PROPOSED ARCHITECTURE AND THE DEVELOPMENT OF NFCAFE: AN NFC-BASED ANDROID MOBILE APPLICATION FOR TRADING TRANSACTION SYSTEM IN CAFETARIA

Fikrul Arif Nadra, Heri Kurniawan, and Muhammad Hilman

Faculty of Computer Science Universitas Indonesia, Kampus Baru UI Depok 16425

Email: fikr4n@gmail.com

Abstract

The development of mobile technology and RFID leads to an innovative mobile payment technology by using NFC. One of the popular mobile device platforms today is Android. This research proposes an architecture of NFC-based payment system in cafeteria – which is called NFCafe – and the implementation of it in Android applications. It is a closed payment systems – without involving third parties such as banks. The security issue is handled by using symmetric and asymmetric encryptions, they are RSA and AES. The application has been developed and successfully passed the testing conducted by several respondents.

Keywords: *RFID, NFC, Android, Payment System*

Abstrak

Perkembangan teknologi *mobile* dan RFID mengarah ke teknologi pembayaran *mobile* yang inovatif dengan menggunakan NFC. Salah satu platform populer perangkat *mobile* saat ini adalah Android. Penelitian ini mengusulkan arsitektur sistem pembayaran berbasis NFC di kantin - yang disebut NFCafe - dan implementasi dalam aplikasi Android. Ini adalah sistem pembayaran tertutup - tanpa melibatkan pihak ketiga seperti bank. Masalah keamanan ditangani dengan menggunakan enkripsi simetris dan asimetris, RSA dan AES. Aplikasi ini telah dikembangkan dan berhasil lulus pengujian yang dilakukan oleh beberapa responden.

Kata kunci: *RFID, NFC, Android, Sistem Pembayaran*

1. Introduction

NFC (Near Field Communication) is a short range wireless technology which enables two devices to communicate when in close proximity. NFC was developed by Philips and Sony in the late 2002. In 2004, Philips, Sony, and also Nokia established NFC Forum to promote this technology. [1]

Compared to other wireless technologies, such as Bluetooth, NFC has several advantages, e.g. the range is short, no need for setup/pairing, and the ability of connection between an active device and either another active device or a passive device (NFC tag). Its short range is considered as an advantage because it makes eavesdropping harder. NFC is predicted to be a standard feature for most mobile phone in the future. [1][2]

NFC is derived from RFID technology, and in the communication there are always an initiator which actively emits RF field and a target which is powered by that field [12]. Its operating range is only 10 cm [1] or 5 cm [12]. It transfers data in 106/216/414 kbps rate [12], or 424 kbps maximum [1]. An NFC device emits low frequency radio wave in 13.56 MHz [1]. The standard message format used in NFC communication is NDEF (NFC Data Exchange Format), it consists of one or more records and each record contains several fields [2]. The important fields to note are TNF (Type Name Format), type, ID (optional), and payload [2].

The Android platform has supported NFC access since API Level 9, with some limitation. The development was continued and improved in API Level 10 and 14. [16]

A simple peer-to-peer data interchange between devices can be conducted by using Android Beam as the user interface. The receiving device should be active (not locked). When the devices in close proximity, the Android Beam

interface will appear so the user can choose whether to "beam" (send the data) or not. [2][16]

In practices, NFC can be used for several things, e.g. goods and services payment, event ticketing, and facility and computer access control [1]. One of the mobile phone platforms that support NFC is Android, a popular smartphone platform today.

On the applications of NFC in payment system, it is used a substitution of credit or debit card. Of course, customers should use bank's services which have supported NFC-based payment. However, NFC-based payment is not limited to payments that involve banks. In other words, NFC technology can be used for an alternative payment system which doesn't involve any third party, namely a closed or internal payment system. It makes the customers does not depend to any bank. Examples of such environment are cafeterias, canteens, or food courts on campus.

Other advantages of using NFC-based payment system is that customers and merchants don't need to touch money (that may contain germs [3]), and merchants don't need more time to seek the change.

2. Related Works

2.1 IDA-Pay

Mainetti, Patrono, and Vergallo introduced an NFC-based payment system that does not need any access to SE (Secure Element) of Android device. It is called IDA-Pay. SE is a restricted storage area that only special application has access to it, namely Google Wallet. Google Wallet is limited to some affiliated credit card issuers, they are MasterCard and Visa. Hence IDA-Pay was developed. [12]

IDA-Pay system consists of a client's Android cellphone, a merchant's desktop computer, and a server that acts as a gateway to credit card networks, e.g. virtual POS (Point-Of-Sale) that supports EMV (Europay, MasterCard, and Visa). Merchant's desktop also acts as a relayer between the client and the server. [12]

Client's cellphone stores the information about client's credit card and his user ID. This information is asymmetrically encrypted by using IDA-Pay's public key, so that only the server can understand the content. This encrypted data along with the payment information will be sent when doing a transaction. To make it more secure, the encrypted data is also encrypted by using symmetric encryption (AES) before being stored in the phone memory. The key is derived from user's PIN. [12]

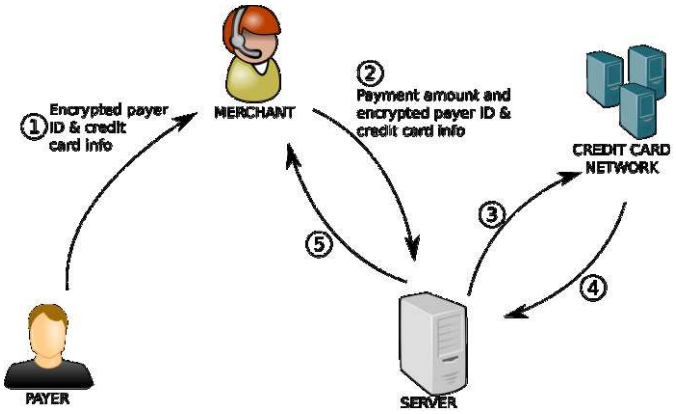

Fig. 1. Illustration of a payment in the IDA-Pay system.

2.2 Tag-to-Tag NFC Protocol

Husni *et al.* proposed an efficient tag-to-tag protocol for secure mobile payment. It is considered efficient because it uses symmetric cryptography only, there is no asymmetric cryptography, so it needs less resources for computation. For micropayment, the device can work offline. But for macropayment, the payer should confirm the payment through a message sent from the third party to increase security. [13]

In tag-to-tag protocol, a merchant is assumed has already had knowledge about the amount of transaction to be applied. So, the amount is sent from the merchant to the payer. However, in this protocol, both payer and merchant should do data transfer through NFC twice each other, which means there are four times data transfer. [13]

3. Design

Payment system in this research is called NFCafe. The architecture is a little similar to IDA-Pay. However, the data is not encrypted, but signed by using asymmetric cryptography (RSA), so that the merchant can read the message but can't modify it. It makes the data transferred transparent to the merchant, so there's no fraud which will harm the merchant. The signed part is customer's ID, amount of transaction, and timestamp of the transaction. The timestamp is intended to avoid, whether intentionally or not, double transactions.

Hence, the customer should have a key pair. In addition, the private key should be kept of stealing, so it is encrypted by using symmetric cryptography (AES) before stored into phone memory. The AES key is derived from customer's password.

The payment system consists of the following entity.

1. *Administrator*, acts as a cashier or bank and also administer the system in general.
2. *Payer* (could also be called as buyer or client), the customer that orders foods/drinks and do the payment.
3. *Merchant*, the entity that sells foods/drinks.
4. *System*, consists of three Android applications (each of them is for the Administrator, the Payer, and the Merchant), the web server, and NFC tags that contains foods/drinks menu sold by the Merchant. The Android applications are NFCafe Payer, NFCafe Merchant, and NFCafe Admin. The target API Level is 15 and the minimum supported API Level is 14.

In general, there are three kinds of activity among the entity above, they are registration, credit reload, and payment.

3.1 Registration

The registration of a new customer that will use NFCafe is done by the Payer to the Administrator as follows. The Payer registers himself and his device (Android cellphone) by sending data to Administrator's device through NFC. The data contains Payer's name, proposed ID, initial credit amount, and the public key. The ID is "proposed" because the given ID by the server may be different. After receiving Payer's data, Administrator's device sends a request along with Payer's data to the server through the internet. The server then sends a response back to Administrator's device. The response contains the status, whether success or not; if success there are also accepted Payer's ID and the credit amount/balance of the Payer (it would be the amount set by the Payer before). The response from the server is then forwarded to Payer's device by Administrator through NFC.

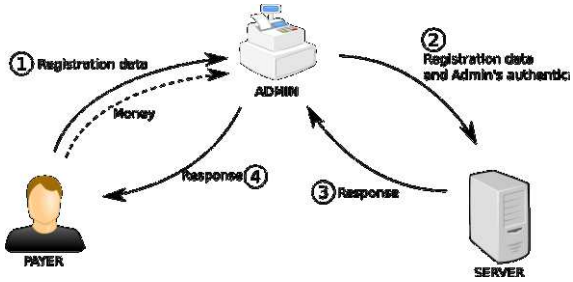

Fig. 2. The Payer registers himself and his device to the Administrator.

3.2 Credit Reload

An already registered customer (Payer) can do such a top up to the Administrator in the following steps. The Payer sends request data for reloading to Administrator's device through NFC. The data contains Payer's ID and the amount to be reloaded (added). After receiving Payer's data through NFC, the Administrator sends a reload request along with the Payer's data to the server through the internet. And then, the server sends the response back to Administrator's device. The response contains a status, whether success or not; if success it also contains the total credit amount of the Payer after addition. The response from the server is forwarded to Payer's device by the Administrator through NFC.

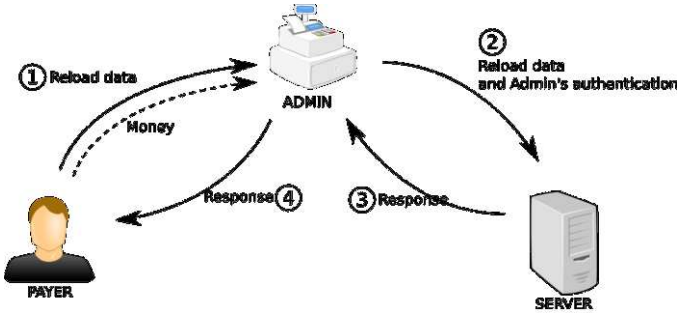

Fig. 3. The Payer reloads the credit through the Administrator.

3.3 Payment

An already registered customer can order foods/drinks to a Merchant in the following steps. The Payer read foods/drinks menu from an NFC tag by using his device. The NFC tag should contains NFCafe formatted message. It contains list of food/drink names and prices. After that, the Payer chooses any food/drink that appears in his device and do the order by beaming (sends data through NFC by using Android Beam as the user interface) to Merchant's device. The data contains Payer's ID, ordered foods/drinks prices, payment timestamp, and the signature. The Merchant can either accept or reject the request. If the Merchant accept it, then Merchant's device send a request along with Payer's request data to the server through internet. And then, the server sends a response back to Merchant's device. The response contains a status, whether success or not; if success it also contains the total credit amount of the Payer after the payment. The Merchant forwards the response to Payer's device through NFC.

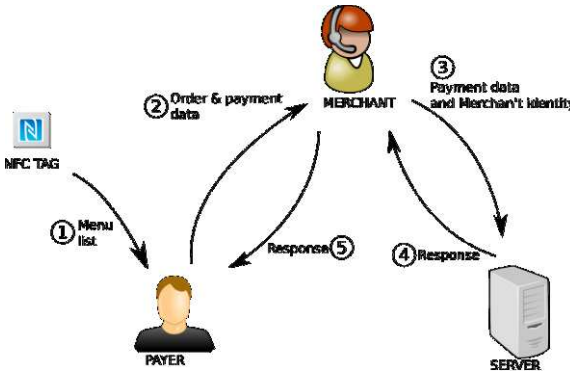

Fig. 4. The Payer reads foods/drinks menu from an NFC tag and does the payment to the Merchant.

4. Implementation

4.1 Message for Registration

A message sent from a Payer candidate to an Administrator for registration is composed of these fields:

1. Payer's proposed ID (16 bytes),
2. Initial credit amount (8 bytes, integer, big-endian),
3. Size of name field in bytes (1 byte, unsiged integer),
4. Name (UTF-8 encoded),
5. Public key.

A message sent from an Administrator to a Payer as the response of the message above if failure happened is an empty message. Otherwise, it is composed of total credit (8 bytes, integer, big-endian) and accepted Payer's ID (16 bytes). The total credit is the initial credit amount which was previously sent to the Administrator. Accepted Payer's ID is given by the server, it may be different from the proposed ID.

4.2 Message for Credit Reload

A message sent from a Payer to an Administrator for credit reloading is composed of Payer's ID (16 bytes) and amount to be reloaded (8 bytes, integer, big-endian).

A message sent from an Administrator to a Payer as the response of the message above if failure happened is an empty message. Otherwise, it is composed of total credit (8 bytes, integer, big-endian) after reloading.

4.3 Message for Payment

A message sent from a Payer to a Merchant for ordering foods/drinks and do the payment is composed of these fields:

1. Payer's ID (16 bytes),
2. Payment timestamp (8 bytes, integer, big-endian),
3. Total price (8 bytes, integer, big-endian),
4. Number of ordered foods/drinks (1 byte, unsigned integer),
5. Ordered foods/drinks list, consists of consecutive items, each of item consists of the price (8 bytes, integer, big-endian), size of the name in bytes (1 byte, unsigned integer), and the name of the food/drink (UTF-8 encoded),
6. Size of Payer's table number in bytes (1 byte, unsigned integer),
7. Payer's table number (UTF-8 encoded),
8. Signature that created from the sequential composition of Payer's ID, the payment timestamp, and the total price.

A message sent from Merchant to Payer as the response of the message above is composed of status (1 byte, integer – where 0 means failure and 1 means success) and total Payer's credit/balance (8 bytes, integer, big-endian) after payment.

4.4 Message in NFC Tag

An NFC tag containing foods/drinks list is NDEF formatted and contains consecutive items, each item consists the price (8 bytes, integer, big-endian), size of the name (1 byte, unsigned integer), and the name (UTF-8 encoded).

4.5 Screenshoots

Here is some screenshots of the developed applications.

Fig. 5. Screenshot of NFCafe Payer when the customer is about to register.

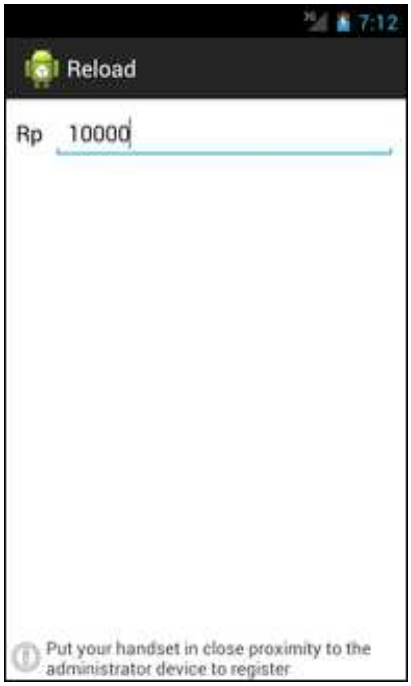

Fig. 6. Screenshot of NFCafe Payer when the customer is about to reload the credit.

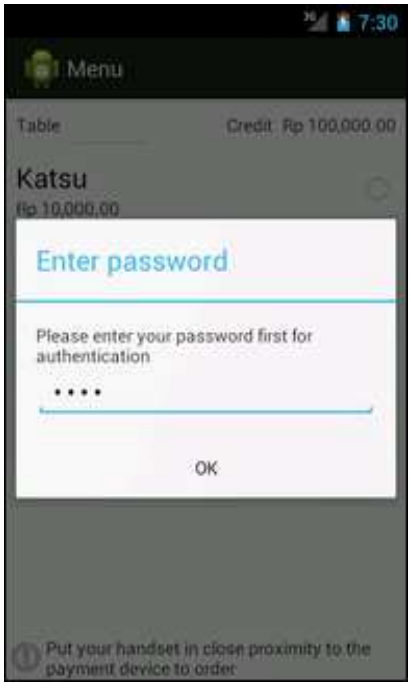

Fig. 7. Screenshot of NFCafe Payer when the customer enters the password before ordering and doing a payment.

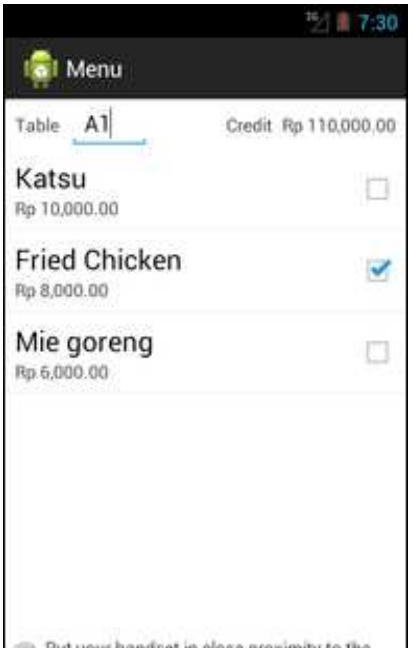

Fig. 8. Screenshot of NFCafe Payer when the customer chooses foods/drinks that has been read from an NFC tag.

The applications have been tested by several respondents who have been familiar with smartphones usage. The scenarios are as follows.

1. *Registration*, the Payer (candidate) does the registration to the Administrator. This test consists of some variations, such as the ideal condition, wrong input from the Payer, the condition with internet connection unavailable, and the condition that the Payer chooses to use no password.
2. *Credit reload*, the Payer reloads the credit to the Administrator. This test consists of some variations, such as the ideal condition, wrong input, and the condition which internet connection unavailable.
3. *Payment*, the Payers read menu from an NFC tag and then orders some food/drink to the Merchant. This test consists of some variations, such as the ideal condition, wrong user input, the condition which the Merchant rejects the order, and the condition which internet connection unavailable.
4. *Menu creation*.

The conducted tests are 100% success, which means all of the scenarios are done and given the results as expected.

5. Conclusion

The NFC technology in Android devices and the platform API can be used to do an NFC-based payment transaction. It has been implemented in this research and both symmetric and asymmetric cryptography are used. There is no respondent that complain about the applications performance, so it can be concluded that the use of asymmetric encryption doesn't cause a significant degradation of performance.

Android Beam is a user interface that make NFC transfer easy. However, the usage mechanism of Android Beam that only allows a one-way transfer for each beam makes the users less comfortable if they should do the beam multiple times for one transaction.

References

[1] S. Ortiz, "Is near-field communication close to success?", *Computer*, vol. 39, 2006, pp18-20.

[2] T. Korak and L. Wilfinger, "Handling the NDEF signature record type in a secure manner", *RFID-Technologies and Applications (RFID-TA), 2012 IEEE International Conference on*, 2012, pp107-112.

[3] M. Febrida, "Tempat yang Jadi Sarang Kuman saat Travelling", *Liputan6.com*, October 30, 2012, http://health.liputan6.com/read/448484/tempat-yang-jadi-sarang-kuman-saat-travelling.

[4] Android Open Source Project, "Android, the world's most popular mobile platform", *Android Developers*, n.d., http://developer.android.com/about/index.html.

[5] Open Handset Alliance, "Android Overview", *Open Handset Alliance*, n.d., http://www.openhandsetalliance.com/android_overview.html.

[6] Android Open Source Project, "Android Technical Information", *Android Open Source*, n.d., http://source.android.com/tech/index.html.

[7] Android Open Source Project, "Android Developers API Guides for API Level 16: <uses-sdk>", n.d..

[8] International Telecommunication Union, "Security in Telecommunications and Information Technology". n.a.: ITU, December, 2003, http://www.itu.int/itudoc/itu-t/85097.pdf .

[9] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for Key Management - Part 1: General (Rev. 3)", Gaithersburg: NIST, 2012, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf .

[10] RSA Laboratories, "RSA Laboratories' Frequently Asked Questions About Today's Cryptography", ver 4.1, n.a.: RSA Security Inc., 2000, http://www.rsa.com/rsalabs/faq/files/rsalabs_faq41.pdf .

[11] RSA Laboratories, "TWIRL and RSA Key Size", *RSA Laboratories*, 2003, http://www.rsa.com/rsalabs/node.asp?id=2004.

[12] L. Mainetti, L. Patrono, and R. Vergallo, "IDA-Pay: An innovative micro-payment system based on NFC technology for Android mobile devices", *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, 2012, pp1-6.

[13] E. Husni, K. Kuspriyanto, N. Basjaruddin, T. Purboyo, S. Purwantoro, and H. Ubaya, "Efficient tag-to-tag near field communication (NFC) protocol for secure mobile payment", *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2011, 2nd International Conference on*, 2011, pp97-101.

[14] NFC Forum, "NFC Forum Technical Specification", NFC Forum, n.d., http://www.nfc-forum.org/specs/spec_list/.

[15] M. Pasquet, J. Reynaud, and C. Rosenberger, "Secure payment with NFC mobile phone in the SmartTouch project", *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*, 2008, pp121-126.

[16] Android Open Source Project (n.d.). "Android Developers API Guides for API Level 16: NFC Basics."

[17] S. R. Pressman, "Software Engineering: A Practitioner's Approach", 7th ed., New York: McGraw-Hill, 2010.