Pers Ubiquit Comput (2012) 16:643–654
DOI 10.1007/s00779-011-0436-7

ORIGINAL ARTICLE

Resuscitating privacy-preserving mobile payment with customer in complete control

Divyan Munirathnam Konidala · Made Harta Dwijaksara · Kwangjo Kim · Dongman Lee · Byoungcheon Lee · Daeyoung Kim · Soontae Kim

Received: 30 January 2011 / Accepted: 30 May 2011 / Published online: 7 August 2011

Abstract Credit/debit card payment transactions do not protect the privacy of the customer. Once the card is handed over to the merchant for payment processing, customers are "no longer in control" on how their card details and money are handled. This leads to card fraud, identity theft, and customer profiling. Therefore, for those customers who value their privacy and security of their payment transactions, this paper proposes a choice—an alternate mobile payment model called "Pre-Paid Mobile HTTPS-based Payment model". In our proposed payment model, the customer obtains the merchant's bank account information and then instructs his/her bank to transfer the money to the merchant's bank account. We utilize near field communication (NFC) protocol to obtain the merchant's bank account information into the customer's NFC-enabled smartphone. We also use partially blind signature scheme to hide the customers' identity from the bank. As a result, our payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant. We emulated our proposed mobile payment model using Android SDK 2.1 platform and analyzed its execution time.

Keywords Smartphone application · Secure mobile payment · Privacy · Near field communication (NFC) · RFID · Customer centric payment model

D. M. Konidala (✉)
Department of Information and Communications Engineering,
Korea Advanced Institute of Science and Technology (KAIST),
Daejeon, Republic of Korea
e-mail: divyan@kaist.ac.kr

M. H. Dwijaksara · K. Kim · D. Lee · D. Kim · S. Kim
Department of Computer Science, KAIST, Daejeon,
Republic of Korea
e-mail: made.harta@kaist.ac.kr

K. Kim
e-mail: kkj@kaist.ac.kr

D. Lee
e-mail: dlee@cs.kaist.ac.kr

D. Kim
e-mail: kimd@kaist.ac.kr

S. Kim
e-mail: kims@kaist.ac.kr

B. Lee
Department of Information Security, Joongbu University,
Chungnam, Republic of Korea
e-mail: sultan@joongbu.ac.kr

1 Introduction

Mobile payment is a payment method, where a mobile phone or a smartphone is used to pay for merchandize and services. Mobile payment is gaining popularity especially in Asia and Europe. The research firm Gartner Inc. predicts that the number of mobile payment users will reach more than 190 million in 2012 [9].

In this paper, we focus on two emerging, promising, and related technologies namely: the radio frequency identification (RFID) and near field communication (NFC). The "Mobile NFC Payment" is one of the applications of NFC that is drawing a great deal of attention. Currently, efforts are being put to deploy a mobile NFC payment model that precisely mimics the Contactless (RFID) Card Payment model, where a NFC-enabled smartphone behaves as a contactless credit/debit card. But through this paper, we emphasize that since credit/debit card payment transactions do not protect the customer's privacy and are also prone to card fraud and identity theft, the Mobile NFC Payment application must also have an “alternate” payment model to support those customers who give high priority to privacy and want to be in complete control of their payments.

Previously proposed anonymous (untraceable) electronic cash models were not viable to be deployed as real-world applications. However, in this paper, we take advantage of smartphones, RFID, and NFC technologies to resuscitate anonymous (untraceable) electronic cash model and propose a simple, efficient, and privacy-preserving mobile payment model, which could be an alternative to the credit/debit card-based Mobile NFC Payment.

1.1 Radio frequency identification (RFID)

RFID [24] technology offers businesses an automated supply chain management system. Inexpensive Passive-RFID tags can be attached to individual product items, cases, pallets, etc. A tag is powered up by an RF signal generated from a RFID reader. The tag’s tiny computer chip contains an electronic product code (EPC) number [7, 8] that uniquely identifies the product to which it is attached to, and its antenna transmit this EPC to readers within the RF range (up to 10 m, without needing line-of-sight scanning as in the case of bar-codes).

Further information associated with a particular EPC is captured and stored on a network of servers and databases, called the EPC-Information Services (EPC-IS) [8]. Therefore, RFID and EPC-IS assist geographically distributed supply chain stakeholders with instantaneous product identification, and “real-time” updating, querying, accessing and sharing of product information such as, shipping and receiving, track and trace, product theft detection, anti-counterfeiting measures, precise product recall [25], etc. As a result, in the near future, we can expect to see tagged items at many retailers.

1.2 Near field communication (NFC)

NFC [14] is a short-range high-frequency wireless connectivity standard (ISO/IEC 18092), which enables the exchange of data between devices when they are touched or waved within 4 cm of each other. NFC is a combination of the already existing proximity-card standard (ISO/IEC 14443, contactless RFID card) and a reader into a single chip, operating at 13.56 MHz and transferring data at up to 424 Kbits/s.

NFC technology brings RFID closer to common people, where a “reader-tag” chip is embedded into data communicating handsets like mobile/smartphones, and personal digital assistants and media players. As a result, a smartphone also becomes a RFID reader and a RFID tag, allowing users to “scan, download, and view detailed information represented by tags (attached to items), and “identify themselves and communicate with other readers”. We also have the NFC Forum [20] to advance the use of NFC by developing specifications, and ensuring interoperability among NFC-enabled consumer electronics, mobile devices, PCs, and services.

2 Motivation and related work

2.1 Drawbacks of credit/debit card payments

In recent years, the number of credit/debit card payment transactions have substantially increased. These cards offer great convenience to customers, eliminating the need to carry cash (banknotes and coins) for most of the payments, and the payments are accomplished much faster. The credit cards also allow customers to obtain instant loans (based on their credit limit), which they can repay at a later time. However, it is well known that the credit/debit card payment transactions have the following critical drawbacks.

2.1.1 Privacy violation and card fraud

Credit/debit card payment transactions do not protect the customer’s privacy. The customer’s card details, the payment amount, and when and where the payment was made are of course exposed to the merchant but also to the merchant’s bank (acquirer), the card companies (e.g., Visa, MasterCard), the customer’s bank (card issuer), and multiple intermediate third-party payment processor companies (Independent Sales Organizations), and internet payment gateway companies. Exposing the customer’s card details to so many entities leads to serious card frauds, skimming, identity theft, and customer profiling [21].

Many times the systems at the intermediate third-party payment processor/gateway companies are breached and a huge number of credit/debit card details are exposed [22]. In the year 2009, the payment processor “Heartland Payment Systems Inc.” disclosed that it became a victim of a massive data breach [26], details of more than 130 million credit/debit cards were believed to be stolen making it the biggest payment card breach to date. Similarly, some of the stores that belong to 7–11, Hannaford Brothers, and TJX Companies Inc. have also been breached by hackers.

2.1.2 Customer not in control of payment

Most of the customers are not confident, instead extremely cautious, while using their credit/debit cards, because once they handover the card to the cashier, they are “no longer in control” on how their card details and money are handled, the card security is potentially compromised from here on. Since customers use their cards extensively, they always fear becoming victims of a card fraud and/or an identity theft. Customers can identify irregularities only after the fraud has been committed, and that too by thoroughly verifying their monthly credit/debit card statements.

2.1.3 Temptation to overspend and bankruptcy

Especially with credit cards, customers are quickly tempted to spend more than they can afford. The credit card debt with high interest rate can lead to financial crisis and bankruptcy [15].

2.1.4 Expensive "per transaction" fees for retailers and inflated pricing for customers

The card companies charge an expensive per transaction "interchange fee" [10] from the merchants and the intermediate third-party payment processor/gateway companies also charge several other fees. To compensate for these transaction fees, merchants may charge customers extra for card payments or inflate the prices of their merchandize, effecting even those customers who do not use credit/debit cards for payments [23]. On the other hand, merchants actually prefer cash payments in order to reduce their "per transaction" credit/debit card processing fees.

2.2 Drawbacks of contactless (RFID) credit/debit card payments and mobile NFC payment

The card companies, MasterCard, and Visa have introduced contactless (RFID) credit/debit cards, PayPass [16], and payWave [27], respectively. These cards are based on the standard for radio frequency (RF) cards—ISO 14443 type:A/B [13]. MasterCard and Visa have also integrated their contactless credit/debit card payment model into NFC-enabled mobile phone as a mobile NFC payment model [17, 28] and are now conducting trails.

Customers instead of handing their card to a cashier bring their contactless card or NFC-enabled mobile phone within one-two inches of a reader at point-of-sale. The stored card details in the mobile phone are sent to the reader, and the rest of the payment transaction procedure is the same as the normal credit/debit card payments. Therefore, even though the contactless card or NFC-enabled mobile phone is in the possession of the customer, the above-mentioned credit/debit card payment drawbacks are not alleviated. For the card companies, contactless RIFD cards or mobile NFC payments are intended for getting more faster credit transactions from customers.

On the other hand, these contactless cards can be scanned from a distance without the knowledge of the customer. Heydt-Benjamin et al. [11] have shown various vulnerabilities in several contactless RFID credit cards. (Verbatim from [11]). Their study observed that the cardholder's name and often credit card number and expiration are leaked in plaintext to unauthenticated readers, some cards may be skimmed once and replayed at will, and they are susceptible in various degrees to a range of other traditional RFID attacks such as skimming and relaying.

2.3 Drawbacks of prepaid contactless card payment

We use prepaid contactless cards for micro-payments at the subway, bus, vending machines, etc., but such cards do have a unique ID and the customer can be tracked and traced based on this unique ID. The customer can protect his/her privacy by frequently canceling these cards and re-issuing new ones. But canceling and re-issuing these cards require the customer to personally visit the authorized entity and it often involves non-refundable registration fees.

2.4 Drawbacks of anonymous electronic cash payment

Anonymous Electronic Cash Payment models [3, 5, 6] are based on (partially) blind signature schemes and they adopt coin/token-based approach. These payment models are a bit complicated, where the customer has to withdraw a big coin from the bank, divide the big coin to smaller coins, pay the merchant with the smaller coins, the merchant submits the coins to the bank, the bank verifies the coins for illegal double spending, and if it detects double spending, it should trace the customer who committed the fraud. The bank also have to verify if the merchant has not re-submitted already cleared coins, as a result, the bank has a huge burden. These payment models have to be implemented and deployed precisely to prevent fraud; therefore, they have not been viable for large-scale deployment.

3 Goals: offering a choice—alternate payment model

We do not want to replace credit/debit card payments. There will be services/merchandize sold solely based on these cards, and customers and merchants who love the convenience and benefits got from these cards. But due to the above-mentioned drawbacks, we also believe that there is a large number of customers and merchants very reluctantly using and accepting credit/debit card payments as they do not have a choice, they can of course deal in cash payments, but it is so inconvenient and unsafe to carry cash at all times. Therefore, in this paper, we offer both customers and merchants another choice to choose from—an alternate payment model that satisfies the following goals:

- Neither a “credit/debit card” nor a “contactless card” payment model
- Simple (is the best), efficient, faster, convenient, and secure payment model
- Protect customer privacy from the banks and merchants
- Provide customers the complete control on their payment transactions
- Prevent customers from waiting in long check-out lines
- Reduce transaction fees for merchants and provide instant payment

4 The big picture of the proposed mobile payment model

We propose a “Pre-Paid Mobile HTTPS-based Payment model”. For ease of describing our idea, let us consider one particular customer called Alice. Let us also assume that: both Alice and the merchant have a bank account in the same bank, the items (in the store) chosen by Alice are all tagged with RFID tags, and Alice’s smartphone is NFC-enabled. We consider 3 entities: (1) Customer Alice with NFC-enabled smartphone (NSP), (2) Bank, and (3) Store. Our proposed payment model involves four procedures as described below:

4.1 Anonymous pre-paid digital cash certificate issuing procedure

Alice can use her computer or her smartphone’s 3G/4G network to establish a secure HTTPS (Hypertext Transfer Protocol Secure) connection [12] with the bank and request for a digital cash certificate of a certain amount. The bank deducts the amount from Alice’s account and returns a digitally signed cash certificate. This pre-paid cash certificate is anonymous, i.e., it does not contain any details of Alice’s true identity, and the bank itself cannot link this certificate to Alice at a later stage. Alice stores this *Anonymous Pre-Paid Digital Cash Certificate* in her smartphone.

4.2 Obtaining digital invoice certificate procedure

Alice visits a department store and chooses some items. She approaches one of the several RFID-kiosks in the store. The kiosk instantaneously scans the RFID-tagged items in the shopping cart and generates a digital *Invoice Certificate*. This *invoice certificate* can contain several details such as store’s name and address, list of items chosen, and their prices, etc., but most importantly, it contains: a unique *invoice ID*, the *invoice amount*, and the *merchant’s bank account details* (account name and account number). Alice must deposit the *invoice amount* into the merchant’s *bank account* in order to complete the payment. Alice’s NFC-enabled smartphone, which is brought closer to the kiosk’s NFC module, can download this *invoice certificate*.

Alice can now walk away from the kiosk, but cannot leave the store, since her chosen items have not been flagged as “sold” in the database and they would trigger the alarm at the store’s exit. However, Alice can complete the remaining payment procedure at her own comfort anywhere within the store, e.g., least crowded area or while sitting at the store’s food court, without having to wait in long check-out lines.

4.3 The payment procedure

Alice uses her smartphone’s 3G/4G network to establish a secure HTTPS connection with the bank and submits a digital *Cheque Certificate*. This *Cheque* contains: Alice’s *Anonymous Pre-Paid Digital Cash Certificate*, the *invoice ID*, the *invoice amount*, and the *merchant’s bank account details* (account name and account number). As mentioned above, the bank cannot link the *Anonymous Pre-Paid Digital Cash Certificate* to Alice, but to prevent unauthorized use of this cash certificate, the *Cheque Certificate* also provides an “anonymous proof” to the bank that the owner of the cash certificate is indeed involved in this payment procedure.

If this is the first time that the bank is receiving this *Anonymous Pre-Paid Digital Cash Certificate*, it registers this cash certificate into its database, along with a *certificate balance* parameter. Initially, the *certificate balance* value is equal to the amount value on the cash certificate. The bank deducts the *invoice amount* from the *certificate balance* value and deposits the *invoice amount* into the merchant’s *bank account*. The bank then updates the deducted *certificate balance* value in its database. From here on, whenever the same *Anonymous Pre-Paid Digital Cash Certificate* is received for other payments, the bank first checks the *certificate balance* in its database and then proceeds with the deposit, else it would respond “insufficient cash”.

The bank sends a digital *Invoice Paid Receipt* to the merchant. This receipt confirms that the *invoice ID* has been paid. The merchant flags the items listed under the *invoice ID* as “sold” in the database and also acknowledges this to the bank. The bank can now send the same *Invoice Paid Receipt* to Alice, confirming that the *invoice ID* payment has been successful. Alice can now leave the store with her purchased items.

4.4 Reclaiming unspent amount procedure

Alice can choose to cancel her *Anonymous Pre-Paid Digital Cash Certificate* whenever she wants or when the smartphone alerts Alice, that her cash certificate is soon expiring and it has some unspent balance amount. Alice has two options to cancel her certificate and reclaim the unspent amount.

1. Alice connects her smartphone to the bank's ATM and "anonymously proves" that she is indeed the owner of the cash certificate. The bank refunds the unspent amount via the ATM's cash dispenser and cancels the cash certificate.
2. Let us assume that Alice obtains another new cash certificate. Alice can use her computer or her smartphone's 3G/4G network to establish a secure HTTPS connection with the bank and "anonymously proves" that she is indeed the owner of both the cash certificate that is to be canceled and the new cash certificate. The bank updates the database by adding the unspent amount value on the to-be-canceled cash certificate to the *certificate balance* value of the new cash certificate and cancels the to-be-canceled cash certificate.

4.5 Benefits/economic motives for the entities

Customer Alice who is in complete control knows the merchant's bank account and the invoice amount. Since the cash certificate is a digital form of physical cash, she keeps a check on her payments and resists overspending. Unlike credit cards, there are no interest fees in this model. Alice can use her smartphone to easily cancel and request new cash certificates anywhere at anytime. Alice's privacy is protected from both the merchant and the bank.

Our proposed payment model can be provided by any bank, unlike the credit/debit cards that are monopolized by few card companies. Merchants pay a small fee to the bank when compared to the expensive fees paid to multiple parties associated with card transactions; thus avoid inflating commodity prices. Customers need not wait in checkout lines and the mobile operator charges customers for using 3G/4G Internet.

4.6 Requirements

Partially blind signature [1] scheme is needed for customer privacy protection: *Anonymous Pre-Paid Digital Cash Certificate* and for the customer to "anonymously prove" the ownership of the *Anonymous Pre-Paid Digital Cash Certificate*. We assume that the NFC-enabled smartphone and the bank are capable of executing partially blind signature procedure.

In 1982, Chaum [6] invented a new cryptographic primitive called blind signature as a primer tool to design electronic payment and electronic voting schemes with user privacy protection in mind. The blind signature is a special kind of digital signature, which allows users to get signatures on their messages from authorized entities/signature issuers (e.g. banks, trusted third parties) without revealing the message contents to the authorized entity. The detailed description of a blind signature scheme based on the RSA digital signature scheme is given in [6].

Blind signatures provide total privacy for users by fully hiding messages (to be signed) from the signer. However, this property is not desired from the signer's point of view because he is responsible for his signatures and he needs to know what he would be signing on. To achieve a compromising solution for both the signer and users, Abe and Okamoto proposed the idea of partially blind signature [1]. A partially blind signature scheme is an extension of an ordinary blind signature scheme. It has two portions, one portion consists of the message that is hidden by the user (as in blind signature scheme) and in the other portion, the signer can explicitly embed necessary information such as issuing date, expiry date, signer's identity etc. The signer and users should of course agree on information being embedded into signatures. Also in [1], Abe and Okamoto presented a concrete construction of a partially blind signature scheme and proved its security. In this paper, we implemented the randomized RSA-based partially blind signature scheme proposed by Cao et al. [4]. The following section describes how the above cryptographic primitive is used in designing our proposed payment model.

The Digital Signature Algorithm (DSA) [19] is needed for entity authentication and data integrity.

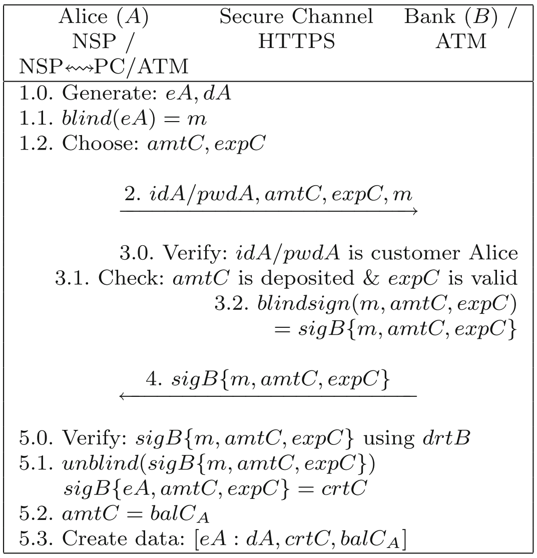
The HTTPS (Hypertext Transfer Protocol Secure) [12] connection is needed for bank (server) authentication and mobile phone (client)-bank (server) data confidentiality.

5 Technical details of the proposed mobile payment model

We consider 3 entities: (1) Customer Alice with NFC-enabled smartphone (NSP), (2) Bank, and (3) Store. Table 1 provides the list of notations we used in this paper. The communication channel between the entities is secured via the standard HTTPS protocol. The procedures described in the Subsects. 4.1, 4.3, and 4.4 are well depicted in the Figs. 1, 2, 3, and 4. Due to the space constraint, we describe below the important steps of these phases, skipping some of the trivial ones.

Table 1 Notations

Notation	Description
A	Customer Alice's NFC smart phone (NSP)
B	Bank
S	Store/service provider (S)
X	An entity: *A*, *B*, *S*
idX	Identity of X
eX	Public key of X
dX	Secret-key of X
drtX	Digital certificate of X
sigX{ }	Digital signature using *dX*
expC	Certificate expiry date
amtC	Amount value of cash certificate
crtC	Anonymous pre-paid digital cash certificate
balC	Available balance amount on *crtC*
date, *time*	Date and time of generating certificate
acctS	Bank account details of S
amtV	Total invoice amount
idV	Unique ID of the invoice
crtV	Digital invoice certificate
crtQ	Digital cheque certificate
crtR	Digital paid receipt certificate



Alice (A) NSP / NSP⇜⇝PC/ATM — Secure Channel HTTPS — Bank (B) / ATM

1.0. Generate: eA, dA

1.1. $blind(eA) = m$

1.2. Choose: $amtC, expC$

2. $idA/pwdA, amtC, expC, m$ ⟶

3.0. Verify: $idA/pwdA$ is customer Alice

3.1. Check: $amtC$ is deposited & $expC$ is valid

3.2. $blindsign(m, amtC, expC)$
$= sigB\{m, amtC, expC\}$

⟵ 4. $sigB\{m, amtC, expC\}$

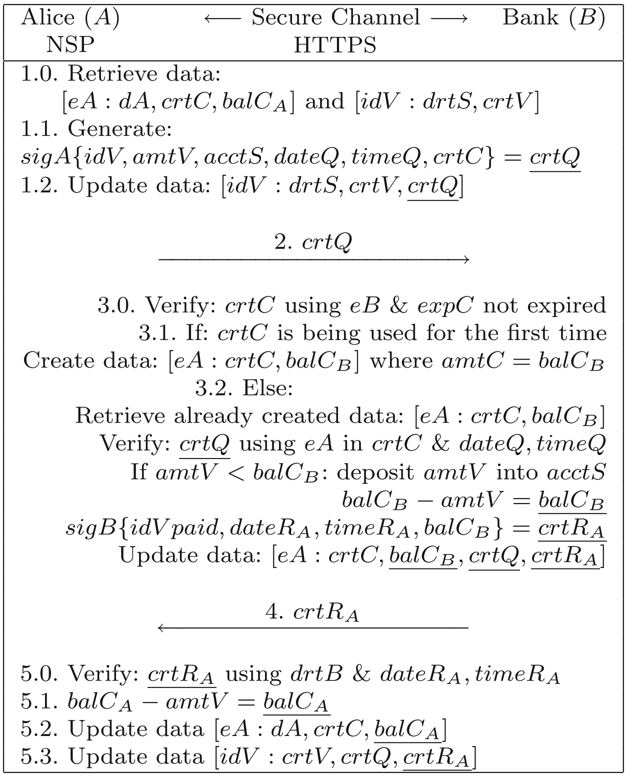
5.0. Verify: $sigB\{m, amtC, expC\}$ using $drtB$

5.1. $unblind(sigB\{m, amtC, expC\})$
$sigB\{eA, amtC, expC\} = crtC$

5.2. $amtC = balC_A$

5.3. Create data: $[eA : dA, crtC, balC_A]$

Fig. 1 Anonymous pre-paid digital cash certificate *crtC* issuing procedure



Alice (A) NSP ⟵ Secure Channel HTTPS ⟶ Bank (B)

1.0. Retrieve data:
$[eA : dA, crtC, balC_A]$ and $[idV : drtS, crtV]$

1.1. Generate:
$sigA\{idV, amtV, acctS, dateQ, timeQ, crtC\} = \underline{crtQ}$

1.2. Update data: $[idV : drtS, crtV, \underline{crtQ}]$

2. $crtQ$ ⟶

3.0. Verify: $crtC$ using eB & $expC$ not expired

3.1. If: $crtC$ is being used for the first time
Create data: $[eA : crtC, balC_B]$ where $amtC = balC_B$

3.2. Else:
Retrieve already created data: $[eA : crtC, balC_B]$
Verify: $\underline{crtQ}$ using eA in $crtC$ & $dateQ, timeQ$
If $amtV < balC_B$: deposit $amtV$ into $acctS$
$balC_B - amtV = \underline{balC_B}$
$sigB\{idVpaid, dateR_A, timeR_A, balC_B\} = \underline{crtR_A}$
Update data: $[eA : crtC, \underline{balC_B}, \underline{crtQ}, \underline{crtR_A}]$

⟵ 4. $crtR_A$

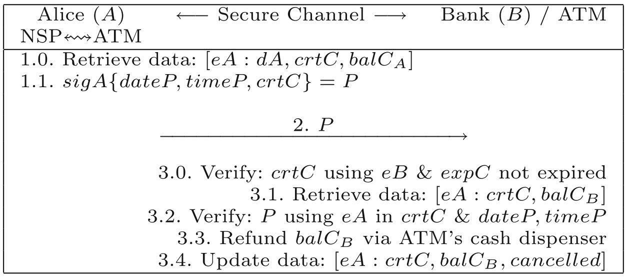
5.0. Verify: $\underline{crtR_A}$ using $drtB$ & $dateR_A, timeR_A$

5.1. $balC_A - amtV = \underline{balC_A}$

5.2. Update data $[eA : dA, crtC, \underline{balC_A}]$

5.3. Update data $[idV : crtV, crtQ, \underline{crtR_A}]$

Fig. 2 The payment procedure



Alice (A) NSP⇜⇝ATM ⟵ Secure Channel ⟶ Bank (B) / ATM

1.0. Retrieve data: $[eA : dA, crtC, balC_A]$

1.1. $sigA\{dateP, timeP, crtC\} = P$

2. P ⟶

3.0. Verify: $crtC$ using eB & $expC$ not expired

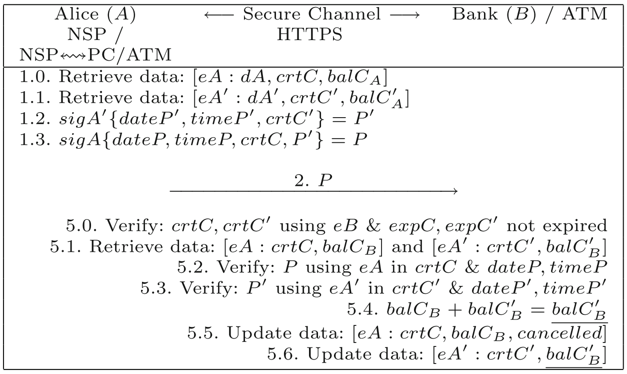
3.1. Retrieve data: $[eA : crtC, balC_B]$

3.2. Verify: P using eA in $crtC$ & $dateP, timeP$

3.3. Refund $balC_B$ via ATM's cash dispenser

3.4. Update data: $[eA : crtC, balC_B, cancelled]$

Fig. 3 Reclaiming unspent amount procedure: option 1



Alice (A) NSP / NSP⇜⇝PC/ATM ⟵ Secure Channel HTTPS ⟶ Bank (B) / ATM

1.0. Retrieve data: $[eA : dA, crtC, balC_A]$

1.1. Retrieve data: $[eA' : dA', crtC', balC'_A]$

1.2. $sigA'\{dateP', timeP', crtC'\} = P'$

1.3. $sigA\{dateP, timeP, crtC, P'\} = P$

2. P ⟶

5.0. Verify: $crtC, crtC'$ using eB & $expC, expC'$ not expired

5.1. Retrieve data: $[eA : crtC, balC_B]$ and $[eA' : crtC', balC'_B]$

5.2. Verify: P using eA in $crtC$ & $dateP, timeP$

5.3. Verify: P' using eA' in $crtC'$ & $dateP', timeP'$

5.4. $balC_B + balC'_B = \underline{balC'_B}$

5.5. Update data: $[eA : crtC, balC_B, cancelled]$

5.6. Update data: $[eA' : crtC', \underline{balC'_B}]$

Fig. 4 Reclaiming (unspent money) phase: option 2

5.1 Anonymous pre-paid digital cash certificate issuing procedure

Fig. 1: 1.0–3.2: The *eA* is just a public key value and not a certificate-authority issued digital certificate (containing a public key and also its owner's identity) and it is blinded as *m*. Alice uses her NSP's 3G/4G network to connect to her bank's Internet banking facility and authenticates herself

with ID/Pwd. Alice types in her desired *amtC* value of the certificate and its expiry duration (*expC*), e.g., number of days. The bank deducts the *amtC* from Alice's account. It verifies if the *expC* value satisfies the bank rules. By utilizing the partially blind signature scheme, the bank embeds *amtC* and *expC* values while signing *m*, to generate: $sigB\{m, amtC, expC\}$.

Fig. 1: **5.1–5.3:** The NSP un-blinds *m* to reveal the *eA* in the *Anonymous Pre-Paid Digital Cash Certificate*: $crtC = sigB\{eA, amtC, expC\}$. The $balC_A$ indicates the up-to-date balance amount on the *crtC* after every payment transaction; therefore, it is initially assigned the value of *amtC*. The *eA* becomes the pseudo-ID of the *crtC*; therefore, the NSP creates a data table in its data storage space with *eA* being the primary reference field; [eA:dA, $crtC$, $balC_A$].

Other options: Anonymous Pre-Paid Digital Cash Certificate can be obtained using the smartphone, but using an ATM or PC will reduce communication and computational burden on the phone. Alice can plug the NSP to her PC and let the PC execute this procedure and transfer *eA*, *dA*, *crtC*, and *drtB* to the NSP. Alice can also connect her NSP to the bank's ATM. The NSP sends *m* and Alice enters *amtC* and *expC* into the ATM, which then returns $sigB\{m, amtC, expC\}$ and *drtB* to the NSP. If Alice does not have a bank account, she can still connect her NSP to the ATM, but Alice must deposit the cash of *amtC* into the ATM.

5.2 Obtaining digital invoice certificate procedure

Alice chooses some tagged items at a department store and approaches one of the several RFID-kiosks in the store. The kiosk instantaneously scans the tagged items in the shopping cart and generates a digital *invoice certificate* $crtV = sigS\{idV, amtV, acctS\}$. Alice must deposit *amtV* into the *acctS* to complete the payment. The *crtV* can contain other details such as store's name and address, list of items chosen and their prices, etc. The NSP is brought closer to the kiosk's NFC module and can thus download the *crtV* and the store's *drtS*, which is used to verify the signature on *crtV*. The NSP adds a data record in its data storage space as [*idV*:*drtS*, *crtV*].

Alice can now walk away from the kiosk but cannot leave the store, since her chosen items have not been flagged as "sold" and they would trigger the alarm at the store's exit. However, Alice can complete the remaining payment procedure at her own comfort anywhere within the store, e.g., least crowded area or while sitting at the store's food court, without having to wait in long check-out lines.

5.3 The payment procedure

Fig. 2: **1.0–2:** To complete the remaining payment procedure, the NSP generates a digital *cheque* (*crtQ*), authorizing the bank to deduct the *amtV* from the *crtC* and deposit the *amtV* into the *acctS*. The NSP connects to the bank via the 3G/4G network, sending "only" the *crtQ*.

Fig. 2: **3.0–3.2:** Since the bank issued the *crtC*, it verifies its signature on the *crtC*. If the *crtC* is being used for the first time, then the bank adds a database record with *eA* as the primary reference and the $balC_B$ initially assigned the value of *amtC* specified in the *crtC*. From here on, whenever the *crtC* is received for other payments, the bank first checks and then updates this data [eA:$crtC$, $balC_B$].

To protect privacy, this procedure "does not" require Alice to authenticate herself to the bank and also the *eA* was blinded from the bank during the *crtC* issuing procedure; therefore, the bank cannot link this *crtC* to Alice, but to prevent unauthorized use of the *crtC*, the bank needs an anonymous proof that the owner of the *crtC* is indeed involved in this payment procedure. Therefore, the *eA* included in the *crtC* must verify the signature on the *crtQ*, proving to the bank that a owner possessing the *dA* has signed the *crtQ*. Thus, the *crtQ* anonymously proves the ownership of the cash certificate.

If the $amtV > balC_B$, the bank responds "insufficient balance on the *crtC*" and ends the payment procedure. Its not shown in the Fig. 2, but the bank sends a digital *paid receipt*: $crtR_S = sigB\{idVpaid, amtV, acctS, dateR_S, timeR_S\}$ to the store confirming the payment for *idV*. The store flags the items listed under *idV* as "sold" and sends an acknowledgment to the bank. Now the bank sends a digital *paid receipt* $crtR_A$ to the NSP, proving the successful completion of the payment. The bank updates its database with the new $balC_B$ value and also adds the *crtQ* and $crtR_A$ as a proof of this payment.

Fig. 2: **5.1–5.3:**The NSP calculates $balC_A - amtV = balC_A$ and updates its data with the new $balC_A$ value and also adds the $crtR_A$ as the proof of this payment. Alice can now leave the store with her purchased items.

5.4 Reclaiming unspent amount procedure

The NSP alerts Alice that her *crtC* is soon expiring, and it has some unspent balance amount. Alice has two options to reclaim this amount. Again, to protect privacy, this procedure "does not" require Alice to authenticate herself to the bank.

Option 1 (Fig. 3): Alice connects her NSP to the bank's ATM and like in the payment procedure, the NSP anonymously proves that she is the owner of the *crtC*. The bank refunds the unspent amount via the ATM's cash dispenser and cancels *crtC*.

Option 2 (Fig. 4): Alice can obtain another cash certificate $crtC'$. Later, Alice can either use her NSP,

NSP$\leftrightsquigarrow$PC to connect to the bank, or NSP$\leftrightsquigarrow$ATM and anonymously prove that she is the owner of both the *crtC* and *crtC′*. The bank/ATM then adds the unspent amount value on *crtC* to the balance amount value of the new *crtC′* and cancels *crtC*.

6 Analysis

Our proposed mobile payment model utilizes Partially Blind Signature Scheme for customer privacy: Anonymous Pre-Paid Digital Cash Certificate and and for the customer to "anonymously prove" the ownership of the *Anonymous Pre-Paid Digital Cash Certificate*. Digital Signature Algorithm for entity authentication and data integrity. HTTPS (Hypertext Transfer Protocol Secure) communication for entity authentication and data confidentiality. Therefore, our solution can easily adhere to and deployable (as smartphone application) based on the secure "Electronic Data Interchange (EDI) via the Internet" [18] model. Currently the Internet payment transactions are carried via HTTPS and EDI model.

6.1 Customer privacy protection

In our proposed scheme, the customer is in charge of the payment scheme and the bank pays the store; therefore, the customer remains anonymous to the store at all times. Our scheme provides restricted customer privacy when dealing with the bank. The bank receives customer's public-key value *eA*; therefore, the customer can still be tracked with his/her *eA* usage, until the customer's cash certificate *crtC* is either expired or canceled. But the real identity of the customer is never revealed, because *eA* acts as a pseudonym for the customer. Also, there is no match between the real ID of the customer and his/her *eA*, because during the cash certificate issuing phase the *eA* is blinded, and the bank knows the value of *eA* only during the payment phase. Our proposed model allows customer to easily cancel and request new cash certificates using smartphone periodically, further protecting his/her privacy.

On the other hand, whenever a smartphones accesses Internet via the 3G/4G network, it is assigned a different IP address each time [2]. This property further protects the customer privacy when establishing a HTTPS communication with the bank.

6.2 Customers' public key collisions

In our proposed payment model, we allow the customers to generate their own public key pair (eA; dA); therefore, the question of another customer having the same public key may arise, thus causing a public key collision. Such a rare scenario could be prevented by the bank strictly accepting only well-proven public-key algorithms like the RSA algorithm with a larger key size e.g., 1,024 bits. Public key is always derived from a private key and not vice-versa, and the private key of a particular customer is never exposed in our model; therefore, an adversary cannot generate the same public key as Alice. Lastly, during every transaction between the customer and the bank, the customer has to make use of his private key to anonymously prove that the public key in the pre-paid cash certificate is indeed generated from the private key he/she possesses.

6.3 Man-in-the-middle attack

Our proposed mobile payment model utilizes HTTPS as in the case of EDI model; therefore, communication channel is well secured from man-in-the-middle attacks. Even if the phone's network is lost, but once the network's re-established, customer will use the unique invoice ID and cash certificate to request the payment status or the paid receipt from the bank. Since we use digital signatures and HTTPS communication, all transactions are atomic and can be easily verified.

6.4 Prevent replay attack

The current date and time are included in every signature to detect re-played messages.

6.5 Prevent double spending

In our payment model the bank keeps track of *eA* and it's corresponding $balC_B$ value to prevent any double spending.

6.6 Non-repudiation

In our payment architecture, non-repudiation is satisfied because both the customer and the merchant trust the bank. The digital receipts generated by the bank prove that the transaction between the customer and merchant has been successful. Also the digital cheque and invoice are proofs of the transaction.

6.7 Little overhead

Our scheme poses little overhead both on the NSP and the bank. It can be noticed that the NSP has to just pass on the *crtQ*. The bank takes over the task of paying the merchant. The bank needs to keep track of just the *eA* and update it's corresponding $balC_B$ value until the cash certificate is expired. The cash certificate's expiry date prevents bank from keeping track of cash certificates for an infinitely long time. Expired certificate cannot be used for payments but can be submitted (within a grace period) for reclaiming unspent money. The merchant receives the payment immediately. The bank must also keep track of canceled cash certificates until their expiry, to prevent an adversary from resubmitting canceled cash certificates.

We assumed that the customer and the merchant have bank accounts at the same bank. But this assumption is purely for clarity and ease of explaining our model. Merchants can have accounts in several banks, and the RFID-kiosk can offer the customers a list of banks to choose from, so that the customer can pick a particular bank that has issued his/her Anonymous Pre-Paid Digital Cash Certificate. As a result the RFID-kiosk can generate the invoice certificate containing the merchant's account details at the bank chosen by the customer.

Limitation: On the other hand, if the merchant has only one account at a bank that is different from the customer's bank, then the customer's bank has to communicate with the merchant's bank and wire transfer the invoice amount. This would delay our payment model's processing time as the customer's bank have to await the confirmation from the merchant's bank.

6.8 Against stolen smartphone and money laundering

Our proposed model can be implemented as a smartphone application that is password protected (including the keys); this prevents un-authorized usage if the smartphone is stolen. One of the other possible countermeasures is storing the critical information in secure hardware, and signing is done inside the secure hardware. The customer can keep a copy of the cash certificates in his/her computer, and in case the phone is lost or broken, he /she can cancel the lost certificates and reclaim the money using the stored cash certificates in the computer.

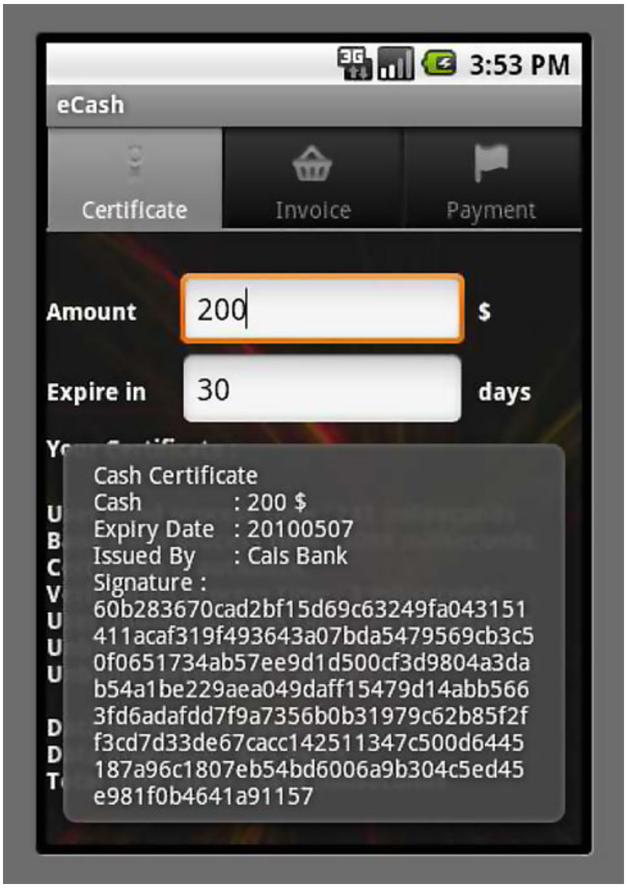
Our model also keeps a check on money laundering; the bank is always involved in transactions. Though the bank cannot identify the customer, it knows the payee's (store) identity and the amount of money being deposited.

7 Proof-of-concept implementation

7.1 Environment

Client: customer alice

- Android SDK 2.1 platform using the emulator: Android Eclair—SDK 2.1



Amount 200 $

Expire in 30 days

Cash Certificate
Cash : 200 $
Expiry Date : 20100507
Issued By : Cals Bank
Signature :
60b283670cad2bf15d69c63249fa043151
411acaf319f493643a07bda5479569cb3c5
0f0651734ab57ee9d1d500cf3d9804a3da
b54a1be229aea049daff15479d14abb566
3fd6adafdd7f9a7356b0b31979c62b85f2f
f3cd7d33de67cacc142511347c500d6445
187a96c1807eb54bd6006a9b304c5ed45
e981f0b4641a91157

Fig. 5 Requesting and obtaining cash certificate (*crtC*) from bank

- Java language on the eclipse IDE environment with java.security package built in on JDK 1.6.
- Ksoap2 for SOAP to contact with web server
- Database Management: SQLite

Server: bank, store

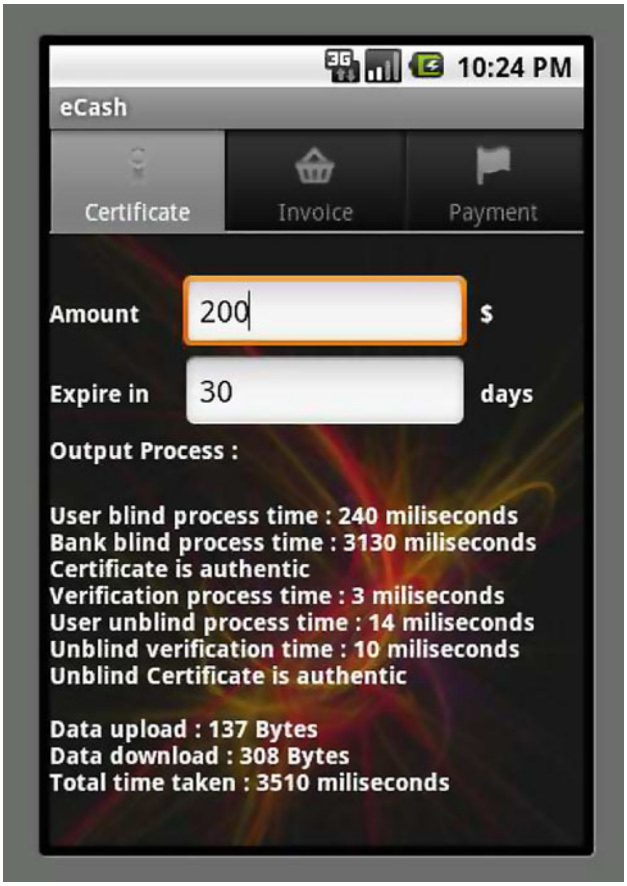
- PHP-based web service (PHP/5.2.9)
- Nu-Soap for building web service and SOAP communication
- Server Specification: Processor : Intel Core 2 Duo E6750 @2.66GHz, Memory : 2GB, Microsoft Windows XP Pro, Web server : Apache HTTPD 2.2.11
- Database Management: MySQL 5.1.33

Secure communication b/w client and server

- HTTPS: OPENSSL for SSL connection (OpenSSL/0.9.8i)

Three procedures

- Cash Certificate (crtC) issuing procedure (Figs. 5, 6)
- Obtaining Invoice Certificate (crtV) procedure (Fig. 7)
- Payment procedure (Fig. 8)



Amount 200 $

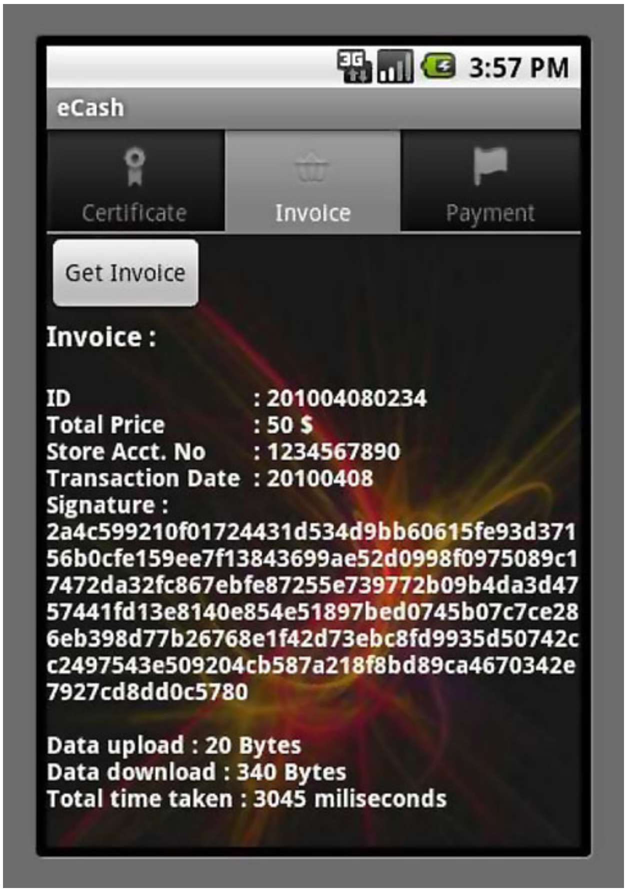
Expire in 30 days

Output Process :

```
User blind process time : 240 miliseconds
Bank blind process time : 3130 miliseconds
Certificate is authentic
Verification process time : 3 miliseconds
User unblind process time : 14 miliseconds
Unblind verification time : 10 miliseconds
Unblind Certificate is authentic

Data upload : 137 Bytes
Data download : 308 Bytes
Total time taken : 3510 miliseconds
```

Fig. 6 Execution time—cash certificate (*crtC*)



Invoice :

```
ID                : 201004080234
Total Price       : 50 $
Store Acct. No    : 1234567890
Transaction Date  : 20100408
Signature :
2a4c599210f01724431d534d9bb60615fe93d371
56b0cfe159ee7f13843699ae52d0998f0975089c1
7472da32fc867ebfe87255e739772b09b4da3d47
57441fd13e8140e854e51897bed0745b07c7ce28
6eb398d77b26768e1f42d73ebc8fd9935d50742c
c2497543e509204cb587a218f8bd89ca4670342e
7927cd8dd0c5780

Data upload : 20 Bytes
Data download : 340 Bytes
Total time taken : 3045 miliseconds
```

Fig. 7 Obtaining invoice certificate (*crtV*) from store

Security modules

- RSA Key Generator 1,024 bits length of key pair
- Partially blind signature scheme
- RSA-based digital signature

7.2 Results

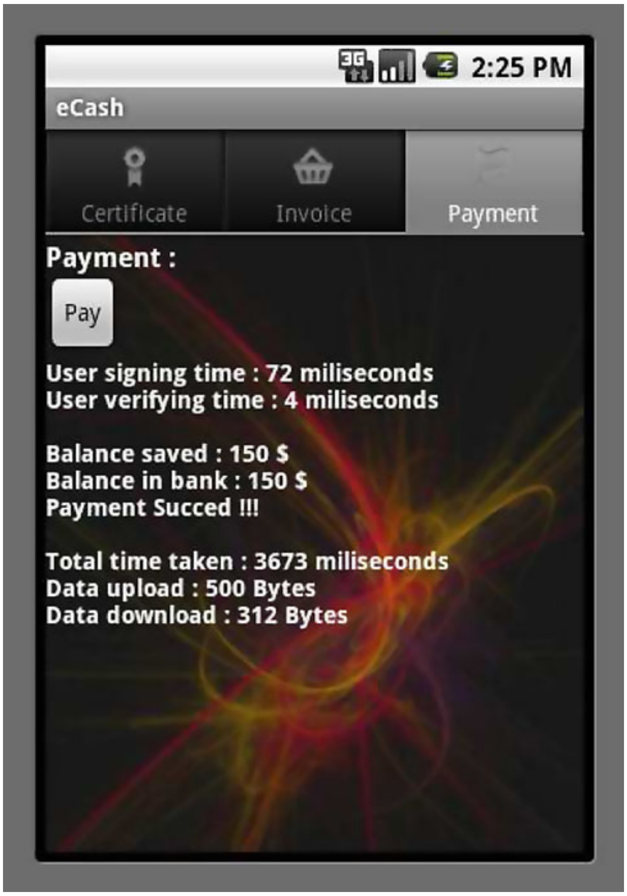
Figure 5 shows the Anonymous Pre-Paid Digital Cash Certificate with the amount value, expiry date, and bank name, and all these details are included in the cash certificate's digital signature generated by the bank.

Figure 6 shows behind the scene processing of Anonymous Pre-Paid Digital Cash Certificate Issuing Procedure. It displays the time taken by the customer's NSP to blind *eA* and establish a HTTPS connection with the bank. It displays the time taken by the bank to generate the signed Anonymous Pre-Paid Digital Cash Certificate, time taken by the customer to verify this certificate, un-blind the certificate, and verify the un-blinded certificate. It also displays the NSP's upload and download data sizes.

Figure 7 shows the digital *invoice certificate* generated by the merchant's RFID-kiosk, which is in turn downloaded by the NSP. It displays the unique *invoice ID*, the *invoice amount* and the *merchant's bank account details*, and all these details are included in the invoice certificate's digital signature generated by the merchant's RFID-kiosk. It also displays the NSP's upload and download data sizes.

Figure 8 shows the time taken by the NSP to generate and send the digital *cheque* to the bank, and the total time it takes for the NSP to receive the digital *paid receipt* from the bank. It also displays the NSP's upload and download data sizes.

We tried the emulation test 20 times, over wired internet as a proof-of-concept. Through our emulation, we calculated the time it takes to execute the (1) Cash Certificate (crtC) issuing procedure (Figs. 5, 6), (2) Obtaining Invoice Certificate (crtV) procedure (Fig. 7), and (3) Payment procedure (Fig. 8). The results are shown in the Table2. The payment procedure took an average execution time of 3.6 s and the payload received is of 500 bytes, this would not be a burden on any 3G/4G smartphones and their battery life. We believe that these procedures can be executed much faster when deployed on a real-world high-performance servers.



User signing time : 72 miliseconds
User verifying time : 4 miliseconds

Balance saved : 150 $
Balance in bank : 150 $
Payment Succed !!!

Total time taken : 3673 miliseconds
Data upload : 500 Bytes
Data download : 312 Bytes

Fig. 8 Pay and receive receipt procedure

Table 2 Execution time and data size

Procedure	Avg. execution time (ms)	Payload sent (bytes)	Payload received (bytes)
Certificate issuing	3,610	137	308
Invoice	3,394	20	340
Payment	3,601	312	500

7.3 Conclusion

In this chapter, we have proposed a privacy-preserving Pre-Paid Mobile HTTPS-based Payment model. Our proposed mobile payment model makes use of emerging technologies like the smartphone, RFID, and NFC. The proposed payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant. The consumer can cancel and obtain new anonymous pre-paid cash certificates whenever and wherever he/she wants using the smartphone's 3G/4G network. Our proof-of-concept implementation using Android SDK shows that our payment model can carried out by a smartphone within 4 s. Our future work would include a real practical implementation of our payment model using a NFC-enabled smartphone and developing a smartphone application that can communicate with the bank server via 3G/4G network and also communicate with a real NFC-module to download the merchant's bank account information and invoice.

Acknowledgments This work was partially supported by Brain Korea 21 (BK21) Project of the Korea Research Foundation (KRF) grants to Made Harta Dwijaksara.

References

1. Abe M, Okamato T (2000) Provably secure partially blind signature. In: proceedings of annual international cryptology conference. LNCS 1880:271–286
2. Balakrishnan M, Mohomed I, Ramasubramanian V (2009) Where's that phone?: geolocating IP addresses on 3G networks. Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, pp 294–300
3. Brands S (1993) Untraceable off-line cash in wallets with observers. In: Proceedings of annual international cryptology conference, pp 302–318, ISBN 3-540-57766-1
4. Cao T, Lin D, Xue R (2005) A randomized RSA-based partially blind signature scheme for electronic cash. Comput Secur 24–1:44–49
5. Chaum D, Fiat A, Naor M (1988) Untraceable electronic cash. In: Proceedings of annual international cryptology conference, pp 319–327, ISBN 3-540-97196-3
6. Chaum D (1982) Blind signatures for untraceable payments. In: Proceedings of annual international cryptology conference, pp 199–203
7. EPCglobal Inc website. http://www.EPCglobalinc.org
8. EPCglobal Specification, The EPCglobal architecture framework. http://www.epcglobalinc.org/standards/
9. Gartner Inc. (2009) Dataquest Insight: mobile payment, 2007–2012. http://www.gartner.com/it/page.jsp?id=995812
10. Hayashi F (2009) Do US consumers really benefit from payment card rewards?. Econ Rev, First Quarter, Federal Reserve Bank of Kansas City, https://www.kansascityfed.org/Publicat/ECONREV/PDF/09q1Hayashi.pdf
11. Heydt-Benjamin TS, Bailey DV, Fu K, Juels A, O'Hare T (2007) Vulnerabilities in first-generation RFID-enabled credit cards. In: Proceedings of eleventh international conference on financial cryptography and data security. LNCS 4886, pp 2–14
12. Internet Engineering Task Force (IETF), Network Working Group, Rescorla E (2000) "HTTP Over TLS", RFC2818. http://tools.ietf.org/html/rfc2818
13. ISO/IEC 14443-1∼4 (2008) Identification cards—contactless integrated circuit cards—proximity cards. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39693
14. ISO/IEC 18092, Near Field Communication Interface and Protocol (NFCIP-1). http://www.iso.org/iso/catalogue_detail.htm?csnumber=38578
15. Massouda N, Saundersb A, Scholnickc B (2010) The cost of being late? The case of credit card penalty fees. J Financ Stability. doi:10.1016/j.jfs.2009.12.001
16. MasterCard Worldwide, Tap & Go with MasterCard PayPass. http://www.paypass.com/
17. MasterCard Worldwide, MasterCard Pioneers Innovation in Payments with NFC Enabled Mobile Phones. http://www.

mastercard.com/hk/personal/en/wce/pdf/19755_Microsoft_Word_-_0411_-_HK-_NFC_release_-_Eng_-FINAL.pdf

18. Michael K, Burrows JH ELECTRONIC DATA INTERCHANGE (EDI). National Institute of Standards and Technology, 1996/04/29. http://www.itl.nist.gov/fipspubs/fip161-2.htm
19. National Institute of Standards and Technology (NIST) (2009) Digital Signature Standard (DSS), The Federal Information Processing Standards (FIPS) Publication 186–3. http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf
20. NFC Forum website. http://www.nfc-forum.org/home
21. Pritchard S (2009) Data lost, not found. Infosecurity 6-4:22–24
22. Roberds W, Schreft SL (2009) Data breaches and identity theft. J Monetary Econ 56-7:918–929
23. Schuhy S, Shyz O, Stavins J (2010) Who gains and who loses from credit card payments? Theory and calibrations. The Economics of Payments IV—Federal Reserve Bank of New York. http://newyorkfed.org/research/conference/2010/econ/reward28.pdf
24. Sweeney II PJ (2005) RFID for dummies. Wiley, ISBN: 0-7645-7910-X
25. VeriSign, The EPCglobal Network: Enhancing the supply chain. White Paper (2005). http://www.verisign.com/static/DEV044095.pdf
26. Vijayan J (2009) Heartland data breach sparks security concerns in payment industry. News article at Computerworld, http://www.computerworld.com/s/article/9126608/Heartland_data_breach_sparks_security_concerns_in_payment_industry
27. Visa USA, VISA PAYWAVE. http://usa.visa.com/personal/cards/paywave/index.html
28. Visa Europe, Visa Contactless—the wave and pay alternative to cash for low value transactions. http://www.visaeurope.com/pressandmedia/factsheets/visacontactless.jsp