A Thesis for the Degree of Master

A Study on Micro-payment Systems for Multiple-shopping

Manho Lee

School of Engineering

Information and Communications University

2002

A Study on Micro-payment Systems for Multiple-shopping

A Study on Micro-payment Systems for Multiple-shopping

Advisor : Professor Kwangjo Kim

by

Manho Lee
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejon, Korea
Jan. 11. 2002
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

A Study on Micro-payment Systems for Multiple-shopping

Manho Lee

We certify that this work has passed the scholastic standards required by the Information and Communications University as a thesis for the degree of Master

Jan. 11. 2002

Approved:

Chairman of the Committee
Kwangjo Kim, Associate Professor
School of Engineering

Committee Member
Younghee Lee, Professor
School of Engineering

Committee Member
Chulsoo Lee, Invited Professor
School of Engineering

M.S. 2000063
Manho Lee
A Study on Micro-payment Systems for Multiple-shopping
School of Engineering, 2002, 42p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

Researches on the secure e-cash systems have been carried on with the proliferation of the Internet and e-commerce. Subsequently, various systems and schemes have been proposed in the literature.

Micro-payment systems have been proposed for small and frequent payment such as database query, software, and e-news different from the established payment system (so called Macro-payment). Micro-payment system is considered to be proper for areas where efficiency is the most critical factor. Because the overhead cost for card company is higher than the price of goods, and the processing time related to public key computation is required too much, most of the previous systems exclude public key computation or admit the minimal computational overhead for efficiency.

From the security point of view, micro-payment system is mainly concerned about preventing or detecting from double spending and forgery.

Assurance of atomicity is strictly needed as well to resolve potential problems, for example, the loss of money from the network failure.

In this thesis, we propose a new micro-payment system that satisfies

basic requirements of micro-payment systems and atomicity. Explicitly, this system can be considered as an improvement and extension of PayWord which is one of the most popular micro-payment systems [24] suggested by Rivest and Shamir.

Our proposed system makes use of only one hash chain but is applicable to multiple vendors. Since PayWord requires using each hash chain for each vendor, the customer has to maintain all indices after dealing with vendors. To solve this problem, we use a receipt which includes a signature of vendor. The customer uses this signature for different vendors to enable them to verify whether the starting index of payment is correct or not. Moreover, we focus on the system that guarantees atomicity and non-repudiation among requirements for payment systems. Atomicity and non-repudiation play an important role in payment systems. In particular, atomicity allows customers to use the payment system without loss of their money in any disaster.

Finally, we analyze not only the efficiency in terms of the storage size but also the security issues in case that a customer uses the micro-payment to several vendors.

Contents

List of Tables

List of Figures

List of Abbreviations

e-cash Electronic Cash

EPO Electronic Purchase Order

μ-i**KP** Micro-payment based on Internet Keyed Payments Protocol

MAC Message Authentication Code

PKI Public Key Infrastructure

UMTS Universal Mobile Telecommunications System

List of Notations

C, B, V Customer, Broker, Vendor

Z entity $Z \in \{C, B, V\}$

Z_{ID} Z's identity

PK_Z, SK_Z Z's public key, private key

$[Data]_K$ symmetric encryption scheme

$\{Data\}_{PK_z}$, $\{Data\}_{SK_z}$ public key encryption scheme with Z's public key and decryption scheme with Z's private key

$Sign_Z\{Data\}$ signature of Z

$X \Rightarrow Y$ X sends one message to Y

$h()$ collision-free hash function

C_Z a Z's certification made from broker

E expiration period of certificate

r_Z a random number made by Z

A_C an IP-address of customer

I_C other customer specific information

M a commitment of customer

P a payment from customer to vendor

$||$ concatenation of messages

I. Introduction

1.1 Electronic Cash System in On-line Business

Many electronic cash ("e-cash" in short) protocols have been proposed with the proliferation of the Internet and the activation of e-commerce. A secure and efficient e-cash system plays an important role to support these businesses safely as a trustful payment over the Internet.

Since Chaum [7] proposed untractable e-cash protocol based on blind signature, various extended schemes and systems have been proposed, which provide functionalities such as anonymity, double spending prevention, unforgeability, untraceability, and efficiency. The e-cash systems [19, 20] by Okamoto and Ohta satisfy the divisibility and transferability in addition. Brands [2] proposed an efficient e-cash system with single-term method which is more efficient compared with the cut-and-choose method. Brands' scheme has been used as a basic model by other researches.

However, Solms and Naccache [27] raised perfect crime by abusing the anonymity of the e-cash system. Other malicious actions such as blackmailing and money laundering were also presented in [3, 14, 21, 27]. The revocable e-cash system [3, 5] in which anonymity can be revoked when needed, becomes one of the active research areas to prevent such misuses. Along with countermeasures [3, 26] against the blackmailing and money laundering, many schemes in [5, 9, 14, 18, 21, 4, 8] have been proposed to resist against the abuse of anonymity. These protocols are designed as medium for large transaction (macro-payment) over 10$.

On the other hand, micro-payment systems have been proposed for small and frequent payment such as database query, software, and e-news. Because the overhead cost for a card company is higher than price of goods, and processing time related to public key computation is required too much, most of the previous systems exclude public key computation or admit the minimal computational overhead for efficiency. Several protocols that use computationally secure one-way hash function were proposed in [10, 11, 22, 24]. Stern and Vaudenay's system [29] used MAC (Message Authentication Code) and an unkeyed one-way hash function instead of using public key cryptography and adopted tamper resistant devices for both customer and merchant sides. MilliCent [17] used "scrips" and hash functions extensively. MicroMint [24] used k-way hash function collision to make money. Micro-payment system is considered to be proper for areas where efficiency is the most critical factor.

Even if a customer loses his e-cash, it is not critical to him due to the small amount. Though an attacker forfeits the e-cash of a customer, computational cost is too big to gain a profit from the attack. Although damage done by system errors or attacks is trivial, customers do not want to lose his money, or goods. To solve these problems, protocols in [1, 6, 28, 30, 31] were proposed to satisfy atomicity.

From the security point of view, micro-payment system is mainly concerned about preventing or detecting from double spending and forgery.

In this thesis, we propose a new micro-payment system, in which a customer generates and maintains only one hash chain for every vendor. Since PayWord requires to use each hash chain for each vendor, the customer has to maintain all indices after dealing with vendors. To solve this problem, we use a receipt which includes a signature of vendor. The customer uses this signature for different vendors to enable them to verify whether the start index of payment is correct or not. Moreover, we design each phase of protocol to guarantee atomicity.

1.2 Organization of the Thesis

In this thesis, we propose a micro-payment system that satisfies atomicity and can use for multiple-shopping. Explicitly, this system is an improvement and extension of PayWord [24].

This thesis is organized as follows. Chapter 2 introduces micro-payment and atomicity. PayWord, One of micro-payment protocol is described in Chapter 3. In Chapter 4, a micro-payment system for multi-shopping is proposed, and the properties which are atomicity, security features, and efficiency are analyzed. Conclusions will follow in Chapter 5.

II. Preliminaries

In this chapter, we summarize the micro-payment systems. We describe basic requirements which are considered when they are designed, and atomicity property.

2.1 An Overview of Micro-payment

Basically, a micro-payment system is composed of three entities, *e.g.*, customer, vendor, and broker.

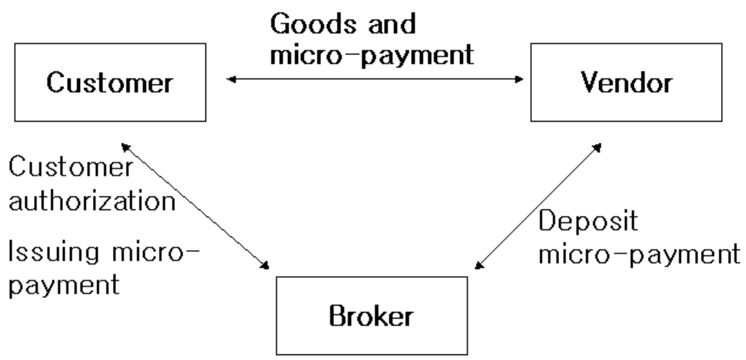

Figure 2.1: A basic architecture of micro-payment

Fig. 2.1 shows a basic architecture of micro-payment system.
Three entities play basic role as follows.

- Customer : does business with vendors by using micro-payment system.
- Vendor : offers customers goods and deposits e-cash through a broker.

- Broker : issues e-cash, validates customers, and manages accounts of customers and vendors.

2.2 Basic Requirements

Micro-payment system is considered to be proper for areas where efficiency is the most critical factor. Along with the efficiency factor, from the security point of view, preventing or detecting from double spending and forgery are mainly concerned.

- Efficiency
  - *Minimization of public key computation* : We do not use public key computation at all or use as few as possible to reduce high cost of public key computation.
  - *Off-Line Verification* : A broker doesn't carry on on-line verification due to limitation of the bandwidth of communication-line during the transaction.
  - *Minimum usage of message* : The number of messages among entities, especially between customers and vendors, should be kept lower during transaction.
  - *Minimum usage of memory space* : Permanent variables should be used as few as possible.
- Security
  - *Double spending prevention or detection* : If the same e-cash is used more than once, then it should be detected.
  - *Unforgeability* : Only publishers who have an authority should be able to issue e-cash.

2.3 Atomicity

In addition to basic requirements, atomicity is also of another importance in serving a complete payment system. If network crashes in the middle of payment protocol, some disputes such as loss of money, repudiation, and no procurement of goods even after a complete payment will happen. Under a network model where customers purchase electronic goods and receive its service, Tygar [30] introduced the problem of atomicity in electronic transactions and defined three classes of atomicity; money atomic, goods atomic, and certified delivery.

- Money atomicity : The transfer of funds from one entity to another should be done without creation and loss of money.
- Goods atomicity : With the money atomic, it should be ensured that customers will receive goods if and only if the money transferred.
- Certified delivery : Including goods and money atomic, both customers and vendors should be able to prove exactly what was delivered. If there is a dispute, this evidence can be shown to a judge to prove exactly what goods were delivered.

Camp *et al.* [6] proposed a method to support atomicity using online TTP (Trusted Third Party). They adopted *two-phase commit* idea which is well-known in the area of database systems. Also, anonymity is considered in their paper. Moreover, Sirbu and Tygar [28], and Bellare *et al.* [1] presented on-line account-based atomicity, and token-based atomicity, respectively. Xu *et al.* [31] addressed money conservation via atomicity in fair off-line e-cash. They assumed some possible scenarios in each phase of e-cash protocol, withdrawal, payment, and deposit protocol. For example, in the withdrawal protocol, the network can crash while a user is waiting for receiving the signature of e-cash. In this case, the e-cash that the user was supposed to receive would be lost because the user didn't receive the e-cash although the bank had already withdrawn the corresponding money. They solved such problems via withdrawal atomicity, payment-delivery atomicity, and deposit atomicity in fair off-line e-cash schemes [8, 14] by using *recovery, resolve, dispute handling protocols*, and *refund function*.

2.4 Classification of Micro-payment

Existing payment systems may be divided into two groups: electronic cash and credit-debit systems. Also, according to the type of information that is exchanged, it is distinguished between account-based and token-based. And, according to authorization method, On-line and off-line is used. Fig. 2.2 shows a classification of micro-payment systems.

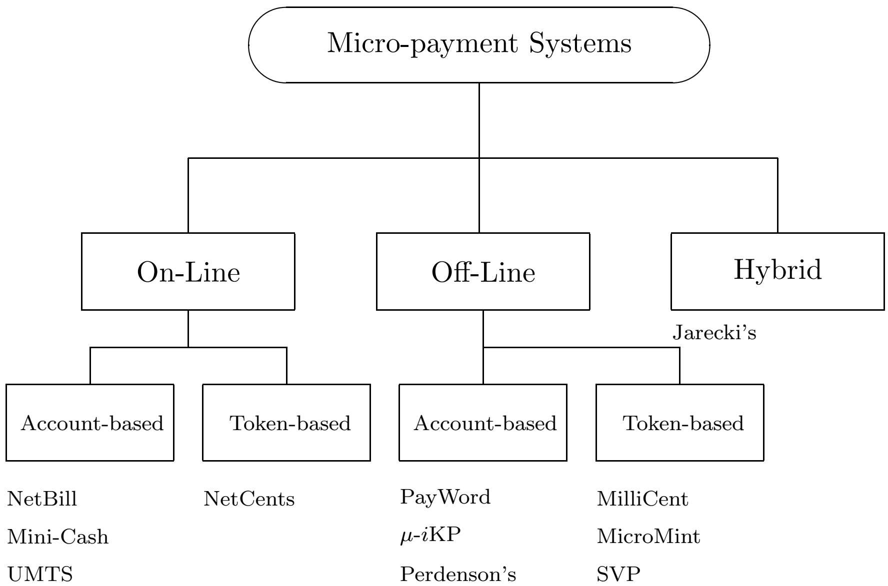

Figure 2.2: Classification of Micro-Payments

Electronic cash resembles conventional cash, when entities exchange e-cash that represent value, it determines the value of paper money. The credit-debit means that money is represented by numbers in bank accounts which are transferred between entities over networks. With an on-line protocol, a central payment authority is contacted to authorize each transaction. In an off-line protocol, the vendor verifies the payment using cryptographic techniques, and commits the payment to the payment authority later.

2.5 Previous Protocols

Table 2.1 compares the various previous micro-payments.

Table 2.1: Comparison of Micro-Payments

	MicroMint	MilliCent	PayWord	Mini-Cash	UMTS	SVP	NetBill
Security Protocols	Hashing	Symmetric Key, Hashing	Public Key, Hashing	Public Key, One-way function	Public Key, One-way function	MAC	Public Key, Kerberos
Credt/ Debit	Credit to V Debit to C and B	Debit	Credit	Credit	Credit	Debit	Credit
Authori-zation	Offline	Offline	Offline	Online	Online	Offline	Online
Denomi-nation	Coins	Scrip	Chains of PayWord	Account	Chains of coins	Coins (Token)	Account
Storage Req. of Coins	Coins	Scrip, Shared Key	PayWord Chain, Commitment, Indices of each chain	Two counters	Certificate, Coin chain, Indices of each chain	Shared Key, customer	Counter
Anony-mity	No	No	No	No	No	No	No
Forgery Prev-ention	By policy	Yes	Yes	Yes	Yes	Yes	Yes
Double Spending Detection	Yes	Yes	Yes	Yes	Yes	Yes	Yes

2.5.1 MicroMint

Rivest and Shamir suggested this scheme in [24] with PayWord which is another maro-payment scheme.

The basic idea is that coins are represented by hash function collision a $k - waycollision$. Thus, there is no use of public key computations. The $k - waycollision$ is a set of k distinct x-values, $x_1, \ldots, x_k$ that have the same hash value y :

$$h(x_1) = h(x_2) = \ldots = h(x_k) = y$$

.

This scheme is debit based between customer and broker, and credit based from the vendor's viewpoint because the vendor does not get the payment until the broker accepts the coins.

Brokers issue new coins at the beginning of each month and sells them to customers. When a customer purchases goods, he gives a coin $(x_1, \ldots, x_k)$ to a vendor. The vendor verifies that it is indeed a good $k - waycollision$ by computing $h(x_i)$ and checks that the values are equal to the price of goods. The vendor returns the coins to broker at the end of each day. The broker checks that the coins have been returned previously, if it is not then he pays the vendor. Unused coins are returned to the broker at the end of month.

There is no detail purchasing process in the original paper.

MicroMint supports both customer-specific and vendor-specific coins.

MicroMint does not use public key computations. It needs only symmetric key computations. Also, it is off-line based scheme. But, generating coins might be a waste of resources than other protocols. MicroMint does not fully prevent from the forgery because it is based on an assumed that forgers have no interest in cheating to gain a negligible amount.

2.5.2 MilliCent

Mannasse proposed this scheme in [17].

MilliCent uses *scrip* which is sold by broker. By verifying the scrip locally at vendor site, communication cost is reduced.

This scheme is debit based to customers. Customers enter into long term relationship with brokers. Also authorization for payments is performed on off-line.

MilliCent uses shared key and scrips as follows:

- *customer_secret*: The customer uses this to prove legitimate ownership of the scrip. The customer receives this upon initial receipt of the scrip. The vendor (or its broker) constructs the $customer\_secret$ with a hash using the customer identifier, $Cust\_ID$ and $master\_customer\_secret$.

$$customer\_secret = h(Cust\_ID, master\_customer\_secret)$$

- *master_customer_secret*: The vendor uses this to derive the $customer\_secret$ from data in the scrip. This is known only to the vendor and to the broker.

- *master_scrip_secret*: The vendor uses this to prevent tampering and counterfeiting.

The vendor and the broker define the contents of the scrip as shown in table 2.2.

Table 2.2: Fields in the vendor scrip

Vendor	Value	ID	Cust_ID	Expires	Props	Cerrificate

- Vendor : vendor identifier
- Value : balance value of the scrip
- ID : a unique serial number for the scrip
- Cust_ID : customer identifier
- Expires : the date on which the scrip becomes invalid
- Props : extra data
- Certificate : proof of the integrity of the scrip
  $= h(Vendor||Value||ID||Cust\_ID||Expires||Props||master_scrip_secret)$

Brokers buy and sell vendor scrip as a service to customers and vendors. Broker scrip serves as a common currency for customers to use when buying vendor scrip. A piece of scrip represents an account the customer has established with a vendor. The balance of the account is kept as the value of the scrip. When the customer makes a purchase with scrip, the cost of the purchase is deducted from the scrip's value and new scrip is returned as change.

There are three types of protocols according to type of scrip such as scrip in the clear, private and secure, and secure without encryption.

2.5.3 PayWord

We will describe this scheme in Chapter 3.

2.5.4 Mini-cash

Jakobsson proposed this scheme in [13].

Mini-cash considers reducing storage costs of customers and vendors, protecting against the bank robbery attack, and implementing as a smart card payment scheme. This scheme uses disposable anonymous account concept: when a coin is spent, the corresponding account is removed, and a new account is created corresponding to a new coin.

This scheme let the broker store information corresponding to funds, leaving the users with only having to store a secret key needed to access the data stored by the broker. In this manner not only eases the load on the user devices, but also allows for the much stronger attack model. This scheme elaborates the method to hold only a constant sized secret seed in the payment devices. The scheme maintains low storage costs for user devices by letting these store only a seed and counters.

This scheme is on-line based model and is needed public key computations to the customer including signature.

2.5.5 UMTS (Universal Mobile Telecommunications System)

Horn and Preneel proposed this scheme in [10].

They considers an efficient public-key protocol for mutual authentication and key exchange for third generation mobile communication system. And, they integrates micro-payment scheme into the authentication protocol.

This scheme is based on PKI to the authentication. And the micro-payment scheme is similar to PayWord.

2.5.6 SVP

Stern and Vaudenay suggested this scheme in [29].

SVP uses MAC and hash function instead of using public key cryptography and adopted tamper resistant devices for both customer and merchant sides.

There are some assumptions as follows:

- The vendors have been given a tamper proof device for validating micro-payments.
- The communications both between the broker and the vendor's device and between the broker and the customer are secure.
- The device has an internal permanent memory and an external memory.

Broker creates secret key, and distributes it in a secure way to the device of each vendor at initialization step.

Broker generates a *token* and spending key, and authorizes the customer to spend a given amount of money with the spending key. When the customer purchases a goods, he introduces himself as having a *token* to vendor. The customer sends it with a random number and an amount value to the vendor. Then the vendor send the identification string of his device and a random number to the customer. The customer then reveals a *purchase* which is made by MAC to the vendor. Then the vendor verifies it by using shared secret key which was transferred at the initialization step.

SVP security is based on strengthening of the shared secret key

2.5.7 NetBill

Tygar describes NetBill in [30]. The scheme uses NetBill server which plays the role of a certification authority in addition to its function as a payment intermediary. It also manages the distribution of keys that are used to encrypt the exchanges between the customer and the vendor.

Customer requests a price from vendor for some goods. Then the vendor makes an offer to the customer. If the customer accepts the offer, then the vendor sends encrypted goods. The customer makes an EPO (Electronic Purchase Order) and sends signed EPO to the vendor.

The vendor countersigns the EPO also signs goods encryption key. The vendor sends both values to the NetBill server. The NetBill server checks the customer's account and the time-out value. After checking all values, if there is no problem, then the NetBill server transfers price funds from the customer's account to the vendor's account. It also stores goods encryption key. It prepares a signed receipt and sends it to the vendor. The vendor stores the receipt and transfers it to the customer.

III. PayWord

In this chapter, we describe PayWord which is one of the most popular micro-payment systems. This system is a credit-based, based on chains of "PayWords "(hash chain). This system employs the cryptographic properties of digital signature and hash chain which is consisted of collision free hash functions. The application of this idea for micro-payments has also been adopted in [10, 11, 22, 24]. This mechanism was originally proposed for use in an authentication scheme by Lamport [16].

The collision free hash function is defined as follows in [12].

Definition 1 *A family of collision free gash functions is a set of hash functions with the following properties :*

- *There is a probabilistic polynomial time algorithm, which on input a value of security parameter selects uniformly and randomly a member of the family with the given value attached.*
- *All functions in the family are computable in polynimial time.*
- *the problem of finding $x \neq y$ such that $h(x) = h(y)$ for a given h in the family is computationally infeasible to solve*

A member of a collision free family is also called collision free. If h is collision free, then h is also one-way.

In this thesis, we use the following common notations to describe protocols and some notations may be slightly different from those in the original protocol.

- C, B, V : Customer, Broker, Vendor
- Z : entity $Z \in \{C, B, V\}$
- Z_{ID} : Z's identity
- PK_Z, SK_Z : Z's public key, private key
- $\{Data\}_{PK_z}$, $\{Data\}_{SK_z}$: public key encryption scheme with Z's public key and decryption scheme with Z's private key
- $h()$: collision-free hash function
- C_Z : Z's certification made from broker
- E : expiration period of certificate

Principal entities consist of customer, broker and vendor. Customers open an account with a broker. The broker issues a digitally-signed certificate, which authorizes the customer to make PayWord chain and assures vendors that the customer's PayWords are redeemable. Customer creates the PayWord chain in reverse order by picking the last Payword w_n at random, and then

$$w_{i-1} = h(w_i),$$

where h is a collision free hash function, and $i = 1, \ldots, n$. Let denote w_0 be the root of the PayWord chain. Customer computes and signs a "commitment"at the first request to vendor that day. The commitment contains the root w_0, but not any PayWord w_i. Then customer provides this commitment and her certificate to vendor. The vendor can verify their signatures.

The i-th payment from customer to vendor consists of the pair (w_i, i), which the vendor can verify using w_{i-1}.

At the end of day, vendor shows to broker the last payment (w_l, l) received from each customer that day together with each corresponding commitment. Then the broker deposits corresponding money to the index in vendor's account.

PayWord is an off-line system: vendor does not need to interact with broker when customer contacts vendor. Also, vendor does not need to interact with broker as each payment is made.

3.1 Issuing Customer Certificate

C establishes an account with B. C provides personal details to B over a secure authenticated channel, such as a credit card number, delivery address, and her public key. The C's certificate has an expiration date. The customer's certificate has to be renewed by the broker regularly; the broker will do so only if the customer's account is in good standing.

The certificate is defined by

$$C_c = \{B_{ID}, C_{ID}, A_C, PK_C, E, I_C\}_{SK_B}$$

The broker includes other information I_C in the certificate such as:

- a certificate serial number,
- credit limits to be applied per vendor,
- information on how to contact the broker,
- broker/vendor terms and condition.

3.2 Protocol Steps

After issuing customer's certificate from the broker, there are three steps in PayWord protocol such as commitment, payment and deposit.

Fig. 3.1 shows overview of PayWord steps.

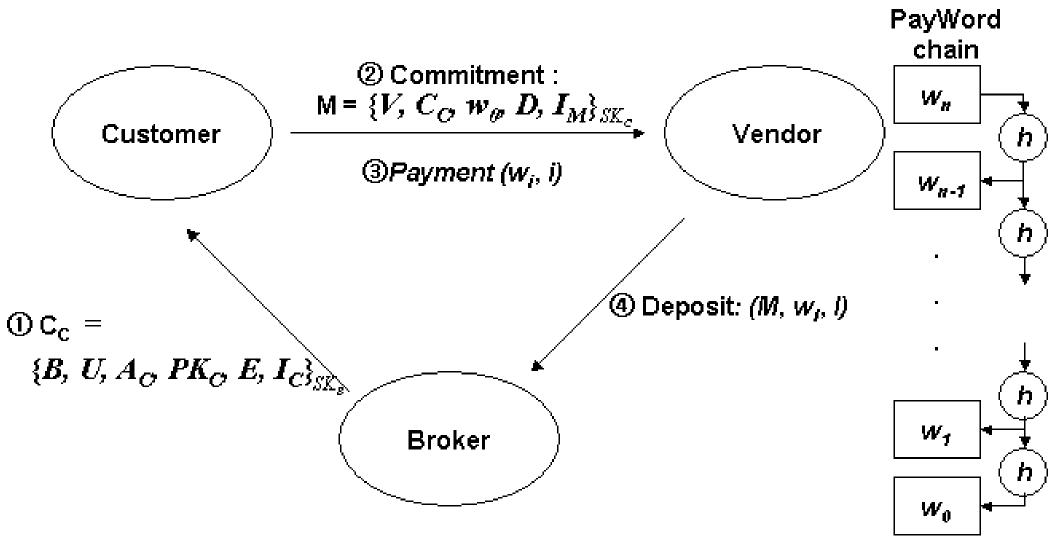

Figure 3.1: PayWord protocols

3.2.1 Commitment

C's certificate authorizes the C to make PayWord chains, and assures V that the C's PayWords are redeemable by B.

When C contacts a new vendor V, she computes a new PayWord chain $w_1, ..., w_n$ with root w_0. Here n is chosen at the C's convenience.

Then, she computes her commitment for that chain:

$$M = \{V, C_C, w_0, D, I_M\}_{SK_C},$$

where V identifies the vendor, C_C is C's certificate, w_0 is the root of the chain, D is the current date, and I_M is any additional information. M is signed by C.

M authorizes B to pay V for any of the PayWord chain which means V can redeem with B before date D. Also M includes both identities of C and V, and so is both customer-specific and vendor-specific.

In advance of business, C commits M to V. Then V verifies C's signature on M and B's signature on C_C, and checks their expiration date. If valid, V stores it.

3.2.2 Payments

When C purchases goods from V, C transfers a payment P to V as its payment. P consists of a PayWord and its index:

$$P = (w_i, i)$$

V computes $h(w_i)$ and confirms it, if valid then keeps it. C spends her PayWord in order: w_1 first, then w_2, and so on. Once C spends w_i, she can not spend w_j for $j < i$.

This payment does not specify what goods it is payment for. So, if V does not send nothing, or the wrong goods then C will face the risk of losing the money.

3.2.3 Deposit

At the end of each day, V sends the last payment information (w_l, l) received from C and its commitment M to B and requests redemption. B needs to verify each commitment M received, and each payment (w_l, l). After verifying, B withdraws eligible money from C's account and deposits it in B's account.

3.3 Problems of PayWord System

PayWord has several problems as follows:

- A customer should keep a different hash chain according to each vendor. Hence, in the customer's side, she must compute expensive public key operations as many as the number of vendors that she wants to transact.

- A customer should preserve all of last indices that she used to deal with vendors because of using a new hash chain for each vendor.
- This system doesn't present the complete steps of business to sale goods.
- When a customer sends the index with respect to goods' price to a vendor, the vendor issues no receipt on it. Due to absence of verification of receiving, it occurs the loss of money.
- A customer can make payment exceeding her credit limit.

In order to address these problems, we propose a new payment protocol in which we make use of only one hash chain with respect to a number of transactions. The new protocol assures atomicity as well.

IV. Proposed System

In this chapter, we describe a new proposed micro-payment system. Before describe details of it, we describe some initial assumptions, configuration of proposed protocol.

Explicitly, this system can be considered as an improvement and extension of PayWord [24] suggested by Rivest and Shamir. In particular, we focus on the scheme that guarantees atomicity and non-repudiation among requirements for payment systems.

Finally, when a customer uses the micro-payment over several vendors,we analyze not only the efficiency in terms of the storage size but also the security issues.

4.1 Initial Assumptions

Initial assumption for the proposed system is as follows:

- There is a trustful PKI (Public Key Infrastructure).
- Customers and vendors completely trust the broker.
- A new proposed system deals with digital goods.

4.2 Configuration of Protocol

The proposed system is composed of four phases: initialization, negotiation, payment-delivery, and deposit phase. Fig. 4.1 shows an overall configuration of each phase.

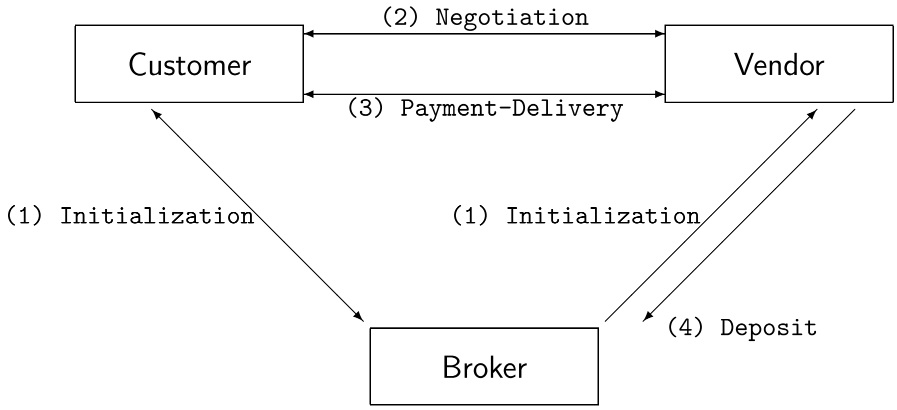

Figure 4.1: Overall configuration

4.3 Details of Protocol

4.3.1 Initialization Phase

Fig. 4.2 shows messages transferring between C and B at the initialization phase.

Message (1):

- C sends an encrypted message that contains the root value of hash chain w_0, the length of chain n, customer identity C_{ID}, and broker identity B_{ID}. This message is encrypted by B's public key.

Message (2):

- B decrypts the received message by his private key and checks whether the generation of payment is available in the customer's

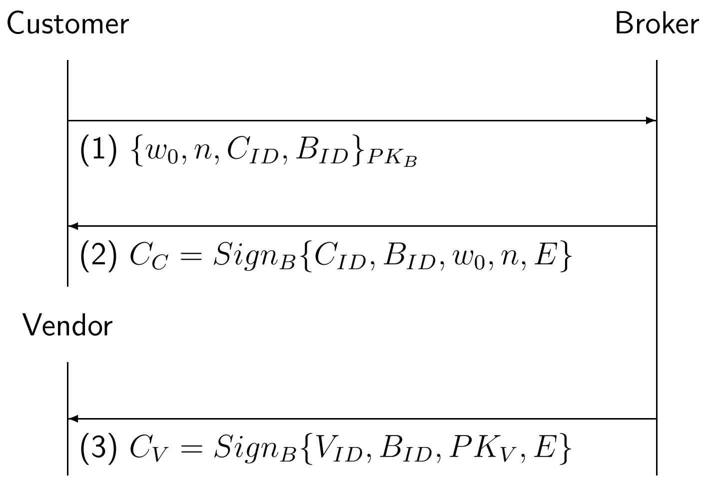

Figure 4.2: Initialization phase

account standing or not. If the length of hash chain does not exceed the limit, B issues a certificate C_C to C.

Message (3):

- Also, B issues a certificate C_V to V. V receives the certificate, and uses it in transaction with C.

4.3.2 Negotiation Phase

Fig. 4.3 shows messages transferring between C and V at the negotiation phase.

Message (4):

- C and V participate this phase to agree on the price, goods, and predetermined time limit. C searches information of prod-

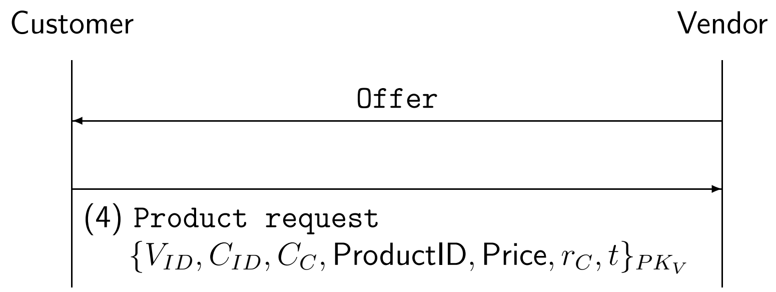

Figure 4.3: Negotiation phase

uct which she wants to purchase through Internet. If C wants to purchase a product, then she sends a product request message.

- ProductID is a product identity which V deals with. It is a signed value by V, $Sign_V\{h(\mathsf{goods})\}$. C uses of it by downloading from V's web site and identifies the product. Also C can use it when she claims that she receives goods different from what she ordered.
- The random number r_C identifies the negotiation.
- The value t is the predetermined time limit from receiving coin of V to deposit it in his own account.
- V verifies the expiration period of C's certificate, root value of hash chain, and the length of hash chain from C's certificate.

4.3.3 Payment-Delivery Phase

Fig. 4.4 shows messages transferring between C and V at the payment-delivery phase.

Message (5):

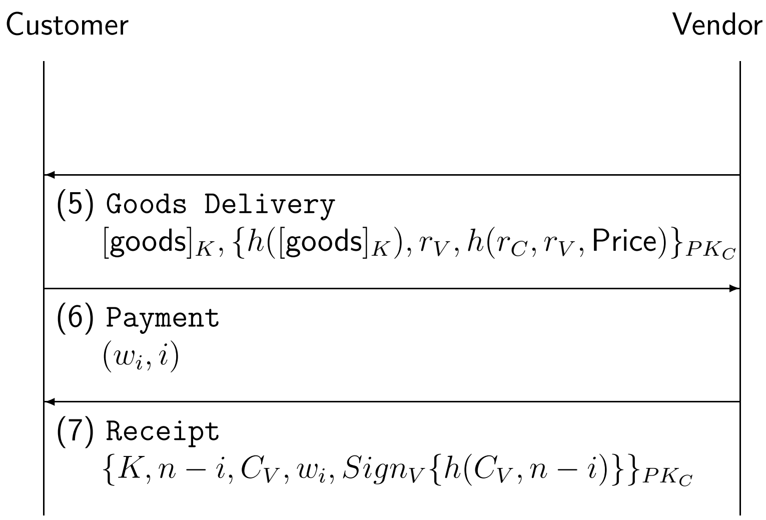

Figure 4.4: Payment-Delivery phase

- After negotiation of goods, V encrypts digital goods by symmetric key and delivers it to C.
- V transfers the product's hash value, a random number generated by himself and the certificate from broker.
- When C receives encrypted goods, she checks if it is the same goods as she previously requested by computing $h(r_C, r_V, \mathsf{Price})$, then identifies the price of goods.

Message (6):

- C pays a designated amount. Payment is made by transferring hash chain value and its index.

Message (7):

- V verifies the length of hash chain and makes sure if it exceeds the limit. V verifies the transferred hash value using the root value.
- After verification of payment, V returns receipt and symmetric decryption key, and K of sold goods. Also, V includes a signature for remaining index length and his certificate. When C uses the same hash chain to another vendor, it will be sent to the next V to verify the previous used index.
- C decrypts purchased goods using decryption key from V and identifies its contents.

4.3.4 Deposit Phase

Fig. 4.5 shows message transferring between V and B at the Deposit phase.

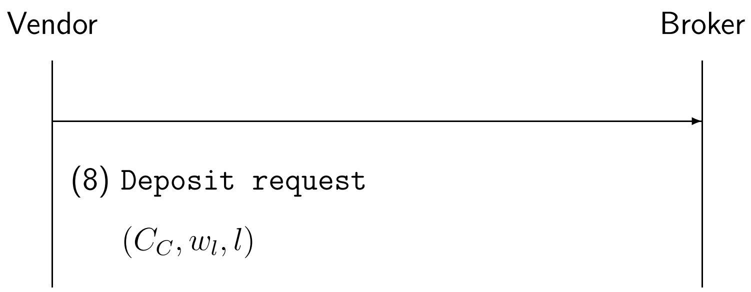

Figure 4.5: Deposit phase

Message (8):

- V sends the latest payment and hash chain index with the certificate of C to B. V should request within the predetermined specific time limit during negotiation phase.

- B verifies C's certificate and confirms the limit length of hash chain. Using the root value, B verifies hash chain value. If the value is valid, B deposits the corresponding amount of money in V's account.

4.4 Multiple Shopping

Assume that C has a hash chain of length n and uses index, from 1 to i to $(k-1)$th V and index, from i to j to kth V as shown in Fig. 4.6

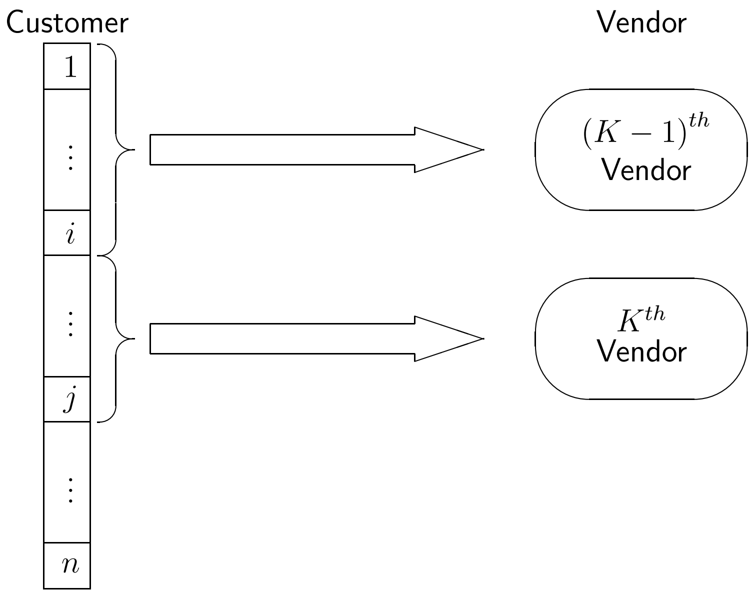

Figure 4.6: Payment k-th vendors

PayWord should issues a new hash chain whenever a customer carries out business with different vendors. Also a customer should preserve all of last indices that she used to deal with different vendors. On the other hand, in the new protocol, customer holds only a unique hash chain and does business with a number of vendors using the same hash chain. The customer also sends the same certificate as one from the previous vendor to the current vendor.

4.4.1 Payment-Delivery Phase

Fig. 4.7 shows messages transferring between C and V at the payment-delivery phase with k vendors.

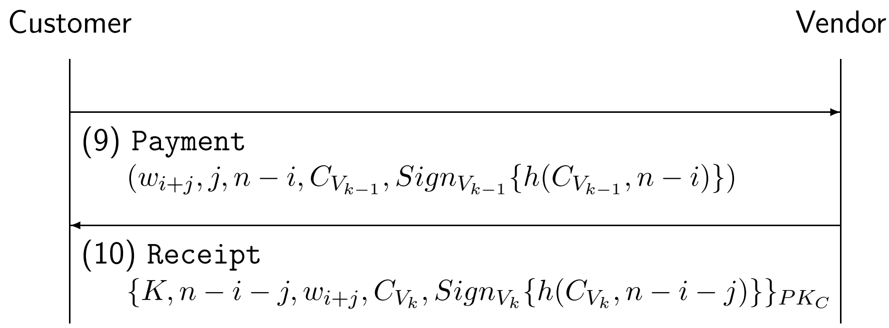

Figure 4.7: Payment-Delivery phase with k-th vendors

Message (9):

- In the case that C pays to kth V, she provides the certificate from $(k-1)$th V and indices i and j together to kth V.

Message (10):

- V verifies the certificate of $(k-1)$th vendor and transfers his certificate and index verification value along with the decryption key of sold goods.

4.4.2 Deposit Phase

Fig. 4.8 shows message transferring between C and V at the deposit phase after dealing with k-th vendors.

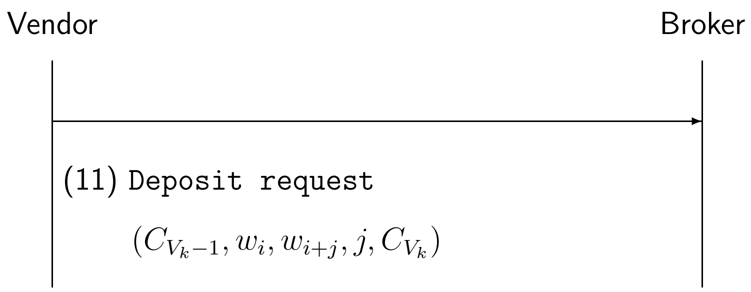

Figure 4.8: Deposit phase after dealing with k-th vendor

Message (11):

- When V requests redemption, V sends the certificate of $(k-1)$th V, $C_{V_{k-1}}$, hash chain value, w_i, his own hash chain value w_{i+j}, the index of redemption request, j and certificate, C_{V_k} to B.
- By the verification of each hash chain value, B can make a decision on redemption to the requested index. If the requested index ranges within the predetermined limit, the redemption will be done. In a similar way, B can check the excess of the limit.

4.5 Analysis of Proposed System

4.5.1 Atomicity

Money Atomicity

Money atomicity is preserved by;

- Only after a customer pays for goods, the customer can receive the key used for the decryption for goods, and the receipt. Therefore, loss of the money can be prevented.

- In the negotiation phase between a customer and a vendor, the predetermined time limit is established over money from the beginning of transaction to the deposit. Therefore, the available period of money can be established.

- Since a broker certifies information on the length of money, a customer can create money at her own discretion.

Goods Atomicity

Goods atomicity is guaranteed only when money atomicity is guaranteed for a customer who receives the goods that has been already paid.

In the case of the proposed system, if a customer doesn't pay for ordered goods, she can't obtain the key for the decryption. Therefore, she can't get the decrypted goods.

On the other hand, although a customer has already completed payment for goods, a customer cannot get the key for the decryption until the predetermine time limit is over. Then, the customer processes as following ways:

- The customer sends the index value (w_i, i) of payment to the broker, the random number r_C used for the transaction, certification C_V of the vendor, the value of predetermined time limit t to the broker, and requests for the confirmation of whether the deposition into the vendor's account is completed.

- If the deposition is completed, the broker sends the index value, and the random number to the vendor in order to request for the customer to send the value of receipt for the transaction. If the vendor refuses, the broker terminates the account with the vendor, registers a blacklist, and considers legal action.

- If the deposition has not yet completed, notify the customer that the transaction with the vendor was cancelled.

Goods atomicity follows from the above procedure.

Certified Delivery

Now, suppose that a customer claims that she receives goods different from what she ordered. In this case the following steps will be taken to guarantee certified delivery atomicity.

- The customer requests the validation of goods to a broker by sending $\mathsf{P}roductID$ value, an encrypted product value, a decrypted key K, and the good's encrypted value signed by a vendor.
- The vendor can't deny the transaction because he signed the product's hash value with his private key. However, in reality, the vendor will not commit such a misconduct because of the little profit from cheating is the risk of losing his credit in the case that the misconduct is revealed, and the risk of the termination of his account from the broker. On the other hand, if it turns out that the customer received the right product, it tells that it was the customer's misconduct.

4.5.2 Security Issues

Double Spending Detection

Our system makes use of a unique hash chain to each vendor so that it has the following advantages:

- When a customer uses an identical indices to a different vendor once more, a broker is able to detect double spending since he verifies the customer's certificate transferred from the vendor to be paid redemption. For the purpose of preserving this property, the broker should keep both the root value and the length information in the certificate that is issued to the customer.

- In case that the previously used index range is reused in a different vendor, for instance, a customer used index range from i to j to kth vendor, and he paid from i to j to arbitrary $(k+1)$th vendor once more. As the same way above, the broker can detect two indices which were spent in k and $(k+1)$th vendors identically.

Broker is able to keep blacklist of the double spenders. The broker also has a flexibility on a term of "validity" according to the length of hash chain and the credit of customer when she issues a certificate.

Overspending Prevention

The length of hash chain is sent to a broker and included in the certificate so that a vendor can verify the excess of the limit by identifying a customer's certificate.

Forgery Preventing

An unused hash chain value cannot be inferred from a used one. If malicious vendors conspired each other, they could know unused hash values. But, the right redemption will not occur since a broker can verify double spending when it requested.

4.5.3 Efficiency

Table 4.1 compares the efficiency between PayWord and our proposed system, where m denotes the number of vendors and w_m denotes the hash chain itself, respectively.

Table 4.1: Comparison of efficiency

System	PayWord	Proposed System
Public key computation	1	3
Indices to be saved with m vendors	$m \times w_m$, $m \times index$	$1 \times w_m$, $1 \times receipt$

In PayWord, it doesn't provide a complete protocol for electronic commerce so that the loss of money happens and atomicity is not assured. On the contrary, our proposed system presents a complete protocol to deal with digital goods. This expansion makes transaction steps and the length of message increase slightly. But, in customers' side, they utilize only one unique hash chain to a number of vendors. Thus, it is not necessary for customers to select a new hash chain to do a business with a different vendor. Furthermore, since the remaining hash index can be used with the signature of previous vendor, all what customers do is to handle the index of one hash chain.

Table 4.2 compares the several properties between PayWord and our proposed system.

Table 4.2: Comparison of properties

Property	PayWord	Proposed System
Atomicity		$\surd$
Double spending detecting	$\surd$	$\surd$
Forgery preventing	$\surd$	$\surd$
Overspending prevention		$\surd$
Non-repudiation		$\surd$
Multiple shopping		$\surd$

V. Conclusion and Further Works

In this thesis, we propose a new micro-payment system that satisfies basic requirements of micro-payment systems and atomicity. Explicitly, this system can be considered as an improvement and extension of PayWord which is one of the most popular micro-payment systems [24] suggested by Rivest and Shamir.

Our proposed system makes use of only one hash chain but is applicable to multiple vendors so that it provides efficiency and convenience of index management. Since PayWord requires to use each hash chain for each vendor, the customer has to maintain all indices after dealing with vendors. To solve this problem, we use a receipt which includes a signature of vendor. The customer uses this signature for different vendors to enable them to verify whether the starting index of payment is correct or not. Moreover, we focus on the system that guarantees atomicity and non-repudiation among requirements for payment systems. Atomicity and non-repudiation play an important role in payment systems. We presents a complete protocol to transact digital goods, and at the same time, preserves atomicity.

Finally, we analyze not only the efficiency in terms of the storage size but also the security issues in case that a customer uses the micro-payment to several vendors.

However, the frequency of public key computation gets so increased that it compromises computational efficiency that PayWord holds.

As further works, we need to design micro-payment scheme that is available in mobile communication environment which requires more efficiency of computation and data store and availability, because micro-payment systems are considered to be proper for areas where efficiency is the most critical factor.

Also, we need to implement this system in the future, and analyze the efficiency of message transactions after implementing this system. Furthermore, we need to find out the optimal business cases to apply this system in the real situation.

Finally, we need to consider the basic requirements of micro-payment systems more carefully.

다중사용 가능한 소액지불 시스템에 관한 연구

이 만호

안전한 전자지불 시스템의 연구는 인터넷 및 전자상거래의 확산과 함께 다양한 기법과 시스템들이 제안되고 있다.

소액지불 시스템 (Micro-payment)은 기존의 전자지불 시스템 (Macro-payment)과는 달리 인터넷에서의 데이타베이스의 조회, 소프트웨어, 전자신문 등의 소액의 디지털 상품을 거래하기 위한 수단으로 제안되었다.

소액지불 시스템은 연산의 효율성 및 저장비용의 효율성등을 요구한다. 따라서 기존에 제안된 소액지불 시스템들은 공개키 연산의 최소화 또는 사용 배제, 연산량의 최소화을 고려하였다.

안전성 요소로는 전자지불 시스템에서 기본적으로 요구되는 이중사용방지 또는 검출, 위조 불가능성 등이 요구된다. 또한 전자상거래서 요구되는 원자성의 보장은 사용자가 전자지불 프로토콜의 진행중 네트워크의 오류 등으로 발생하는 화폐의 분실 등의 문제점을 해결하기 위해 요구된다.

이 논문에서는 소액지불 시스템에서 요구되는 특성을 만족하고 원자성을 보장하는 소액지불 시스템을 제안한다.

제안된 시스템은 소액지불 시스템중 PayWord [24] 를 기반으로 설계하였다. PayWord 시스템에서는 사용자가 각 상점에 대하여 각 각의 해쉬체인을 생성하고 보관해야 하며, 사용 금액에 대한 인덱스를 각 상점별로 보관해야 한다.

이러한 문제점을 해결하기 위하여 제안된 시스템에서는 사용자는 하나의 해쉬체인 만을 사용자가 사용한다. 사용자는 사용한 금액에 대한 인덱스에 대한 서명값과 상점의 인증서를 거스름으로 받는다. 사

용자는 다음 상점의 거래에 대한 지불시 이전 상점에서 사용한 인덱스 이후의 값과 이전 상점의 서명값 및 인증서를 사용한다. 이러한 방식으로 하나의 해쉬체인으로 다중 상점에 사용할 수 있다. 또한, 제안된 시스템은 PayWord에서 제공하지 못하는 원자성을 보장하도록 설계하였으며, 상품거래에 대한 부인방지 또한 제공한다.

제안된 시스템과 함께 기존의 PayWord시스템으로 다중 상점과의 거래시의 효율성 및 저장요소를 비교하였고, 제공하고 있는 보안 요구사항 특성들을 분석하였다.

References

1. M. Bellare, J. Garay, C. Jutla, and M. Yung, "VarietyCash: A Multi-Purpose Electronic Payment System", *Proc. of the 3rd USENIX Workshop on Electronic Commerce*, pp.9-24, 1998.

2. S. Brands, "Untraceable Off-line Cash in Wallets with Observers", *Advances in Cryptology-Proc. of CRYPTO'93*, LNCS 773, pp.302-318, 1994.

3. E. Brickell, P. Gemmell and D. Kravitz, "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Exchange", *Proc. of 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp.457-466, 1995.

4. J. Camenisch, U. Maurer and M. Stadler, "Digital Payment Systems with Passive Anonymity-Revoking Trustees", *Proc. of ESORICS'96*, LNCS 1146, pp.31-43, 1996.

5. J. Camenisch, J. M. Piveteau and M.Stadler, "An Efficient Fair Payment System", *Proc. of 3rd ACM Conference on Computer and Communications Security*, ACM Press, pp.88-94, 1996.

6. J. Camp, M. Harkavy, J. D. Tygar, and B. Yee, "Anonymous Atomic Transactions", *Proc. of the 2nd USENIX Workshop on Electronic Commerce*, pp.123-133, 1996.

7. D. Chaum, "Blind Signatures for Untraceable Payments", *Advances in Cryptology-Proc. of CRYPTO'82*, Plenum Press, pp.199-203, 1983.

8. Y. Frankel, Y. Tsiounis and M. Yung, "Indrect Discourse Proofs": Achieving Efficient Fair Off-Line E-Cash", *Advances in Cryptology-Proc. of ASIACRYPT'96*, LNCS 1163, pp.286-300, 1996.

9. E. Fujisaki and T. Okamoto, "Practical Escrow Cash System", *Proc. of 1996 Cambridge Workshop on Security Protocols*, LNCS 1189, pp.33-48, 1997.

10. G. Horn, B. Preneel, "Authentication and Payment in Future Mobile Systems", *Computer Security ESORICS 98*, LNCS 1485, pp.277-293, 1998.

11. R. Hauser, M. Steiner, and M. Waidner, "Micro-Payments based on iKP", *Proc. of the 14th Worldwide Congress on Computer and Communications Security Protection*, pp. 67-82, 1996.

12. I. B. Damgard, "Collosion Free Hash Functions and Public Key Signature Schemes", *Advances in Cryptology-Proc. of EUROCRYPT '87*, LNCS 304, pp.203-216, 1988.

13. M. Jakobsson, "Mini-Cash : A Minimalistic Approach to E-Commerce", *Proc. of PKC1999*, LNCS 1560, pp.122-135, 1999.

14. M. Jakobsson and M. Yung, "Revokable and Versatile E-money", *Proc. of 3rd annual ACM Conference on Computer and Communication Security*, pp.76-87, 1996.

15. S. Jarecki and A. Odlyzko, "An Efficient Micropayment System Based on Probabilistic Polling", *Financial Crypto 97*, LNCS 1318, pp.173-192, 1997.

16. L. Lamport, "Password Authentication with Insecure Communication", *Communication of the ACM*, Vol.24, No.11, pp.770-772, 1981.

17. M. S. Manasse, “The Millicent Protocol for Electronic Commerce”, *Proc. of the 1st USENIX Workshop on Electronic Commerce*, 1995. http://www.millicent.org/works/details/papers/millicent-w3c4/millicent.html

18. D. M’Raïhi, “Cost Effective Payment Schemes with Privacy Regulations”, *In Advances in Cryptology-Proc. of ASIACRYPT’96*, LNCS 1163, pp.266-275, 1996.

19. T. Okamoto and K. Ohta, “Universal Electronic Cash”, *In Advances in Cryptology-Proc. of CRYPTO’91*, LNCS 576, pp.324-337, 1991.

20. T. Okamoto, “An Efficient Divisible Electronic Cash Scheme”, *In Advances in Cryptology-Proc. of CRYPTO’95*, LNCS 963, pp.438-451, 1995.

21. H. Petersen and G. Poupard, “Efficient Scalable Fair Cash with Off-line Extortion Prevention”, (Technical Report, ENS, 33 pages, 1997), short version *Proc. of Int. Conference on Information and Communications Security (ICICS’97)*, LNCS 1334, pp.463-477, 1997.

22. T. P. Pedersen, “Electronic Payments of Small Amounts”, *Proc. of Cambridge Workshop on Security Protocols*, LNCS 1189, pp.59-68, 1996.

23. T. Poutanen, H. Hinton, and M. Stumm, “NetCents: A Lightweight Protocol for Secure Micropayments“, *Proc. of the 3rd USENIX Workshop on Electronic Commerce*. pp. 25-36, 1998.

24. R. L. Rivest and A. Shamir, “PayWord and MicroMint: Two Simple Micropayment Schemes”, *CryptoBytes*, pp.7-11, 1996.

25. M. Satyanarayanan, "Fundamental Challenges in Mobile Computing", *Proc. of the 15th ACM Symp. on Principle of Distributed Computing*, pp.1-7, 1996.

26. M. Stadler, J. M. Piveteau and J. Camenisch, "Fair-Blind Signatures", *Advances in Cryptology-Proc. of EUROCRYPT'95*, LNCS 921, pp.209-219, 1995.

27. S. von Solms and D. Naccache, "On Blind Signatures and Perfect Crimes", *Computers and Security*, pp.581-583, 1992.

28. M. Sirbu and J. D. Tygar, "NetBill: An Internet Commerce System Optimized for Networked Delivered Services", *IEEE COMPCON'95*, pp.20-25, 1995.

29. J. Stern, S. Vaudenay, "SVP : a Flexible Micropayment Scheme", *Financial Crypto 97*, LNCS 1318, pp.161-172, 1997.

30. J. D. Tygar, "Atomicity in Electronic Commerce", *Proc. of the 15th ACM Symp. on Principle of Distributed Computing*, pp.8-26, Philadelphia, 1996.

31. S. Xu, M. Yung, G. Zhang, and H. Zhu, "Money Conservation via Atomicity in Fair Off-Line E-Cash", *Proc. of the 2nd Int. Information Security Workshop*, LNCS 1729, pp. 14-31, 1999.

Acknowledgement

I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor. I am so much thankful for his constant inspiration which leads me to carry out my research. Special thanks goes to Prof. Younghee Kim and Prof. Chulsoo Lee for their generosity and agreeing to serve as advisory committee members of my thesis.

I would like to thank all CAIS Lab. members; Byoungcheon Lee, Jinho Kim, Jaegwan Park, Myungsun Kim, Wooseok Ham, Jongseong Kim, Hyunrok Lee, Hyungki Choi, Kyusuk Han, Vo Duc Liem, and Yan Xie for giving me good comments through the course of my research and great help in my school life in ICU.

And I would like to give my thanks Saehoon Kang of CN Lab., Jungyeun Lee , Jin Kim, and Sangwon Lee of C Lab., and Sebon Ku of my dormitory roommate for their help and kindness during here in ICU.

I am also grateful to former C&IS Lab. members; Moonseog Seo and Gookwhan Ahn of SECUi.COM, Jaeseung Go of Hanaro Telecomm Inc., Heesun Kim of ETRI, Boyeon Song of KISA, Hyuncheol Park of Ahn Lab., and Joonsang Baek for their encouragement and advice.

Specially, I thank to Minje Kim for helping to come here ICU.

I think everything is due to God's grace.

I would like to extend my thanks really to my mother for her devotion. I would also thank my family, two elder brothers and their wives. Their consistent love and belief encouraged me to overcome many troubles.

Finally, I will never forget my life in ICU. It filled up my poor knowledge and made me a grown-up person.

Curriculum Vitae

Name : Manho Lee

Date of Birth : December. 1. 1968

Sex : Male

Nationality : Korean

Education

1986.3–1990.2	Computer Engineering Chungbuk National University, Korea (B.S.)
2000.3–2002.2	Information Security Group, School of Engineering Information and Communications University, Korea (M.S.)

Career

2001 Spring	Graduate Teaching Assistant ICE525 Computer Security and Electronic Payment System
2001 Fall	Graduate Teaching Assistant ICE673 Security Management

2001.3– Graduate Research Assistant
Development of Electronic Voting System for World Cup 2002
International Research center for Information Security

2000.3–2001.8 Graduate Research Assistant
A study on the Information Security and Authentication Technologies in Distributed Environments
The Ministry of Information and Communications

2000.8–2000.12 Graduate Research Assistant
Characterization of Network Cyber Terror and Research on its Countermeasure
Electronics and Telecommunications Research Institute

2001.12–2001.12 Graduate Research Assistant
e-Biz Security and PKI Applications
LG-EDS Systems

Publications

(1) 2001.11 Manho Lee and Kwangjo Kim, "이동컴퓨팅 환경을 위한 소액지불시스템 (An Efficient Micropayment for Mobile Computing Environment) ", *KIISC 종합학술발표회(CISC2001), 종합학술발표논문집, Vol. 11, No. 1*, pp. 181–184, 2001.

(2) 2000.11 Heesun Kim, Joonsang Baek, Gookwhan Ahn, Jinho Kim, Hyuncheol Park, Boyeon Song, Manho Lee, Jaegwan Park, Jaeseung Go, Byoungcheon Lee, and Kwangjo Kim, "Design and Implementation of Revocable Electronic Cash System based on Elliptic Curve Discrete Logarithm Problem", *Proc. of WISA2000*, pp. 85–102, Korea, 2000.

(3) 2001. Manho Lee, Gookwhan Ahn, Jinho Kim, Jaegwan Park, Byoungcheon Lee, Kwangjo Kim, and Hyuckjae Lee, "Design and Implementation of An Efficient Fair Off-line E-Cash System based on Elliptic Curve Discrete Logarithm Problem", *To appear in JCN (Journal of Communications and Networks)*, 2002.

(4) 2001.12 Manho Lee and Kwangjo Kim, "A Micro-payment System for Multiple-Shopping", *To appear in the 2002 Symposium on Cryptography and Information Security (SCIS2002)*, Wakayama, Japan, 2002.