how NFC is used in existing Android applications. In this section, we will analyze the many results that turned up in the search, attempt to categorize them, and point out some of the most unusual applications.

4.1 NFC Diagnosis and Management Applications

As one would expect, among all NFC applications, there are ones intended for managing and diagnosing the NFC subsystem itself, rather than making any actual use of it. These include simple applications for checking the availability of NFC on the device (e.g. NFC Enabled? by Espen "Rexxars" Hovlandsdal) and widgets for enabling and disabling the device's NFC inter- face (e.g. NFC Widget by AIT APPs).

4.2 Tag Reader/Writer Applications

The largest portion of applications using NFC focuses operations with NFC tags. Naturally, this would make use of passive NFC in reader emulation mode.

There are many applications that provide low-level support for NFC tags, such as displaying basic information about a tag (make, model, supported NFC technology, serial number, etc.), reading data from a tag, writing data to a tag, and tag cloning. It can be assumed that some knowledge of the NFC technology is required in order for the user to find any value in these applications, so average users are probably not the target audience. Examples of these applications are NFC Tools by wakdev, NFC ReTag by WidgApp Mobile Solutions, NFC Reader by Adam Nybäck, and NFC TagWriter by NXP Semiconductors.

Certain specialized applications allow extracting more information from NFC tags used in existing real-world systems:

- Credit Card Reader NFC (EMV) by Jullien Millau uses NFC "to read public data on an NFC banking card compliant with EMV [norms]", such as the contactless credit and debit cards made by Visa, American Express, and MasterCard. The application can show the card type, number, expiration date, number of remaining PIN entries, and in some cases the card's holder's name and the card's transaction history.
- Metro tickets of Moscow by Dmitriy V. Lozenko uses NFC to read information off of such tickets. The information includes the ticket's type, number, date of issue, date of expiration, the number of trips (both total and remaining), and the name of the last station (presumably the last station where the ticket communicated with a reader).
- Octopus by Octopus Cards Limited uses NFC for reading so-called "Octopus cards", which can be used for transportation, parking, retail shopping, and other facilities in Hong Kong, as well as for online payment using an NFC-enabled Android device [10].
- saldoBip NFC by YANKO uses NFC to check the balance of a "bip! card", which is a prepaid ticket used by the public transportation company Transantiago operating in Santiago de Chile [11].
- ShareMoreTransport by Share More Studio uses NFC to check the balance and transaction records on public transport cards used in various Chinese cities, such as Beijing, Shenzhen, Wuhan, Hongkong, Qingdao, and Xian.
- T-money Balance Check by RW MobiMedia uses NFC to check the balance on a South Korean contactless payment card for transportation and some convenience stores.

Another frequent use case is the use of NFC tags to trigger certain actions or change certain settings in the device that reads them. This would allow a user to place NFC stickers in certain places of interest, such as different rooms at their home, in their car, etc., and have the phone enable or disable wireless interfaces, change the ringtone volume, open certain applications, etc., based on the tag which is scanned. Examples of these applications are Trigger by Egomotion and NFC Tasks by wakdev. Another peculiar instance is NFC Alarm Clock by kamituel, which requires the user to scan an NFC tag in order to mute a ringing alarm.

A related concept is implemented in WifiTap by Andreas Rossbacher, which allows writing Wi-Fi credentials to, and reading those credentials from, a writable NFC tag. This can be useful for providing Wi-Fi access to guests at home, at an office, or at a public event.

There are a few applications which use the NFC reader mode in a novel way. For example:

- NFC Developer by Thomas Rorvik Skjolberg allows a software developer to prepare data for an NFC tag on a computer, transfer that data onto their Android phone using a QR code, and finally program an NFC tag using the phone, thus eliminating the need for a separate tag reader.
- SmartPassLock NFC by DreamOnline, Inc. adds another method of unlocking an Android device's screen, requiring a registered NFC tag to be present (the application's description only mentions support for Japanese IC cards).
- Crypto NFC by rolios allows users to write notes on their Android device, encrypt them, and keep the decryption key on an NFC tag.

4.3 Android Beam Applications

Some of the default Android applications allow sharing content, such as web pages, contacts, and YouTube videos, through Android Beam [5]. Since Android 4.1 Jelly Bean, it is also possible to share photos and videos from Gallery with the help of a transient Bluetooth connection [9].

Applications like NFC Transfer Beta by Abhinava Srivastava and NFC File Transfer by apps4u@android use Android Beam to transfer arbitrary files. SuperBeam by LiveQoS offers the same functionality but uses NFC or QR codes for pairing and Wi-Fi Direct for file transfer.

Share Tether NFC by Javi Pacheco uses Android Beam to "share tethering between two devices using NFC".

4.4 Miscellaneous NFC Applications

Applications like CardShake by Tesla System Co., Ltd. or Business Card Holder with NFC by ATSolution allow users to exchange electronic business cards over NFC.

Certain printers by HP Inc. can print content received via NFC from an Android device with the HP ePrint application installed.

NFC Porter by IMA s.r.o. uses NFC for premises access control when used together with a compatible identification system.

PassWallet - Passbook + NFC by Above Mobile Limited allows storing tickets and coupons in Apple's Passbook file format and redeeming them through NFC readers.

5 Conclusion

Android's support for NFC is quite extensive, with the most recent major update being part of Android 4.1, a version of the operating system released in 2012. Recent Android devices should, therefore, be able to use all of the features mentioned in this article, provided that their hardware supports NFC.

Without installing any applications, NFC can already be used for sharing of content between two Android devices using Android Beam. This functionality is user-friendly as well as programmable. However, developers are not given the option to extend Android Beam's capabilities too much beyond its original purpose.

The Google Play Store lists many additional applications that expand on the device's use of NFC. Enthusiasts can choose from the variety of reader/writer applications to handle NFC tags. Hardware vendors can support NFC in their products to make them easier to connect to Android devices.

Android's NFC API is the most complete in regards to card reading and writing. For manufacturers of contactless cards, this API should provide everything necessary to create an Android application to complement such cards. Some applications made for specific contactless cards have been listed as examples in Sect. 3.2.

Android provides API for host-based card emulation, but with no support for using the secure element, so applications must store all data in the phone's memory. Certain security models cannot be applied to this scenario, which disallows implementation of certain applications, especially those for contactless payment.

Although Android's support for NFC is quite extensive, we believe that the secure element and custom peer-to-peer communication are two features whose support should be added to Android to make it a mature and versatile platform for NFC applications.

We found no information about Android using encryption in its NFC communications. While security-sensitive applications will likely use an encrypted layer in their own protocol stack, developers should be aware that using Android's API for NFC may, by default, leave their communications susceptible to eavesdropping and unauthorized manipulation. In addition, the possibility of a Denial of Service attack must always be taken into account.

While the Google Play Store already provides a decent variety of NFC applications, we believe it could still be expanded upon. For instance, peer-to-peer NFC could be used for two-way synchronization of photo collections, certain types of notes, game data, etc. Moreover, we have yet to find an application that uses NFC for securely pairing two devices and exchanging cryptographic material, a use case that is also the subject of our thesis to be finalized in May 2016.

Acknowledgements. This project has been carried out with a financial support from the Brno University of Technology, Faculty of Information Technology through the specific research grant no. FIT-S-14-2299: Research and application of advanced methods in ICT.

References

1. List of nfc phones (2016). http://www.nfcworld.com/nfc-phones-list/. Accessed 28 Feb 2016
2. Oertel, B., Wölk, M., et al.: Security aspects and prospective applications of RFID systems. Bundesamt für Sicherheit in der Informationstechnik, Bonn (2005). http://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/RFID/RIKCHA_englisch_pdf.pdf?__blob=publicationFile
3. Haselsteiner, E., Breitfuß, K.: Security in near field communication (nfc). In: Workshop on RFID Security, RFIDSec 2006 (2006). http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf
4. Finkenzeller, K.: RFID Handbook, 3rd edn. Wiley, Hoboken (2010). ISBN 978-0-470-69506-7
5. Android developers: Near field communication. https://developer.android.com/guide/topics/connectivity/nfc/nfc.html. Accessed 11 Dec 2015
6. Google play store. https://play.google.com/store/. Accessed 01 Jan 2016
7. Android developers: Host-based card emulation. https://developer.android.com/guide/topics/connectivity/nfc/hce.html. Accessed 02 Jan 2016
8. Android developers: Sharing files with nfc. https://developer.android.com/training/beam-files/index.html. Accessed 01 Feb 2016
9. Jelly bean feature: Sending photos and videos over android beam. http://www.androidcentral.com/jelly-bean-feature-sending-photos-and-videos-over-android-beam. Accessed 02 Jan 2016
10. Octopus hong kong: Where can I use it? http://www.octopus.com.hk/get-your-octopus/where-can-i-use-it/en/index.html. Accessed 02 Jan 2016
11. Qué es la tarjeta bip! (Spanish). http://www.tarjetabip.cl/como-funciona.php. Accessed 02 Jan 2016