An Interoperable Payment Protocol for the Public Transit Fare Payment System

SANGWOO LEE, YOUNGSAE KIM, JINMAN CHO, KYOIL JUNG
Information Security Research Division
Electronics and Telecommunications Research Institute
161 Gajeong-Dong, Yuseong-Gu, Daejeon 305-350
KOREA
{ttomlee, vincent, zmzo, kyoil}@etri.re.kr

Abstract: - The market for the public transit fare payment system using contactless smart cards is rapidly growing, however, the payment systems provided by different vendors are not interoperable. This paper presents an interoperable payment protocol for the public transit fare payment system using contactless smart cards. We also present implementation results of a PSAM (Purchase Secure Application Module - a secure device, typically, a chip that is embedded on the card terminal) that executes the proposed protocol to support interoperability among different contactless smart card based payment systems.

Key-Words: - Payment protocol, PSAM, Contactless smart card, Interoperability, Transit fare payment system

1 Introduction

Worldwide, we are using a lot of smart cards in electronic payment systems. Especially, the market for the public transit fare payment system using contactless smart cards is rapidly growing. For example, many contactless smart cards are used to pay public transit fare in South Korea; however, different smart cards and PSAMs provided by different electronic cash vendors are not interoperable. This means that cardholders cannot use cards issued by one electronic cash vendor in card terminals provided by other vendors because different PSAMs have different functions, cryptographic algorithms and authentication mechanisms.

Many specifications are available that support interoperability in payment systems using smart cards. Among them, CEPS (Common Electronic Purse Specifications) defines requirements for all components in payment systems to implement a globally interoperable electronic purse schemes [3, 4, 5]. CEPS requires compatibility with the EMV specifications and defines the requirements for an interoperable card application, the card-to-terminal interface, the terminal application for point-of-sale and load transactions, data elements, and recommended message formats for transaction processing. CEPS also provides functional requirements for electronic purse scheme participants and uses public key cryptography for enhanced security.

However, CEPS is not appropriate for public transit fare payment systems using contactless smart cards. CEPS uses public key cryptographic algorithms for mutual authentication between electronic purses and PSAMs and has too many passes in a purchase procedure. Such characteristics hinder fast transaction, which is one of the most import prerequisites for public transit fare payment systems using contactless smart cards.

In this paper, we propose a new payment protocol to support interoperability among different electronic purses and PSAMs issued by different vendors specifically for the public transit fare payment system. In the proposed protocol, a PSAM manages security key sets and balances classified by identifiers of electronic cash vendors to support interoperability. Then we implemented a PSAM that executes the proposed protocol on an AT90SC6464C, which is 8-bit microcontroller based on the AVR RISC architecture for smart cards. This work will be a good practical example of standardization for the public transit fare payment system using contactless smart cards.

The organization of this paper is as follows. In Section 2, we propose a new payment protocol to support interoperability. We describe how to design the PSAM software architecture and implementation results in Section 3. Finally, concluding remarks are found in Section 4.

2 A New Payment Protocol

2.1 Purchase Protocol

We propose a new payment protocol for the public transit fare payment system using contactless smart cards. Entities in the proposed protocol are defined as follows:

- **PSAM (Purchase Secure Application Module)** is a secure device, typically, a chip that is embedded on the card terminal. The PSAM contains security keys, authenticates a smart card during purchase transaction, and stores the transacted fare from the card.
- **EP (Electronic Purse)** is a contactless smart card which has a unique identifier, secure keys for payment procedure, and pre-paid balance for fare payment.
- **CT (Card Terminal)** is a device that can detect an electronic purse, communicate with the electronic purse following ISO/IEC 14443, and transfer messages between the electronic purse and a PSAM.

Notation	Meaning
$\|\|$	Concatenation
{Data, Key}	Generated MAC(Message Authentication Code) with Key
ID_{CENTER}	An identifier of electronic cash vendor
ID_{EP}	An identifier of EP
NT_{EP}	Number of transaction of EP
R_{EP}	Random number of EP
M_{PDA}	Purchase device transaction amount
BAL_{EP}	Balance of EP
ALG_{EP}	An algorithm identifier of EP
VK_{EP}	Version of secure keys
SES_{EP}	Session key generated by EP
ID_{PSAM}	An identifier of PSAM
NT_{PSAM}	Number of transaction of PSAM
SC_{PSAM}	Status code of PSAM
NI_{PSAM}	Number of individual transaction of PSAM
BAL_{PSAM}	Balance of PSAM
SES_{PSAM}	Session key generated by PSAM
DPK	Derived purchase key
PK	Purchase key
INDK	Individual Transaction Key

Table 1. Notation and its meaning

Some notations to describe the protocol are shown in Table 1.

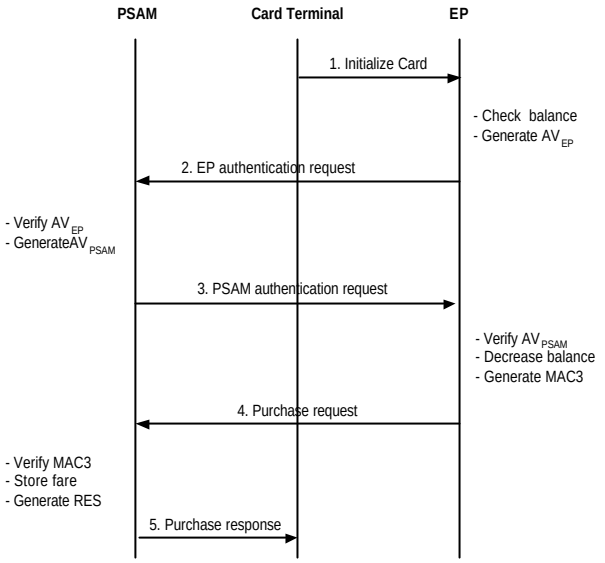

Fig.1. An overview of purchase protocol

Fig. 1 shows an overview of purchase protocol. The protocol goals can be stated as follows:

- Mutual authentication of an electronic purse and a PSAM
- Fare payment from an electronic purse to a PSAM
- Interoperability of a PSAM with different smart cards developed by different vendors

An electronic purse and a PSAM authenticate each other by showing knowledge of a secret key-PK (Purchase Key), which is shared between them when they are issued. The PSAM in the proposed protocol has the PKs and the balances classified by identifiers of different vendors. The detailed procedure of the proposed protocol is as follows:

1) CT => EP: Initialize Card

CT detecting EP sends Initialize Card message including M_{PDA}, which is transit fare in this protocol. EP receiving Initialize Card from CT executes the following steps:

- Check if BAL_{EP} is larger than received M_{PDA}.
- Generate R_{EP} and increase NT_{EP} if BAL_{EP} is larger than M_{PDA}.
- Generate a session key SES_{EP} = {ID_{CENTER} || ID_{EP} || NT_{EP} || R_{EP}, DPK} and compute MAC1 = {ALG_{EP} || VK_{EP} || BAL_{EP}, SES_{EP}}, where DPK is derived from PK by off-line.

The PK is a shared long term secrete key between EP and PSAM.

2) EP => PSAM: EP authentication request

EP sends AV_{EP} to PSAM through CT.

$AV_{EP} = \{ALG_{EP} \| VK_{EP} \| BAL_{EP} \| ID_{CENTER} \| ID_{EP} \| NT_{EP} \| R_{EP} \| M_{PDA} \| MAC1\}$

PSAM receiving AV_{EP} executes the following steps:

- Check if SC_{PSAM} indicates normal status or not.
- Derive $DPK = \{ID_{CENTER} \| ID_{EP}, PK\}$ from its own PK with received ID_{CENTER} and ID_{EP} if SC_{PSAM} indicates normal status.
- Generate $SES_{PSAM} = \{ID_{CENTER} \| ID_{EP} \| NT_{EP} \| R_{EP}, DPK\}$ with received values and DPK, which is derived by itself.
- Compute $MAC1' = \{ALG_{EP} \| VK_{EP} \| BAL_{EP}, SES_{PSAM}\}$ and compare it with received MAC1.
- Increase NT_{PSAM} and generate $MAC2 = \{M_{PDA} \| ID_{PSAM} \| NT_{PSAM} \| R_{PSAM}, SES_{PSAM}\}$.
- Record a transaction log that is composed of ID_{EP}, NT_{EP}, and M_{PDA}.

3) PSAM => EP: PSAM authentication request

PSAM sends AV_{PSAM} to EP.

$AV_{PSAM} = \{ID_{PSAM} \| NT_{PSAM} \| R_{PSAM} \| SC_{PSAM} \| MAC2\}$

When receiving AV_{PSAM}, EP executes the following steps:

- Generate $MAC2' = \{M_{PDA} \| ID_{PSAM} \| NT_{PSAM} \| R_{PSAM}, SES_{EP}\}$ and check if MAC2' matches with MAC2 received from PSAM.
- Deduct BAL_{EP} by M_{PDA}.
- Generate $MAC3 = \{ID_{PSAM} \| NT_{PSAM} \| BAL_{EP}, SES_{EP}\}$.
- Record a transaction log that is composed of ID_{PSAM}, NT_{EP}, and M_{PDA}.

4) EP => PSAM: Purchase request

EP sends MAC3 to PSAM. On receiving MAC3, PSAM executes the following steps:

- Verify if the received MAC3 matches with MAC3' computed by itself.
- Increase BAL_{PSAM} and NI_{PSAM} by M_{PDA} and by one respectively. BAL_{PSAM} and NI_{PSAM} are distinguished by ID_{CENTER}.
- Set SC_{PSAM} as normal status.
- Generate $MAC4 = \{ID_{CENTER} \| ID_{EP} \| NT_{EP} \| BAL_{EP} \| M_{PDA} \| ID_{PSAM} \| NT_{PSAM} \| NI_{PSAM} \| BAL_{PSAM}, INDK\}$, which is used to inform the completion of purchase transaction.

5) PSAM => EP: Purchase response

PSAM sends RES to inform the completion of purchase transaction to CT.

$RES = \{ID_{CENTER} \| ID_{EP} \| NT_{EP} \| BAL_{EP} \| M_{PDA} \| ID_{PSAM} \| NT_{PSAM} \| NI_{PSAM} \| BAL_{PSAM} \| MAC4\}$

Finally, CT saves the received RES from EP in its memory.

When errors occur in the fourth message (purchase request) of the purchase protocol, the electronic purse deducts BAL_{EP} by M_{PDA}, but the PSAM cannot increase BAL_{PSAM}. To solve this problem, we modify the purchase protocol to handle errors. The modified procedure is the same as the purchase protocol in normal except for several steps. The differences between the procedures in normal mode and in error handling mode are described here. A comparison step between the transaction log of PSAM and that of EP is added in the functions of PSAM and EP in the mutual authentication phase for PSAM to confirm that the current EP is the one that executed the previous purchase protocol with the fourth erroneous message. In detail, AV_{EP} in the second message (EP authentication request) should be changed as follows:

$AV_{EP} = \{ALG_{EP} \| VK_{EP} \| BAL_{EP} \| ID_{CENTER} \| ID_{EP} \| NT_{EP} \| R_{EP} \| \text{Transaction log} \| MAC1\}$

On the receipt of AV_{EP}, PSAM checks if the received transaction log matches with the transaction log of its own. After the comparison step succeeds, EP maintains BAL_{EP} without reducing it by M_{PDA} because BAL_{EP} was already deducted in the previous purchase procedure. PSAM executes the same steps to increase BAL_{PSAM} in normal mode.

2.2 Key Management

The PSAM in the purchase protocol described in Section 2.1 has 2 kinds of keys to generate MACs. The length of each key is 16 bytes. PK is used to derive DPK that is needed to generate a session key in purchase transaction. INDK is used to generate MAC4 during purchase transaction. To support interoperability, the PSAM stores and manages key sets, BAL_{PSAM}, and NI_{PSAM} indicated by ID_{CENTER}. In

purchase transaction, the PSAM receives ID_{CENTER} in the second message AV_{EP}. The PSAM, then, selects security keys sorted by received ID_{CENTER} and generates MACs for mutual authentication. The PSAM also increases BAL_{PSAM}, indicated by the ID_{CENTER}, by M_{PDA}. As a result, the PSAM can classify each fare from different electronic purses according to vendors and store it separately.

3 Design and Implementation

We describe a practical implementation of the PSAM that executes the proposed payment protocol in this section.

3.1 Message Format

Between a PSAM and a card terminal, serial communication methods are used. The message format is described in Fig. 2. LEN is the length in bytes from LEN to API DATA. NAD is a field that indicates pre-issued applications of vendors. It is needed to allow pre-issued public transit fare cards in the proposed system. API DATA consists of CLA (instruction class identifier), INS (instruction identifier), and DATA (instruction specific information). Checksum is a value of CRC-16 for the data from LEN to API DATA.

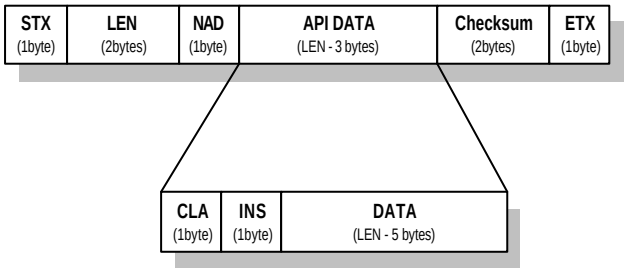

Fig. 2. Message Format

3.2 PSAM software architecture

Fig. 3 describes the data flow model between an electronic purse and a PSAM. A PSAM and a card terminal communicate in serial communication method. The interface between a card terminal and an electronic purse follows ISO/IEC 14443 [8].

The PSAM and the electronic purse have their own cryptographic modules to generate MACs and to compute session keys and CRC functions to check errors in transmission, and instruction processing units to perform specific instructions. The PSAM and the

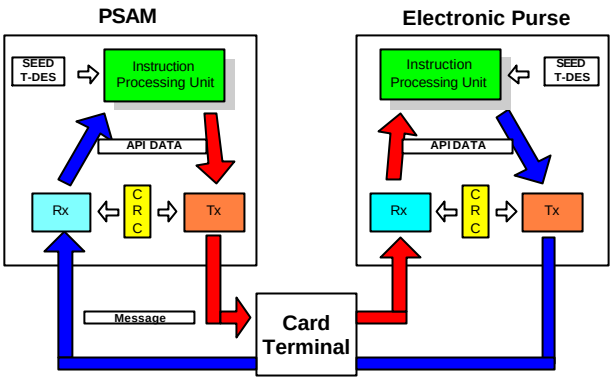

Fig. 3. Data flow model of proposed payment system

electronic purse have layered architecture with the following two layers:

- **Transport layer** sends and receives a message in Fig. 2, verifies errors in transmission using CRC-16, and sends API DATA of a message to the application layer.
- **Application layer** classifies an API DATA from the transport layer according to CLA and INS fields, executes the instruction, and sends the response to the transport layer.

Since the payment system has layered software architecture, it is easy to manage and update internal instruction processing units and to adopt different protocols in the transport layer, such as block-oriented protocol, in order to speed up transmission rate. There exist three modules in the transport layer. Rx modules check if errors occur in transmission using CRC-16 and verify if the received message follows the message format in Fig. 2. Tx modules append STX, ETX, and the CRC result of a message to be sent to the API DATA received from the instruction processing units and send the formatted message to the card terminal in byte-oriented transmission protocol. In the application layer, the payment system has cryptographic modules and the instruction processing units. Cryptographic modules perform either SEED or Triple-DES according to the values of ALG_{EP}. The PSAM and an electronic purse use Triple-DES or SEED to authenticate mutually. SEED is a national 128-bit block cipher standard in South Korea and its global standardization is in progress at ISO/IEC JTC 1/SC 27 [14, 16, 17]. We use those symmetric cryptographic algorithms in CBC (Cipher Block Chaining) mode to generate MACs and session keys, and to derive DPK from PK. The instruction processing units classify the API DATA from Rx

modules by CLA and INS fields and execute the instructions such as generation or verification of authenticated vectors, increment or decrement of balances, and management of keys.

3.3 Implementation Results

We implemented the PSAM on an AT90SC6464C. The AT90SC6464C is based on the 8-bit AVR RISC architecture and has 64 Kbytes of Flash program memory, 64 Kbytes of EEPROM user memory, and 2.5 Kbytes of RAM [11].

We used the hardware accelerator for Triple-DES provided by the AT90SC6464C. We implemented CRC functions and SEED in software using the AVR core. The code size of the PSAM software is 22.7 Kbytes. The execution of the purchase protocol takes around 103ms on the implemented PSAM. This execution time is short enough for the public transit fare payment system using contactless smart cards.

	Feature
Code Size	22.7 Kbytes
Excution time of Purchase transaction	103 ms

Table 2. Feature of the implemented PSAM

4 Concluding Remarks

Public transit fare payment systems in South Korea are good examples for application of contactless smart cards. However, different cards provided by different vendors are not interoperable with different PSAMs on card terminals. In this paper, we introduced a new payment protocol to enable those cards to be read by one PSAM on any card terminals. The PSAM in the protocol that we proposed can communicate with different electronic purses developed by different providers and manage security key sets and balances classified by identifiers of vendors in order to supply interoperability.

We also presented a practical implementation of this PSAM, which executes the proposed protocol on an AT90SC6464C. The execution time of purchase protocol is about 103ms, which is appropriate for the payment of public transit fare that requires promptness. The proposed payment protocol and the implemented PSAM can also be useful for micro-payment systems in different environments.

References:

[1] Electronics and Telecommunications Research Institute, A Study for Expansibility of Electronic Payment system for Public Transit Fare, Technical Report, December 2001.

[2] EMV2000, Integrated Circuit Card Specifications for Payment Systems, version 4.0, December 2000.

[3] CEPSCO, LLC, Common Electronic Purse Specification (CEPS), Business requirements, version 7.0, March 2000.

[4] CEPSCO, LLC, Common Electronic Purse Specification, Functional Requirements, version 6.3, September 1999.

[5] CEPSCO, LLC, Common Electronic Purse Specification, Technical Specification, version 2.3, March 2001.

[6] ISO 10202-2, Financial transaction cards - Security architecture of financial transaction systems using integrated circuit cards - Part 2: Transaction process, 1999.

[7] ISO 10202-4, Financial transaction cards - Security architecture of financial transaction systems using integrated circuit cards - Part 4: Secure application modules, 1999.

[8] ISO/IEC 14443, Identification cards - Contactless integrated circuit(s) cards - Proximity cards, 2000.

[9] Europay International, PBS A/S and Visa International Service Association, Terminal Architecture for PSAM Applications, Overview, version 2.0, April 2000.

[10] ISO/IEC 7816-3, Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols," 1997.

[11] Atmel Corporation, AT90SC6464C Data Sheet, 2001.

[12] W. Rankle and W. Effing, Smart Card Handbook, John Wiley and Sons, Ltd, 1999.

[13] A. J. Menezes, P. V. Oorschot, and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[14] Telecommunications Technology Association, 128-bit Symmetric Block Cipher (SEED), 1999.

[15] ISO/IEC 10116, Information technology - Security techniques - Modes of operation for an n-bit block cipher, 1997.

[16] ISO/IEC JTC 1/SC 27 N 2975rev1, ATT. 2 Korean contribution, Performance and Implementation Cost of SEED, October 2001.

[17] ISO/IEC JTC 1/SC 27 N3213, Third Party Evaluation on SEED by CRYPTREC, April 2002.