A PTD-based transaction protocol for public transportation

Chin-Ling Chen*

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan 41349, ROC. E-mail: clc@mail.cyut.edu.tw.

SUMMARY

Advances in wireless network technology and the increasing number of users of Personal Trusted Device (PTD) users make the PTD an ideal channel for offering personalized services to mobile users. In this paper, we propose using a PTD as a payment tool in a mobile transaction system for public transportation. To overcome the inherent weakness of computing resources in a PTD, we use a trusted observer to coordinate the mobile transaction and to integrate cryptology (such as a digital signature and one way hash function).

The proposed scheme satisfies the requirements for mobile transactions. These requirements include fairness, non-repudiation, anonymity, off-line capability, no forgery, efficient verification, simplicity, and practicability. Because a PTD is more portable and personal than a personal computer and the because public transportation can be a necessity in our daily lives, our scheme proposes a novel use of PTDs in mobile commerce.

KEY WORDS: Fairness, non–repudiation, anonymity, hashing chain, mobile ticket, mobile commerce

* Corresponding author: Chin-Ling Chen, Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan 41349, ROC
E-mail: clc@mail.cyut.edu.tw (Chin-Ling Chen)
1. INTRODUCTION

Using personalized mobile devices to conduct business has become a general phenomenon in today’s society. Anybody can access various services using a Personal Trusted Device (PTD, such as a mobile phone or a personal digital assistant) anytime from anywhere. A PTD is more portable and personal than a personal computer, and thus is more convenient for certain applications. For example, people can use a PTD to access various services or to buy some digital products anytime from anywhere. In Finland, people even can even pay for a car wash using a mobile phone. Using a PTD as a payment tool has become a common practice in Finland, the country with the highest per capita cell phone use. A PTD can also be used as a payment tool for public transportation. Customers can buy the Add-Value cards, which are part of a “touch and go” IC ticketing system for metros, buses, and car parks, to use for the related services. Add-Value cards can be used for years, and the value added (automatically by Add-Value machines or manually by agents) and stored in a card will be debited automatically [1]. In such design, an Add-Value card leads to cost and waste. Thus, integrating current mobile devices and the subscriber identity module (SIM) for use as a payment tool is worth considering.

For example, public transportation companies can cooperate with a mobile network service provider (MNSP) to provide transportation services. People can carry a PTD with an Add-Value card (i.e. SIM card) to use for public transportation. However, it is a fact that PTDs lack computing resources [2-5]. If we attach a traditional transaction bill to a customer’s telephone bill directly, the account may cause problems for MNSP. Thus, attaching a transaction bill to a customer’s telephone bill is not practical. Instead, a customer should deposit money to an MNSP account or exchange paper money for digital cash before performing a mobile transaction. Then the MNSP can store the digital cash in a SIM card via public
loading center (for example: a kiosk or an ATM machine which can maintain a secure and private channel with the MNSP).

Current mobile transaction systems, which are based on the PTDs, almost neglect the importance of digital signatures. Thus, a buyer or the seller may repudiate a transaction. Because many value-added services are provided around the world, when a transaction bill increases or when a PTD is stolen, it becomes difficult to deal with. Therefore, many practical issues should be reconsidered. In this paper, we propose a novel protocol using PTDs as part of a mobile transaction system to pay for public transportation. In such a design, we involve an observer (the customer’s proxy) to construct a fair and non-repudiated transaction platform.

Generally speaking, the requirements of a mobile transaction system are similar to those of an for the e-cash system [6] and digital ticket issuance system [7-11]. Thus, we propose similar requirements for the public transportation transaction payments. We summarize below some requirements that have been presented before by other researchers.

(1) Fairness [12-13]: A fair system must ensure that unfair users will not gain any advantage over users who use system fairly.

(2) Non-repudiation [14-16]: Non-repudiation services protect the transacting parties against any false denials when a particular event or action has taken place, in which evidence will be generated, collected, and maintained to enable the settlement of disputes.

(3) Anonymity [7, 11]: No private information is disclosed to the service administrator or the verifier during the withdrawal and the consumption phase respectively.

(4) Off-line verification [9]: The verifier does not need to main a connection to the service administrator during verification.
(5) No forgery [6, 9]: Digital cash can not be forged.
(6) Efficient verification [9]: Verification should be fast.
(7) Simplicity [6]: As a mobile device lacks computing resource, mobile device operations should be designed as simple as possible.
(8) Practicability [9]: A useful protocol of a mobile transaction should be easily applied to current mobile communication systems without the need of extra infrastructures.

The rest of this paper is organized as follows. In section 2, we present our scheme for our mobile transaction protocol. In section 3, we will analyze the mentioned requirements. Finally, we give our conclusions in Section 4.

2. OUR SCHEME

The lack of computing resource in PTDs is an issue in mobile commerce. On the basis of our past researches on mobile commerce [8, 17-18], this problem needs to be solved. Thus, we involve a key role—observer to coordinate transaction scenarios. In this section, we will introduce our protocol.

In Figure 1, we show briefly the parties and messages involved in the whole process. Five parties are involved in our scheme. We will describe the related parties as follows.

- **Customer(C):** People who use PTDs, which store digital cash, as a payment tool to take public transportation.
- **Observer (O):** A trusted web site, which is committed to being a customer’s proxy.
- **Service Administrator (SA):** A cooperative society of public transportation providers, which cooperates with a mobile network service provider to provide
a public transportation service.

Mobile Network Service Provider (MNSP): A mobile network service provider, which provides a secure and stable wireless network. It also issues digital cash for a customer and clears the funds with the service administrator periodically.

Figure 1 The participating parties in a mobile transaction and coordinating messages in whole process

- **Mobile Network Service Provider (MNSP):** A mobile network service provider, which provides a secure and stable wireless network. It also issues digital cash for a customer and clears the funds with the service administrator periodically.
• **Verifier (V):** A ticket-checking server, which verifies the customer’s digital cash chain.

Most of the mobile commerce networks often involve Wireless Transport Layer Security Specification (WTLS) [19] and the Internet Secure Socket Layer (SSL) protocol [20-21] to protect end-to-end communication. In our proposed scheme, we integrate these security mechanisms as well as some cryptology (such as public key infrastructure, hashing chain, and digital signature) to propose a realistic mobile transaction scheme.

Below, we introduce the notations which are used in our scheme.

$+$ : additive operation.

$\oplus$ : bitwise exclusive-OR operation.

$h(\cdot)$ : a collision-free one-way hash function, which can be used to generate a series of non-reversible hashing chain.

$a_0, b_0$ : two random seed numbers which are used to generate two sets of hashing values $(a_1, a_2, ..., a_n)$ and $(b_1, b_2, ..., b_n)$, where $a_1 = h(a_0)$, $a_2 = h(a_1)$, \ldots , $a_n = h(a_{n-1})$ and $b_1 = h(b_0)$, $b_2 = h(b_1)$, \ldots , $b_n = h(b_{n-1})$.

$PID_C$ : a customer’s pseudonym that is generated by the MNSP.

$ID_X$ : the identity of party $X$.

$x$ : current ticket price, in $x$ units, required for a customer to take public transportation.

$M$ : $M$ dollars which are deposited in an MNSP account.

$n$ : $n$ units of digital cash that are withdrawn by a customer, where $n \leq M$.

$t$ : the timestamp.

$r_{\text{cash}}$ : the current remaining digital cash of a customer.

$chain_i$ : the current digital cash chain of a customer, $chain_i = h^{'(chain_{i+1})}$, where
chain\textsubscript{i+\textsuperscript{x}} is the digital cash chain of the previously spent ticket price in \textit{x}' units.

\textit{S\textsubscript{X}(m)} : uses the \textit{X}'s secret key to sign message \textit{m}.

\textit{V\textsubscript{X}(m)} : uses the \textit{X}'s public key to verify message \textit{m}.

\textit{Sig\textsubscript{X}} : \textit{X} party’s signature.

\textit{msg\textsubscript{x}} : the message \textit{x}, for example: \textit{msg\textsubscript{req}} means the requested message.

Initially, a customer pre-coordinates two sets of hashing values, \((a_1, a_2, \ldots, a_n)\) and \((b_1, b_2, \ldots, b_n)\), with the MNSP and the observer, respectively. On the basis of the indivisible hash chain, the MNSP and the observer can use it to authenticate messages to each other during a mobile transaction.

2.1. Registration phase

\textit{Step1}: Before processing a mobile transaction, each mobile customer should register first. To keep a customer anonymous, the MNSP creates a unique pseudonym, \textit{PID\textsubscript{C}}, for the customer. The mobile customer needs to deposit \textit{M} dollars in an MNSP account, and then the MNSP only permits the mobile customer to withdraw the digital cash from a loading center (for example, a kiosk or an ATM machine). On depositing \textit{M} dollars in the MNSP account, the digital cash \textit{r\_cash} in the customer’s database is updated to \textit{M} dollars. The MNSP computes \(h\text{\_value} = h(PID\text{\textsubscript{C}}, \text{\textit{r\_cash}})\). Then the MNSP stores the \((PID\text{\textsubscript{C}}, \text{\textit{r\_cash}}, h\text{\_value})\) in its database.

\textit{Step2}: The customer can query and verify the deposit from the loading center or web site as follows:

\[ h(PID\text{\textsubscript{C}}, M) = h\text{\_value} \]
2.2. Withdrawal phase

Step1: Before using the PTD as a payment tool, the customer needs to make a withdrawal request with the MNSP. The customer makes a withdrawal request for \( n \) units of digital cash, as follows:

\[
msg_{\text{req}} = (msg_{\text{withdrawal}}, ID_O, ID_{\text{MNSP}}, PID_c, n)
\]

\[
X_{\text{req}} = (msg_{\text{req}}) \oplus b_i + b_{i-1}
\]

Then he sends \((msg_{\text{req}}, X_{\text{req}})\) to the observer.

Step2: The observer uses the pre-coordinate hashing values \( b_i \) and \( b_{i-1} \) to authenticate the customer’s request, as follows:

\[
(msg_{\text{req}}) \oplus b_i + b_{i-1} = X_{\text{req}}
\]

The observer uses his secret key to sign the customer’s withdrawal request, as follows:

\[
Sig_{O-\text{withdrawal}} = S_O(msg_{\text{req}})
\]

Then the observer sends \((msg_{\text{req}}, Sig_{O-\text{withdrawal}})\) to the MNSP.

Step3: The MNSP uses the observer’s public key to verify the withdrawal signature, as follows:

\[
V_O(Sig_{O-\text{withdrawal}}) = msg_{\text{req}}
\]

If the currently available digital cash \( r_{\text{cash}} \geq n \), then the MNSP updates the remaining digital cash in the customer’s database, as follows:

\[r_{\text{cash}} = r_{\text{cash}}' - n\]

where \( r_{\text{cash}}' \) is the remaining digital cash after the last withdrawal.

If \( r_{\text{cash}} \leq n \), the MNSP rejects the withdrawal request.

The MNSP computes the \( n \) units of the digital cash chain, as follows:
\[ \text{chain}_i = h^n (\text{msg}_{\text{req}}) \]

The MNSP uses the MNSP’s secret key to issue the digital cash, as follows:

\[ \text{digital} \_ \text{cash} = S_{\text{MNSP}}(\text{chain}_i, ID_o, PID_c, r \_ \text{cash}, n, t) \]

Then the MNSP sends \((\text{digital} \_ \text{cash}, \text{chain}_i, ID_o, PID_c, r \_ \text{cash}, n, t)\) to the observer.

\textbf{Step 4:} The observer uses the MNSP’s public key to verify the MNSP’s signature (the issued digital cash), as follows:

\[ \overset{?}{V}_{\text{MNSP}}(\text{digital} \_ \text{cash}) = (\text{chain}_i, ID_o, PID_c, r \_ \text{cash}, n, t) \]

Then the observer determines whether the issued \text{digital} \_ \text{cash} meets the customer’s withdrawal request, as follows:

\[ h^n (\text{msg}_{\text{req}}) = \text{chain}_i \]

\[ r \_ \text{cash} = r \_ \text{cash}' - n, \text{ where } r \_ \text{cash}' \text{ is the remaining digital cash after the last withdrawal.} \]

The observer verifies result, as follows:

\[ \text{msg}_{\text{verify}} = (\text{yes/no, digital} \_ \text{cash, chain}_i, ID_o, PID_c, r \_ \text{cash}, n, t) \]

where “yes/no” indicates whether the customer’s withdrawal request equals the issued \text{digital} \_ \text{cash} issued by the MNSP. Then the observer uses the next pre-coordinate hashing values \(b_{i-1}\) and \(b_{i-2}\) to compute

\[ X_{\text{msg}_{\text{verify}}} = (\text{msg}_{\text{verify}}) \oplus b_{i-1} + b_{i-2} \]

The observer then sends the verified message \((\text{msg}_{\text{verify}}, X_{\text{msg}_{\text{verify}}})\) to the customer.

\textbf{Step 5:} After the customer makes the withdrawal request, the MNSP sends \((\text{digital} \_ \text{cash, chain}_i, ID_o, PID_c, r \_ \text{cash}, n, t)\) to the SA. Afterward, the SA
uses the MNSP’s public key to verify the digital cash, as follows:

\[ V_{\text{MNSP}}(\text{digital_cash}) = (\text{chain}_i, \text{ID}_D, \text{PID}_C, r_{\_\text{cash}}, n, t) \]

The current digital cash chain \( \text{chain}_i \) and \( r_{\_\text{cash}} \) of the customer \( \text{PID}_C \) is recorded in the SA’s database.

2.3. Payment phase

**Step1**: The customer uses the pre-coordinate hashing values \( b_{i-1} \) and \( b_{i-2} \) to authenticate the message \( \text{msg}_{\text{verify}} \), which is verified by the observer, as follows:

\[ X_{\text{msg}_{\text{verify}}} \oplus b_{i-1} + b_{i-2} = \text{msg}_{\text{verify}} \]

Afterward, the customer can use the PTD, which stores the digital cash \( \text{digital_cash} \) (which is included in \( \text{msg}_{\text{verify}} \)), to pay for public transportation. Subsequently, the customer writes the payment message, as follows:

\[ \text{msg}_{\text{cons}} = \left( \left( \text{digital}_\_\text{cash}, \text{chain}_i, \text{ID}_D, \text{PID}_C, r_{\_\text{cash}}, n, t, x, \text{chain}_{i+x} \right) \right) \]

where \( \text{chain}_i = h^x(\text{chain}_{i+x}) \) is the current digital cash chain of the customer; \( \text{chain}_{i+x} \) is the previously spent \( x' \) units of digital cash, and \( x \) is the current unit of digital cash spent. The PTD writes the messages \( \text{MXchain} \) and \( \text{OXchain} \) with the hashing values \((a_i, a_{i-1})\) and \((b_{i-2}, b_{i-3})\) for the MNSP and the observer, relatively, to verify these messages.

\[ \text{MXchain} = (\text{msg}_{\text{cons}}) \oplus a_i + a_{i-1} \]

\[ \text{OXchain} = (\text{msg}_{\text{cons}}) \oplus b_{i-2} + b_{i-3} \]

The customer sends \((\text{msg}_{\text{cons}}, \text{MXchain}, \text{OXchain})\) to the verifier via bluetooth or infrared short-range radio technology.

**Step2**: The verifier checks the \( \text{digital_cash} \) (which is included in \( \text{msg}_{\text{cons}} \)), as
follows:

\[ V_{\text{MNSP}}( \text{digital} \_\text{cash}) = (\text{chain} \_\text{i}, \text{ID} \_\text{c}, \text{PID} \_\text{c}, r \_\text{cash}, n, t) \]

Moreover, the verifier verifies whether or not the current ticket price \( x \) meets the customer’s withdrawal request, as follows:

\[ r \_\text{cash} \geq x \]

\[ \text{chain} \_i = h^x (\text{chain} \_i + x') \]

If the customer passes the above authentications, then the verifier permits the customer to take the public transportation. The verifier writes a signature \( \text{Sig}_{V\_\text{cons}} \), as follows:

\[ \text{Sig}_{V\_\text{cons}} = S_V(\text{msg}_{\text{cons}}) \]

and then forwards \((\text{msg}_{\text{cons}}, \text{Sig}_{V\_\text{cons}}, \text{MXchain}, \text{OXchain})\) to the SA for further verification. Otherwise, the verifier rejects the request.

**Step 3:** Upon receiving the payment messages \((\text{msg}_{\text{cons}}, \text{Sig}_{V\_\text{cons}}, \text{MXchain}, \text{OXchain})\), the SA uses the verifier’s public key to verify the verifier’s signature, as follows:

\[ V_V(\text{Sig}_{V\_\text{cons}}) = \text{msg}_{\text{cons}} \]

If it is authentic, the SA writes a signature \( \text{Sig}_{S_A\_\text{cons}} \), as follows:

\[ \text{Sig}_{S_A\_\text{cons}} = S_{S_A}(\text{msg}_{\text{cons}}) \]

and then forwards \((\text{msg}_{\text{cons}}, \text{Sig}_{S_A\_\text{cons}}, \text{MXchain}, \text{OXchain})\) to the MNSP to verify, using batch processing way, whether or not the customer is double-spending.

**Step 4:** The MNSP uses the SA’s public key to verify the received signature, as follows:

\[ V_{S_A}(\text{Sig}_{S_A\_\text{cons}}) = \text{msg}_{\text{cons}} \]
If it is authentic, the MNSP uses the pre-coordinate hashing values $a_i$ and $a_{i-1}$ to verify the received message, as follows:

$$msg_{cons} \oplus a_i + a_{i-1} = MXchain.$$  

The MNSP gets the last used digital cash chain $chain_{i+x'}$ from the received payment message $msg_{cons}$. Subsequently, MNSP compares $h^x(chain_{i+x'})$ with the current database digital cash chain $chain_i$ of the customer $PID_C$, as follows:

$$chain_i^? = h^x(chain_{i+x'})$$

If the above equality holds, then the MNSP updates the remaining digital cash in the customer’s database such that $r_{_cash} = r_{_cash-x}$ and the current digital cash chain $chain_i = h^x(chain_{i+x'})$. Simultaneously, the MNSP checks whether the customer occurs is double-spending. The MNSP then uses its private key to sign this payment message, as follows:

$$Sig_{MNSP-cons} = S_{MNSP}(yes/no, msg_{cons}),$$

where “yes/no” indicates whether the status of the customer is in double-spending status.

Then the MNSP sends $(yes/no, msg_{cons}, Sig_{MNSP-cons})$ to the SA. The SA records MNSP’s signature $Sig_{MNSP-cons}$ and the payment message $msg_{cons}$. Thus, the SA can see whether or not the customer’s credit is good.

**Step 5:** The MNSP writes a notification message, as follows:

$$msg_{notif} = (PID_C, r_{_cash}, chain_i)$$

$$MXmsg = msg_{notif} \oplus a_{i-1} + a_{i-2}$$

and then sends $(msg_{notif}, MXmsg)$ to the customer. Therefore, the customer can use the pre-coordinate hashing values $a_{i-1}$ and $a_{i-2}$ to verify the
received message, as follows:

\[ msg_{\text{notif}} \oplus a_{i-1} + a_{i,2} = MXmsg \]

The customer can see the remaining digital cash \( r_{-\text{cash}} \) and the currently used digital cash chain \( chain \) after each transaction. Thus, the customer can determine whether or not to re-deposit the digital cash for the next payment.

2.4. Clearing funds phase

**Step 1**: The MNSP forwards \( (msg_{\text{cons}}, Sig_{\text{MNSP-con}}) \) to the observer.

**Step 2**: The observer uses the MNSP’s public key to verify the payment message, as follows:

\[ V_{\text{MNSP}}(Sig_{\text{MNSP-con}}) = (\text{yes/no}, msg_{\text{cons}}) \]  \hspace{1cm} (1)

Next, the observer uses the pre-coordinate hashing values \( b_{i-2} \) and \( b_{i,3} \) to verify the payment message, as follows:

\[ msg_{\text{cons}} \oplus b_{i-2} + b_{i,3} = \text{OXchain} \]  \hspace{1cm} (2)

Only Equations (1) and (2) have equalities. The observer is a customer’s proxy to commit the transaction and uses his own private key to sign the payment message and get the signatures, as follows:

\[ Sig_{\text{O-con}} = S_O(msg_{\text{cons}}). \]

Then the observer sends \( (msg_{\text{cons}}, Sig_{\text{O-con}}) \) to the MNSP.

**Step 3**: The MNSP uses the observer’s public key to verify the received signature, as follows:

\[ V_O(Sig_{\text{O-con}}) = msg_{\text{cons}} \]

If the above equation has equality, the transaction enters the e-billing
management system; the transaction bill will be attached into the customer’s database. In addition, the MNSP should clear the funds with the SA periodically.

3. ANALYSIS

We proposed a mobile transaction system for public transportation based on PTDs. Now we want to examine whether or not the issues mentioned before are satisfied.

3.1. Fairness issues

In traditional commercial behavior, the “fairness” means to achieve the aim of “cash on delivery.” However, this is a difficult goal to achieve in web transactions. To ensure that some users do not gain any advantages over other users who use the transaction fairly and correctly, the transaction are always designed such that the buyer and seller receive verifiable proof of each other during the transactions. Therefore, a digital signature is used to give verifiable proof of each relevant party. However, it is not suitable for mobile devices to execute exponential operations. For this reason, we involve a trusted observer to coordinate the transactions and to apply the indivisible hashing chain mechanism such that a fair transaction platform can be established.

In the registration phase, when a customer deposits $M$ dollars in an MNSP account or exchange paper money for digital cash, the customer can check the digital cash $r_{\text{cash}}$ via the following simple operation:

$$h(PID_c, r_{\text{cash}}) = h_{\text{value}}$$

On the other hand, the customer can use the pre-coordinate hashing values $(b_i$ and $b_{i-1})$ and simple operations to request a withdrawal $(msg_{\text{req}}, X_{\text{req}})$ for the
observer to authenticate. The observer can authenticate the customer’s withdrawal request, as follows:

$$(msg_{req}) \oplus b_i + b_{i-1} = X_{req}$$

In the same way, the customer can check whether the message $msg_{verify}$ has been verified by the observer, as follows:

$$Xmsg_{verify} \oplus b_{i-1} + b_{i-2} = msg_{verify}$$

Afterward, the customer only writes the following payment messages:

$$msg_{cons} = ((digital\_cash, chain, ID_O, PID, r\_cash, n, t), x, chain)$$

$$MX\text{chain} = (msg_{cons}) \oplus a_i + a_{i-1}$$ and

$$OX\text{chain} = (msg_{cons}) \oplus b_{i-2} + b_{i-3}$$

for the MNSP and the observer to authenticate. In our scheme, the observer is selected freely by the customer. The designated observer is the customer’s proxy who signs each mobile transaction request and verifies whether or not the other party’s signature is authentic, as follows:

$$Sig_{O-\text{withdrawal}} = S_O(msg_{req})$$ (in withdrawal phase)

$$V_{MNSP}(digital\_cash) = (chain, ID_O, PID, r\_cash, n, t)$$ (in withdrawal phase)

$$Sig_{O-\text{cons}} = S_O(msg_{cons})$$ (in clearing funds phase)

$$V_{MNSP}(Sig_{MNSP-\text{cons}}) = (yes/no, msg_{cons})$$ (in clearing funds phase)

Because of the observer’s participation, the MNSP or SA dare not forge the signature. Based on the pre-coordinate hashing values and the non-reversible property of the hashing chain, the customer cannot be denied a mobile transaction that is signed by the observer and confirmed by the customer. All users will leave traceable evidence such that a fair platform can be constructed. The observer plays a key role in
establishing a fair situation so that dishonest behavior will not occur.

3.2. Non-repudiation issues

According to the non-repudiation issue, if each transaction scenario can leave a traceable proof, the relevant parties cannot deny anything they have done. Based on the characteristics of the hashing chain, the customer’s request is bound by the indivisible set of hashing values \((b_1, b_2, ..., b_n)\). The mobile customer and observer can authenticate each other’s messages, and neither one can later deny having made the request. Therefore, when the mobile customer makes a withdrawal request, the observer will verify it and then sign this withdrawal request message for the MNSP, as follows:

\[
(m \text{_{req}}) \oplus b_i \oplus b_{i-1} = X_{\text{req}}
\]

\[
\text{Sig}_{O-\text{withdrawal}} = S_O(m \text{_{req}})
\]

Later, the MNSP can issue the digital cash based on the observer’s signature. In addition, the observer acts as a proxy to verify whether or not the digital cash \(\text{digital\_cash}\) (which is included in the \(m \text{_{verify}}\)) is valid.

\[
V_{\text{MNSP}}(\text{digital\_cash})=(chain_i, ID_O, PID_C, r_{\text{cash}}, n, t)
\]

\[
h^e(m \text{_{req}}) = \text{chain}_i
\]

Afterward, the mobile customer can use the PTD, which stores the digital cash, to take public transportation. In our scheme, when the mobile customer makes the service request, the SA will forward the payment message to the MNSP to re-compute the remaining digital cash \(r_{\text{cash}} = r_{\text{cash}} - x\). In addition, the customer’s payment request is also bound by the hashing chain. The customer’s payment can be authenticated by the MNSP, as follows:
Neither one can later deny having processed the service. The MNSP sends back a response message \( \text{Sig}_{\text{MNSP-cons}} = S_{\text{MNSP}}(\text{yes/no}, \text{msg}_{\text{cons}}) \) to the SA as the evidence of the clearing of funds between the MNSP and SA. Therefore, there will be evidence of each mobile customer’s transaction, and the database will be updated. Simultaneously, the MNSP will send \( (\text{msg}_{\text{notif}}, \text{MXmsg}) \) to the customer. That is, the customer will be notified of his remaining digital cash after each transaction. Our scheme, thus, will not suffer from non-repudiation issues.

### 3.3. Anonymity issues

In the payment phase, the verifier or the SA only checks the \( \text{digital\_cash} \) and current ticket price \( x \). We use the pseudonym mechanism to achieve anonymity. The customer’s identity will not be exposed to the verifier or the SA during the transaction.

### 3.4. Off-line verification issues

In consideration of network bottleneck, a practical mobile system should be designed taking into consideration an off-line model. In our scheme, when a mobile customer uses a PTD that stores the digital cash to take public transportation, the verifier only verifies whether the remaining cash \( r\_\text{cash} \) and the ticket price \( x \) is correct:

\[
? r\_\text{cash} \geq x
\]

\[
? \text{chain}_i = h^x(\text{chain}_{i+x'})
\]

With regard to double-spending verification, we left this problem for the MNSP to check during batch processing. That is, the verifier, the SA, and the MNSP...
periodically synchronize their databases to each other off-line. Such a design is similar to the e-cash system.

3.5. No forgery issues

Because to the digital cash digital_cash is issued by the MNSP, no one can generate a valid digital cash signature (where 
\[ digital\_cash = S_{MNSP}(\text{chain}_t, \text{ID}_o, \text{PID}_c, \text{r\_cash}, n, t) \] ) except the MNSP. It is actually infeasible for anyone to forge the signature without knowing the MNSP’s secret key.

3.6. Efficient verification issues

In our scheme, when a mobile customer uses his PTD to take public transportation, the verifier only checks whether the remaining cash is greater than the ticket price \( x \) ( \( \text{r\_cash} \geq x \) ) and whether the current digital cash chain \( \text{chain}_i \) is correct or not ( \( \text{chain}_i = h^x(\text{chain}_{i+x'}) \) ). This is efficient.

After the transaction, the customer’s current digital cash chain \( \text{chain}_i \) and the available remaining digital cash \( \text{r\_cash} \) will be re-computed, and the database batch processing will be refreshed off-line. Therefore, when the MNSP’s database synchronizes with the databases of the SA and the verifier, double-spending will be detected. If dishonest behavior is discovered, the customer will be added to a watch list. The customer may be refused public transportation in the future.

Simultaneously, authentication between customer and observer, or between customer and MNSP, depends on the pre-coordinated hashing chain. In this way, the identification mechanism is efficient. Our scheme does not suffer from inefficient verification.
3.7. Simplicity issues

Since PTDs do not have much computing resource, the client side only performs a simple operation (such as exclusive-OR and additive operations, but no exponential operations). This can be easily implemented into the current PTD hardware design. Our scheme only uses the lightweight operations; and will not suffer from complex operations. It is designed to use the simple operations to speed up the verification performance.

3.8. Practicability issues

We proposed a mobile transaction system to use for public transportation. It meets the mentioned requirements. Our scheme is suitable for performing mobile transactions using PTDs that lack computing resource. Practically, the proposed system provides a fair, non-repudiable, and secure platform. It can be easily applied to current mobile communication system without the need for extra infrastructures.

4. CONCLUSIONS

The PTD is becoming an extremely popular tool for people to conduct their business anytime from anywhere. Such a device cannot only be used to communicate with other people but can also be used as a payment tool; it has become an attractive business tool. Although the PTD provides more portable, personal, flexible and dynamic environment than the personal computer, it still has many inherent disadvantages that need to be overcome.

In this paper, we proposed a scheme that is used in mobile transaction systems, and a practical protocol that uses PTD for public transportation systems. In our scheme, we involve an observer (the customer’s proxy) to coordinate mobile
transactions such that that fairness, non-repudiation, anonymity, off-line capability, no forgery, efficient verification, simplicity and practicability issues can be guaranteed.

Further applications using the PTD can include paying parking tickets and using coupon. Our scheme can facilitate mobile commerce, and it will bring more profits for mobile transaction service providers. People will also find it convenient to conduct their business via PTD in the future.

ACKNOWLEDGEMENTS
This research was supported by National Science Council, Taiwan, Republic of China under contract number NSC 95-2221-E-324-025. The referees’ insightful comments helped to improve the paper significantly.

REFERENCES
1. Taipei Smart Card Corporation (TSCC),


15. Cox B., Tygar J. D. and Sirbu M., NetBill Security and Transaction Protocol,
16 Zhou J. and Gollmann D., Observations on non-repudiation, Lecture Notes in
Kyongju, Korea, 1996; Vol. 1163, 133-144.

17. Chen Y.Y., Jan J.K. and Chen C.L., A Fair and Secure Mobile Billing System,

18. Chen C. L., Lin H.Y., Chen Y.Y., and Jan J. K., A Fair Transaction Model in
Processing and Information Technology (ISSPIT 2006)*, Vancouver, Canada,
Aug. 27 - 30, 2006; 484-489.


http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html


21. Wagner D. and Schneier B., Analysis of the SSL 3.0 protocol, *Proceedings of
the second USENIX Workshop on Electronic Commerce*, USSENIX Press,
Oakland, California, USA. November, 29-40, 1996.
AUTHOR BIOGRAPHY

Chin-Ling Chen was born in Taiwan in 1961. He received the B.S. degree in Computer Science and Engineering from the Feng Cha University in 1991; the M.S. degree and Ph.D. in Applied Mathematics at National Chung Hsing University, Taichung, Taiwan, in 1999 and 2005 respectively. He is a member of the Chinese Association for Information Security. From 1979 to 2005, he was a senior engineer at the Chunghwa Telecom Co., Ltd. He is currently an assistant professor of the Department of Computer Science and Information Engineering at Chaoyang University of Technology, Taiwan. His research interests include cryptography, network security, communication network and electronic commerce.