An Open, PKI-Based Mobile Payment System

Marko Hassinen, Konstantin Hyppönen, and Keijo Haataja

University of Kuopio, Department of Computer Science
POB 1627, FIN-70211, Kuopio, Finland
{Marko.Hassinen, Konstantin.Hypponen, Keijo.Haataja}@uku.fi

Abstract. Most mobile commerce applications require a secure mobile payment solution for performing financial transactions. However, it is difficult to strongly authenticate users remotely and provide non-repudiation of transactions. In this paper, we present a novel mobile payment scheme which supports both virtual point-of-sale (POS) and real POS transactions. For user authentication, our scheme uses PKI-SIM cards. In virtual POS payments, the mobile phone communicates with a service provider through SMS messaging or IP-based data transfer (e.g. GPRS). In real POS payments, Bluetooth is used as the communication channel. Communication with a bank is done using either SMS messaging or IP-based data transfer. The system is open to any mobile network operator, any merchant, and any financial institution.

1 Introduction

Mobile payment (MP) is a potential killer application in future mobile networks. Active development efforts in the mobile payment domain are boosted by tough competition between mobile network operators, financial institutes, and payment service providers. The common goal is enhancing customer service by providing new payment solutions, with the hope to grow the customer base and, ultimately, increase revenues.

In the third quarter of year 2005 there were 2.03 billion mobile phone users globally [1]. In many countries it is more common for people to have a mobile phone than to have a credit card. Considering vending machine payments, it is often easier for a user to have their mobile phone in a pocket, rather than a suitable set of coins or banknotes. Mobile payment is therefore an attractive service for many mobile network users. Moreover, in some cases switching to mobile payments provides benefits not only to customers, but also to the service providers. For example, vending machines that accept cash or credit cards have moving parts and thus require regular maintenance. Besides that, someone has to remove money from the machine regularly.

Various mobile payment scenarios have been devised (see [2] for a survey), and many of them are now in active commercial use. However, expectations that exist among participants of an MP service are rather divergent, and meeting them have proved to be difficult. It is very often, for instance, that an MP solution deals only with micro-payments or mini-payments (less than $20). Services that can process

G. Müller (Ed.): ETRICS 2006, LNCS 3995, pp. 86–100, 2006.

macro-payments usually involve independent MP providers that act as mediators between mobile network operators and credit companies. This increases the cost of a payment, and makes the system less transparent from the user's perspective, decreasing trust. Moreover, most of the solutions support only virtual point-of-sale (POS) transactions, while customers expect them to work at a real POS. Many payment models rely on the traditional Short Message Service (SMS) as the carrier of payment-related data. However, SMS messages can be forged [3], potentially making the MP solution untrustworthy and insecure.

Our results: This paper describes a mobile payment scheme that is based on a governmental Public Key Infrastructure (PKI). The scheme does not involve any MP mediator. Two mobile payment protocols are presented: a protocol for virtual POS payments, and a protocol for real POS (or vending machine) payments.

Before proceeding with more detailed description of our MP scheme, we give an overview of existing MP solutions. We concentrate in more detail on a few examples of schemes that either provide the same functionality, or use similar ideas in their implementation.

1.1 Related Work

The simplest mobile payment schemes are based on calling a premium phone number or sending a premium SMS message. The amount of payment is then charged on the phone bill. A drawback of this approach is that there is no possibility to authenticate the phone user. This creates a problem for instance in the case when a phone gets stolen. Since the system does not provide any kind of non-repudiation, users may argue that they have not used their phones for making payments. Because of this particular reason current systems have been limited to products with small monetary value, such as newspapers, candies or lemonade.

The risks related to poor authentication and non-repudiation are mitigated in many schemes by introducing an extra MP provider that takes care of them. Users sign up for the MP services and either establish a pre-paid account within the MP provider system, or register a debit/credit card to be charged for future payments. In addition to better support for macro-payments, this solution provides flexibility: users of different mobile networks and even from different countries can use the same MP provider company. A drawback of such schemes is that payment mediators charge an extra premium, making use of the system more expensive. Moreover, handling pre-paid accounts used in many schemes and controlling balance on them is an extra burden placed on users of the system.

Mobipay[1] is a typical example of a payment mediator. It works currently in Spain, and is expected to be introduced in other countries in the future. Mobipay works for both virtual and real POS payments. In a real POS payment the merchant enters the user's identifier (their phone number or alias), or scans the barcode attached to the user's phone. Details of the payment, such as the price of purchased products and the name of the shop, are then sent to the user's phone. USSD (Unstructured Supplementary Service Data, a session-oriented version of

[1] http://www.mobipay.com

SMS) is used as the channel for sending this message. The user confirms the payment by entering their PIN code associated with Mobipay.

In a virtual POS payment on the Internet, the customer gets the reference number that has to be entered at the mobile phone along with the PIN code. The user sends the message to Mobipay, after which both the user and the merchant receive a confirmation of the payment.

Mobipay can also handle vending machine and invoice payments, which are performed in a similar way. Payments are charged to a debit or credit card, or to the user's pre-paid account.

In Mobipay, the network operator is a trusted party. However, it does not produce sufficient evidence for later adjudication in case if a dispute arises. Users can theoretically repudiate transactions, claiming that they never entered the PIN code, and that everything was generated by the network operator. This problem must be solved by additional security measures, for example, by introducing a reliable storage for transaction logs on the network operator side, and keeping it under control of a trusted auditor.

MP schemes that use extra PIN codes for confirmation of payments provide better authentication than those based on simple SMS messaging. Stolen or lost mobile phone is not a problem in these systems. An exception, however, is the case when the attacker first learns the PIN code for mobile payments by shoulder surfing, and then steals the phone.

To provide stronger authentication and non-repudiation, some systems use mobile PKI. Several smart card vendors manufacture SIM cards with PKI capabilities, providing an off-the-shelf solution for mobile network operators. Mobile PKI can be used as a way to strongly authenticate a user in numerous mobile applications [4]. A message with a valid digital signature can be used to show commitment, and it can provide non-repudiation. Confidentiality of the communication can be also guaranteed.

SmartPay (MobilHandel)[2] is an MP scheme developed by Telenor, a Norwegian mobile operator. SmartPay uses mobile PKI for authentication and non-repudiation. The certificate of each user is stored in their SIM card, in a PKI application implemented as a SIM toolkit applet. SmartPay can handle virtual POS payments of orders created by SMS messages or by browsing the merchant's WAP pages. When the merchant receives an order, it sends a request to Telenor mCommerce PKI-server to confirm the payment. The user is identified by their mobile phone number. The PKI-server generates an SMS request with a transaction value for the payment and sends it to the user's phone. In the phone the request is forwarded to the PKI application. At this step the user can choose the means of payment: phone bill or a credit card registered in the system. The transaction value is signed using the user's private key and sent back to the mCommerce PKI-server. Upon successful verification of the signature, the server sends confirmations of the payment to both the merchant and the user.

In some MP schemes not only the GSM functionality of the mobile phone is used, but also other communication technologies that are implemented in either the

[2] http://telenormobil.no/mobilhandel

phone or the SIM card. Short-range communication channels are used for payments at a cash register or for exchange of electronic cash between two mobile phones. An incomplete list of these channels contains Bluetooth, infrared ports (IrDA), and direct wireless connection to the SIM card. Other prominent technologies are RFID and NFC; however, we are not aware of any current MP schemes that use them.

An example of such systems is *Beamtrust*[3], an MP system developed in Denmark. It supports in-store payments and allows to withdraw money from ATMs. The mobile phone uses Bluetooth or infrared link for communication with the cash register or the ATM. In the case of an in-store payment, the user brings their mobile phone in a close proximity to the payment terminal. The total price of purchased goods is transmitted to the phone via Bluetooth or IrDA and shown on the screen. The user accepts the payment by entering their PIN code.

Although functionality of our system is similar to that of Mobipay, and the use of mobile PKI corresponds to that in SmartPay, the ideology of our MP scheme is rather different. Instead of relying on independent payment processing companies or on agreements between mobile network operators and credit card companies, our system uses governmentally controlled PKI. The SIM card contains a certificate issued by the Population Register Centre (PRC) of Finland. The certificate database maintained by PRC is freely accessible to everybody. Therefore, our system is open to any mobile network operator, any merchant, and any credit card company.

The rest of the paper is organized as follows. Section 2 gives a short introduction to the technologies used in our MP scheme. Section 3 provides an overview of the mobile payment model and describes two protocols for mobile payments: one for virtual POS payments, and another one for real POS payments. We further discuss the protocols and provide their security and privacy analysis in Sect. 4. Finally, Sect. 5 summarizes our conclusions.

2 Underlying Technologies

This section gives an overview of technologies used in the design of our MP system. We describe the public key infrastructure, communication technologies, and issues related to certificate validity assurance.

2.1 FINEID

For authentication of users, we use PKI provided by the Finnish Population Register Centre [5]. The centre issues electronic identity cards that contain three certificates:

1. Card holder's authentication and encryption certificate;
2. Card holder's non-repudiation certificate;
   (The `key usage` objects of these two certificates define different key usage policies; otherwise certificates are technically the same.)
3. Population Register Centre's own Certification Authority (CA) certificate.

[3] http://www.beamtrust.com

The card holder's private keys are stored in the memory of this tamper resistant card. There are no other copies of these keys, and it is practically impossible to manufacture duplicates of the card. This suits perfectly our requirements for authentication and non-repudiation.

Finnish Electronic Identification (FINEID) [5] application manages the contents of the electronic identity card and provides a command interface for performing private key operations. The card authenticates its user by a PIN code.

Population Register Centre maintains an online certificate directory (FINEID directory). Each registered individual gets a unique Finnish Electronic User ID (FINUID). Public keys of each user can be downloaded upon a search with appropriate criteria. Besides that, revocation list of invalid certificates is available from the FINEID directory.

Recently, it has become possible to include the FINEID functionality on SIM cards for mobile phones. In our MP scheme SIM cards perform digital signature and decryption operations, whereas encryption and signature verification are done by the mobile phone. Validity of certificates used in the MP scheme is checked upon the FINEID directory.

2.2 Bluetooth and NFC

Bluetooth [6] is a technology for short range wireless data and two-way voice transfer providing data rates up to 3 Mb/s. It operates at 2.4 GHz frequency in the free ISM-band (Industrial Scientific Medicine) using frequency hopping, and is supported by a wide range of various devices. The price of a Bluetooth chip has become reasonable and it is very common in modern mobile phones. In our MP scheme, Bluetooth is used as a communication channel in vending machine payments.

Although all data exchanged via Bluetooth is encrypted using built-in encryption with 128-bit keys, we use Bluetooth as an untrusted transport media. All sensitive data is encrypted on the application level. Integrity and freshness of messages is ensured by digital signatures, timestamps, and nonces.

In real POS payments, Near Field Communication (NFC) could be used instead of Bluetooth or in addition to it. The benefit of using NFC is its shorter working distance (about 20 cm). In places where POS terminals are placed close to each other NFC provides an easier way for ensuring that a proper terminal is contacted. NFC can be also used to initiate and configure the Bluetooth communication. The drawback, however, is that NFC is still supported only by a few devices, whereas Bluetooth is already widespread.

2.3 J2ME and SATSA

We propose to use Java 2 Micro Edition (J2ME) [7] as the programming platform for implementing the mobile phone part of the MP application. Theoretically, other platforms could also be used, as long as they provide a way to access extended features of the SIM card (the FINEID application). In J2ME, this is achieved by an optional package, *The Security and Trust Services API* (SATSA) [8]. Among other features, the SATSA specification defines methods for communication with applications on the SIM card, by exchanging messages in the APDU format [9].

A number of new mobile phones support features defined by the SATSA specification. Expectedly, this number will grow in the near future.

2.4 Secure Message Exchange

To provide confidentiality, authentication and non-repudiation of messages that constitute a payment transaction, messages are encrypted and signed. Figure 1 depicts the secure message exchange scheme, showing the process of delivering a message from the vending machine to the mobile phone. The same scheme is used also if a message is sent from the bank; however, SMS or IP-based data transfer is used instead of Bluetooth in this case. Operations with the private key of the mobile user are performed on the SIM card. If a message is originating from the mobile phone, FINEID application on the SIM card signs the message.

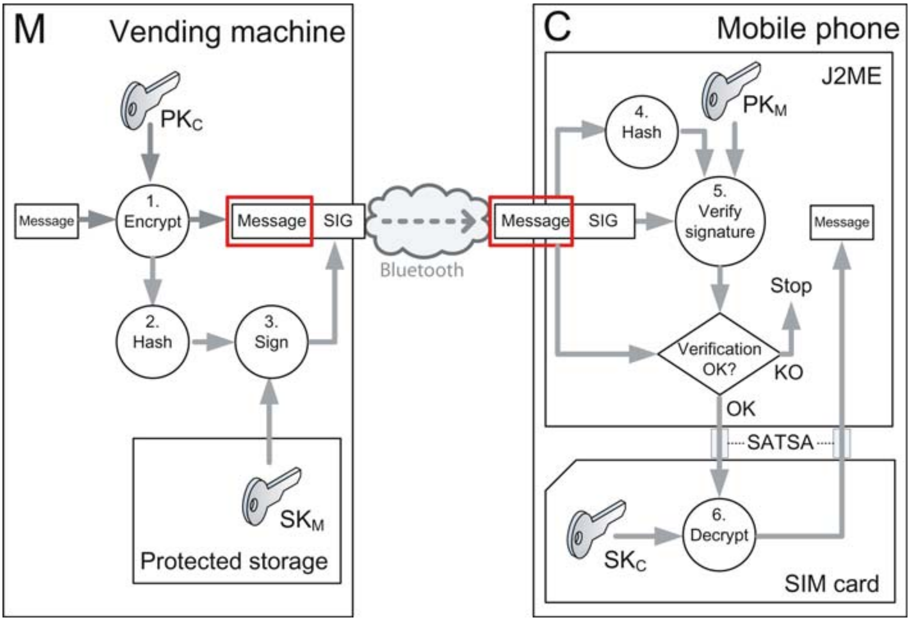

Fig. 1. An example of secure message exchange. SMS or IP-based data transfer can be used instead of Bluetooth.

2.5 Certificate Validity Assurance

With mobile devices, one clear restriction is the amount of available permanent storage. With the traditional certificate revocation list (CRL) approach we have to download the full CRL to the client, and the amount of storage needed for this may be too big for a mobile device. Moreover, downloading the full list to a mobile device can be rather slow and expensive to the user.

Protocols such as OCSP (Online Certificate Status Protocol) [10] and DPV (Delegated Path Validation) [11] can be used to offload most of the validation process from the client to a server. These protocols relieve the client from downloading the bulky CRL. However, as in some cases a POS terminal does not have a connection to the Internet, it cannot use the abovementioned protocols. We propose a solution where clients (mobile phones) provide proofs of the validity of their certificates. The client can request such for their certificate from the OCSP server. The proof contains the status of the certificate and it is digitally signed (see 2.2 in [10]). The POS terminal can then verify this signature and be confident that the certificate had the stated status at the time the proof was issued.

One problem still remains, since the POS terminal cannot get a current timestamp from the OCSP server. This means that it might be possible for a client to replay an old OCSP token in a fresh message. To avoid this we use a challenge-response scheme, where the terminal sends the client a challenge for which the client has to show a timestamped response by the same OCSP server. For simplicity we propose to have one signature which ties together the response, the message, the timestamp, and the certificate validity statement.

3 Payment Scheme

Our mobile payment scheme includes the following parties. A *customer* is a user of a hand-held device. The customer has received a SIM card with the FINEID applet, which includes the public key certificate of the user and a corresponding private key. Identity of the customer is their Finnish Electronic User ID (FINUID).

A *merchant* is an owner of a point-of-sale terminal (or a vending machine) or a service provider that accepts mobile payments. The merchant has a secret key and a corresponding public key certificate registered in the FINEID system.

A *bank* or another credit organization like VISA or MasterCard is a financial institution that acts as a payment processor. The customer has an account in the bank, or has been issued with a credit card operated by it. If the customer has multiple accounts or credit cards within the bank, the bank has been informed which of them should be used for mobile payments. The bank has the right to charge the customer's account or credit card when presented with a payment order signed by the customer's private key.

In this section we describe two mobile payment protocols: one for a virtual POS payment, and another for a real POS (vending machine) payment. The following notation is used in the description: C is a customer, M is a merchant, and B is a bank. ID_X is the identity of subject X. SK_X is the secret RSA key of subject X, and PK_X is the corresponding public key. Cert_X is the public key certificate of subject X. $\{m\}_K$ denotes RSA encryption of the message m under the key K. SIG_{X_Y} is a digital signature generated by X, intended to be verified by Y. H is a hash function; we use SHA-1 in our protocol.

3.1 Virtual POS Payment

Our protocol for a virtual POS payment contains the following steps (Fig. 2):

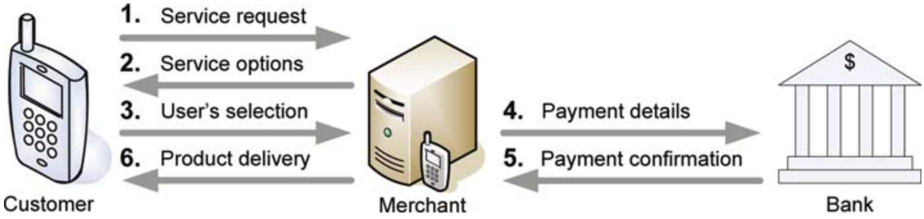

Fig. 2. Virtual POS Payment Model

1. Service request. In phase 1, the customer initiates the protocol with the merchant by requesting product options. The request may contain information which limits possible options.

$$C \xrightarrow{\text{Service request}} M$$

2. Service options. In phase 2, the merchant sends a list of options to the mobile device. The list includes short descriptions of products and pricing information. The merchant also attaches its certificate to the list of options.

$$M \xrightarrow{\text{Service options}|\text{Cert}_M} C$$

3. Product selection. In phase 3, the customer is prompted by the mobile device to select a product from the list. The information on the customer selection is sent to the merchant. The selection is signed using the private key of the customer. The message is

$$C \xrightarrow[\text{SIG}_{C_B}=\{H(TS_C|ID_M|AM)|H(PD|N_C)\}_{SK_C}]{\text{MSG}=\left\{PD|N_C|TS_C|\{H(PD|TS_C)\}_{SK_C}|ID_B|\text{SIG}_{C_B}\right\}_{PK_M}} M$$

where N_C and TS_C are a random nonce and a timestamp generated by the customer C, AM is the amount of money the purchase will cost and PD is a string that describes product details.

4. Payment request. Phase 4 of the protocol includes the merchant sending the payment details to the credit company. This payment information is signed using the merchant's private key and encrypted using the public key of the credit company. The message in phase 4 includes merchant's details and payment details, such as amount, id of the customer and the signed message received in phase 3.

$$M \xrightarrow[\text{SIG}_{M_B}=\{H(ID_M|ID_C|TS_C|AM|H(PD|N_C))\}_{SK_M}]{\text{MSG}=\left\{ID_M|ID_C|TS_C|AM|H(PD|N_C)|\text{SIG}_{M_B}|\text{SIG}_{C_B}\right\}_{PK_B}} B$$

In this message SIG_{C_B} is the same signature as in phase 3. After receiving this message the credit company B checks that the timestamp TS_C is newer than the timestamp of the previous communication to detect any replay attacks.

It is possible for the merchant to sign a contract for processing mobile payments with a single acquiring bank. In this case M sends the message to the acquirer. The acquirer has to pass the message to B, receive the payment confirmation (see step 5) and forward it to M.

5. Payment confirmation. The indicated amount of money is transferred from the account of the buyer to the account of the seller. Phase 5 is initiated by the credit company if this transaction can be processed and finalized. The credit company sends a confirmation message to the merchant. The message is signed using the private key of the credit company.

$$B \xrightarrow{\text{MSG}=\{H(ID_M|ID_C|TS_C|AM|H(PD|N_C))\}_{SK_B}} M$$

From this message the merchant can check that the payment was made with the agreed amount AM from the account of C to the account of M. The hash value $H(PD|N_C)$ is meant for the customer to make sure that the merchant can not claim that the customer bought something else than the original product.

6. Product delivery. Finally, in phase 6, the merchant checks the message received in phase 5. If the message is valid and the payment has been done, the merchant delivers the product to the customer. The merchant also sends the customer a message stating that the payment has been made and the product has been delivered.

$$M \xrightarrow{\text{MSG}=\{H(ID_M|ID_C|TS_C|AM|H(PD|N_C))\}_{SK_B}} C$$

The customer can check that the amount of money AM, product details PD, the nonce N_C and the timestamp all match the original values to be sure that the correct amount was paid for the correct product.

3.2 Real POS (Vending Machine) Payment

Our protocol for a secure vending machine payment contains the following steps (see Fig. 3):

1. Initiation. The customer C initiates the protocol with the merchant M by choosing a product. In case the vending machine supports several ways of payment, the user may need to explicitly select the mobile payment option. Optionally, the protocol can be initiated by the vending machine, which detects the device when it comes in the range of the Bluetooth communication. No messages are sent in this phase.

2. Bluetooth pairing. To enable exchange of messages, Bluetooth pairing must be performed between the vending machine and the mobile phone. If several Bluetooth devices are in the range, the machine can use a random PIN code for pairing and show this PIN on its display. User must enter this PIN code in the mobile phone.

3. Product offer. If the user has not selected a product yet, the vending machine sends a message with information about available products and their prices. In case phase 1 was initiated by the user, and the product is already selected, the list of products contains only the selected item. In addition to this data, the vending machine sends its own certificate Cert_M and a random nonce N_M.

$$M \xrightarrow{\mathrm{MSG}=\mathrm{Cert}_M|N_M|\texttt{List of products}} C$$

After receiving the message, C extracts M's certificate and checks its validity.

4. Product selection. The user is prompted by the mobile device for selection of a product, unless it has already been selected. The information on the user selection is sent to the vending machine. Also, the customer's certificate Cert_C is included in the message.

The mobile phone must store the price AM of the selected product, as it will be needed later on for payment.

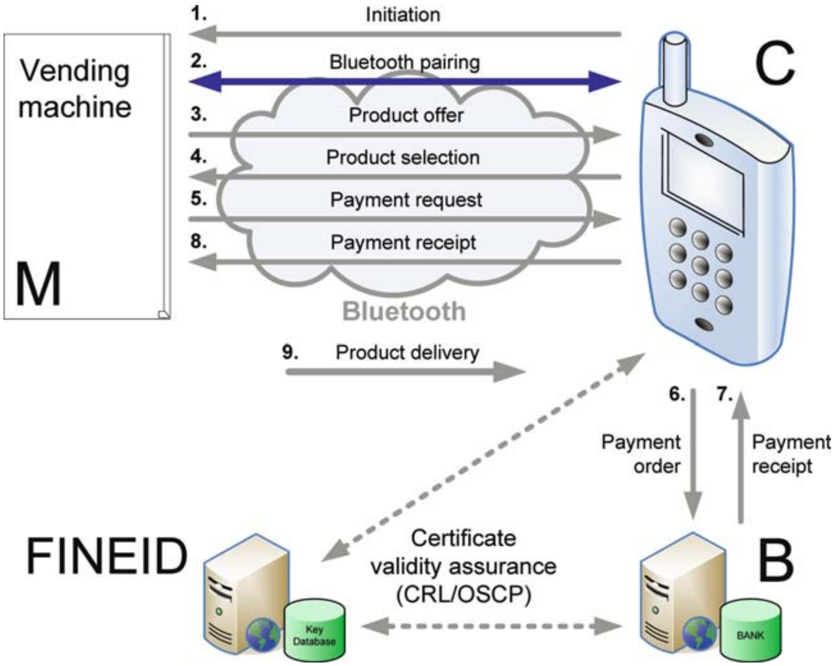

Fig. 3. A Mobile Payment Protocol for Vending Machines

The message in phase 4 consists of three parts. The first part is the user's selection S, and a nonce N_C generated by the mobile device. This part of the message is encrypted with the vending machine's public key PK_M. Second, the user's certificate Cert_C is appended to this message. The last part of the message is a signature $\mathrm{SIG} = \{H(S|N_M|N_C)\}_{SK_C}$.

$$C \xrightarrow[\mathrm{SIG}=\{H(S|N_M|N_C)\}_{SK_C}]{\mathrm{MSG}=\{S|N_C\}_{PK_M}|\mathrm{Cert}_C|\mathrm{SIG}} M$$

After receiving the message, M extracts C's certificate and verifies it. After this M decrypts the message MSG obtaining S and N_C. To conclude, M verifies the signature SIG using the customer's public key. The vending machine could also check the certificate revocation list to see that the user certificate has not been revoked, but this checking can also be made responsibility of the bank.

5. Payment request. The vending machine sends a payment request to the mobile device. The request is signed using the vending machine's private key SK_M.

The payment details include the account number ACN_M, and a reference id of the vending machine ID_M. Note that the price of the product is not sent with the payment details, since C already knows it. However, it is included in a hash in the second part of the message. Namely, C's certificate, price of the product AM, and two nonces N_M and N_C are concatenated, hashed, signed with the vending machine's private key and appended to the message MSG. The last part of the message is a signature $\mathrm{SIG} = \{H(\mathrm{MSG})\}_{SK_M}$.

$$M \xrightarrow[\mathrm{SIG}=\{H(\mathrm{MSG})\}_{SK_M}]{\mathrm{MSG}=ACN_M|ID_M|\{H(\mathrm{Cert}_C|AM|N_M|N_C)\}_{SK_M}|\mathrm{SIG}} C$$

C will later send the signed hash of C's certificate, price and both nonces to the bank B. The bank will use (and optionally store) this as a proof of transaction authorization by vending machine. This way C can not offer one certificate to M and another to B, or change the amount to be paid. The signed hash can also be stored by C as a receipt from the vending machine. Combined with a receipt from the bank (see phase 7), it can be used later as a proof of purchase if a dispute arises.

After receiving the message MSG, C verifies the signature SIG using M's public key.

6. Creation of a payment order. The customer C sends a payment order to the bank B. In addition to the information received from M in the previous steps, an account number of the customer ACN_C and a timestamp TS are needed for the transaction.

C creates a payment order $PO = ACN_C|ACN_M|ID_C|ID_M|AM|N_M|N_C|$ $|TS|\,\{H(\mathrm{Cert}_C|AM|N_M|N_C)\}_{SK_M}$. The payment order is sent to the bank encrypted with the bank's public key and signed with the C's private key.

$$C \xrightarrow[\text{MSG}=\{PO\}_{PK_B}|\text{SIG};\ \text{SIG}=\{H(PO)\}_{SK_C}]{PO=ACN_C|ACN_M|ID_C|ID_M|AM|TS|\{H(\text{Cert}_C|AM|N_M|N_C)\}_{SK_M}} B$$

In this message everything except C's account number ACN_C was received from M in the previous stage. The bank B is obviously the one where the mobile phone user has an account.

Here we assume that the mobile device already has the public key of the bank. Certificates of participating banks can be installed into the device when the software is installed. We can also include a procedure for importing a certificate of a bank which has joined the protocol after the software was installed.

7. Payment processing. After receiving and decrypting the payment order, the bank verifies C's signature attached to it. For this, the bank retrieves C's certificate from the FINEID directory; ID_C is used as the search key. In the same way B gets M's certificate in order to verify the signature $\{H(\text{Cert}_C|N_M|N_C))\}_{SK_M}$. This is done to make sure that the same certificate and nonces were used in communication between C and M. In addition, the bank checks that both certificates are not on the revocation list. The bank also compares the timestamp TS to the stored timestamp of the previous payment order received from C (if any) to defeat replay attacks. Upon successful pass of all checks, the bank transfers the amount of money from C's account to M's account. In case M's account is in another bank, usual interbank procedures are used for crediting money to M. If the transaction can be processed and finalized, the bank sends a confirmation message (receipt) to the mobile phone.

The receipt provides a proof that the payment has been made. The bank account number of the vending machine, amount of money and nonces N_M and N_C are hashed and signed using the bank's private key:

$$B \xrightarrow{\text{MSG}=\{H(ACN_M|AM|N_M|N_C)\}_{SK_B}} C$$

C has all information needed for calculation of the same hash and verification of the bank's signature.

8. Proof of payment. In phase 8, the mobile phone forwards the bank receipt to the vending machine. In order to specify which bank's public key must be used for verification of the receipt, the bank's id ID_B is included in the message.

$$C \xrightarrow{\text{MSG}=ID_B|\{H(ACN_M|AM|N_M|N_C)\}_{SK_B}} M$$

We assume that the vending machine already has certificates of participating banks. Therefore, the vending machine can decrypt the receipt using the bank's public key. The vending machine then calculates hash $H(ACN_M|AM|N_M|N_C)$ and verifies that its value is the same as in the receipt.

The vending machine must have a list of valid public keys of different banks. In case the vending machine does not have a network connection, updating and revoking bank certificates may be cumbersome. The protocol may be extended to check the validity of bank certificates by forwarding Online Certificate Status Protocol (OCSP) requests through the mobile phone to a trusted server (see 2.5 for more details).

4 Security and Privacy

The mobile payment scheme described in this paper satisfies the following security and privacy requirements. In the description, we follow the list of requirements given in [12].

Bank requirements. *Proof of transaction authorization by customer.* Customer signs the payment order that includes id of the vendor, amount of money to be paid, and a timestamp. The signature provides an undeniable proof that the customer has authorized the payment. Signatures are protected against replay attacks by timestamps. Due to their legal acceptance, signatures can be used to resolve possible disputes between the customer and the bank.

Proof of transaction authorization by vendor. Payment requests are signed by the vendor using its private key. Payment requests are not replayable neither by an external adversary nor by the customer due to use of timestamps (in the virtual POS payment) or nonces (in the real POS payment).

Merchant requirements. *Proof of transaction authorization by customer.* The mobile user signs the selection of product or service using their private key. The signature is an unforgeable proof that the customer has authenticated the transaction.

Proof of transaction authorization by bank. If the bank transaction is successfully processed, the bank generates and signes a receipt which is delivered to the merchant. If the merchant does not receive the receipt, or if verification of the signature fails, product is not delivered to the customer. The merchant can store the receipt as a proof of transaction authorization by the bank. Replaying of bank receipts is prevented by the use of timestamps and nonces.

Customer requirements. *Unauthorized payment is impossible.* It is not possible to produce valid signatures, unless one possesses a practically unforgeable token (FINEID card) of the customer and knows the PIN code corresponding to it. The security level is thus comparable to that of ATM cards.

Proof of transaction authorization by bank. In both protocols the customer receives a signed receipt from the bank. In case of a virtual POS payment, the receipt is forwarded by the merchant to the customer. In a real POS payment, the receipt is sent by the bank directly to the customer. Bank receipts are protected against replays by inclusion of timestamps and nonces.

Certification and authentication of merchant. In our MP scheme, the customer receives merchant's certificate directly from the merchant. The customer can check the validity of the certificate by submitting a query to the FINEID directory. Messages that contain product selections are encrypted under the merchant's public key. Nonces are included in these messages, to enable challenge-response authentication of the merchant. Unless the merchant possesses the secret key associated with the public key in the certificate, it cannot proceed with the payment protocol.

Receipt from merchant. In virtual POS payment the merchant forwards a signed bank receipt to the customer. The receipt states that the bank has authorized the payment, which in turn means that the merchant had asked for a payment and thus agreed to deliver the product or service. It must be noted that the merchant can always refuse forwarding the bank receipt to the customer. However, in this case the customer can use the next bank statement as a proof of purchase. In real POS payment, the customer receives two receipts: one from the vending machine as an authorization of transaction, and another one from the bank as a proof of payment.

Privacy. In an ideal payment system, merchants should not learn identities of their customers, and banks should not receive any information about the products that their customers purchase. Clearly, confidentiality of order and payment details should be protected from eavesdropping. Our MP scheme does partially satisfy these requirements. The customer sends product selection details to the merchant in encrypted form, preventing eavesdropping. Messages with payment details are also encrypted. The bank does not receive any information about the purchase except its price, and identities of the customer and the merchant. Note that although in a virtual POS payment the bank receives a hash $H(PD|N_M)$, where PD is a description of the product, a nonce N_M prevents the bank from guessing PD. The merchant learns the identity of their customer, and in a real POS payment also the name of the bank used by the customer. These are the same details as in a credit card payment.

Implementation note. The protocols, clearly, do not guarantee that there are no delays or errors in delivery of messages. There is a number of implementation details to be considered, for example, error handling. They are, however, out of scope of this paper.

5 Conclusions

In this paper we gave a short survey of current mobile payment methods and proposed a novel mobile payment scheme. The scheme can be used in real POS (Point-Of-Sale) as well as in virtual POS. The main advantage of our system is that it does not require any mediator. This reduces the total cost of a payment. We also described two protocols, one for virtual POS and another one for real POS.

The system described in this paper utilizes a governmental PKI infrastructure, namely the FINEID, making it an affordable solution since administration of the system is provided by the government. Furthermore, as citizens have adopted this system for secure electronic transactions, it has a high level of trustworthiness. Our system is built using Java to gain the best possible portability across device platforms.

The proposed mobile payment solution provides strong authentication of communicating parties, integrity of data, non-repudiation of transactions, and confidentiality of communication. Based on governmental PKI, the system is open to all merchants, financial institutions and mobile users.

References

1. GSM Association statistics, Q3 2005. http://www.gsmworld.com
2. Karnouskos, S.: Mobile Payment: A Journey through Existing Procedures and Standardization Initiatives. IEEE Communications Surveys & Tutorials, vol. **6**, no. 4, Oct. 2004.
3. Risks and Threats Analysis and Security Best Practices. Mobile Payment Forum, http://www.mobilepaymentforum.org/pdfs/MPF_Security_Best_Practices.pdf, May 2003.
4. Hassinen, M., Hyppönen, K.: Strong Mobile Authentication. Proceedings of the 2nd International Symposium on Wireless Communication Systems, pp. 96–100, Sept. 2005.
5. Finnish Population Register Centre: FINEID S1 Electronic ID Application. http://www.fineid.fi
6. Bluetooth SIG: Bluetooth specifications 1.0, 1.1, 1.2 and 2.0+EDR. Technical specifications, 1999-2004. https://www.bluetooth.org
7. Sun Microsystems, Inc.: Java 2 Platform, Micro Edition (J2ME). http://java.sun.com/j2me/
8. Java Community Process: JSR-000177 Security and Trust Services API for J2ME. http://jcp.org/aboutJava/communityprocess/final/jsr177/
9. ISO/IEC 7816-4:1995. Integrated circuit(s) cards with contacts. Part 4: Interindustry commands for interchange.
10. Myers M., Ankney R., Malpani A., Galperin S., and Adams C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560, June 1999.
11. Pinkas D. and Housley R.: Delegated Path Validation and Delegated Path Discovery Protocol Requirements. RFC 3379, Sept. 2002.
12. Bellare, M., Garay, J., Hauser, R., Herberg, A., Krawczyk, H., Steiner, M., Tsudik, G., and Waidner, M.: iKP – a family of secure electronic payment protocols. In Proceedings of the 1st USENIX Workshop on Electronic Commerce, July 1995.