Digital Right Schemes for Limited Issue and Flexible Division

Chia-Chen Lin[†], Chia-Chi Wu[††] and Chin-Chen Chang[††, †††]

[†]Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan
[††]Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan
[†††]Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan

Summary
Digital right can be applied to various e-commerce applications, such as coupons, tickets and e-cash. The existing digital right schemes only help the issuer to issue digital rights without quantity limitation. However, issuer needs to limit the quantity of his digital rights due to marketing strategy. In this paper, we propose a flexible digital right scheme, such that the issuer can easily control circulation of each broker. Furthermore, we propose one variant based on our first scheme to make a digital right that can be flexibly divided according to owner's demand. The variant scheme conquers the weakness of paper-based gift coupon, and makes the digital rights more flexible in redemption. Both schemes not only satisfy the confidentiality, anonymity, secure transference, preventing double spending and so on, but also expand the applications of digital rights.

Key words:
Digital right, limited issue, flexible division, PKI.

1. Introduction

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

In the past years, many scholars have dedicated to study various electronic payment systems. Certainly, current electronic payment systems can support various useful functions. Take digital cash for instance, it makes sure the owner is untraceable, and it is physically independent, transferable, divisible, off-line capable, and machine-understandable. However, digital cash still has its limitations. For example, the digital cash cannot be specified for particular applications or special goods. Hence, "digital ticket" concept was proposed for widespread use. In general, a digital ticket is a certificate that guarantees certain rights of the ticket owner. We can say the digital cash and micro-payment are special applications of digital ticket.

Some digital tickets, e.g, E-gold [14] and E-Stamp [6], have already been developed. In 1998, Fujimura and Nakajima defined a digital ticket as comprising issuer, promise and owner [8]. Based on above definitions, they clarified the requirements of general-purpose digital ticket and its four unique properties, which are not required for digital cash. They are (1) machine-understandability of ticket contents, (2) state-transitionality of ticket status, (3) composability of multiple tickets, and (4) parameterization of ticket features on untraceability, transferability and divisibility. In addition, they used thirteen properties to compare digital cash with digital ticket. The comparison results are listed in Table 1 as follows.

Table 1. The comparisons between digital cash and digital ticket [8]

Properties	Digital cash	Digital ticket
(1) Secure	Yes	Yes
(2) Anonymous	Yes	Traceable/Untraceable
(3) Physical independence	Yes	Yes
(4) Transferable	Yes	Transferable
(5) Divisible	Yes	Only once/ Specified times/Infinite times
(6) Off-line capable	Yes	Yes
(7) Persistent	Yes	Persistent/ Specified period
(8) Machine-understandable	No	Yes
(9) State manageable	No	Yes
(10) Composable	No	Yes
(11) Wide acceptability	Yes	Yes
(12) User friendly	Yes	Yes
(13) Monetary freedom	Yes	No

In 1999, Fujimura et al. developed a comprehensive digital ticket circulation model shown in Fig. 1 [9]. In Fujimura et al.'s model, there are six entities: CA, issuer, service provider, user, broker and shop. An issuer is in charge of creating, signing, issuing a digital ticket and authorizing brokers to sell digital ticket; a user redeems the ticket; and a service provider fulfills the service expressed by the ticket. A broker sells digital tickets to users. They also defined three types of ticket transactions: (1) issuance: is an action in which issuer grants ownership of tickets to users, (2) transfer: is an action in which a user transfers ownership of ticket to the other user, and (3) redemption: is an action in which a user redeems the rights expressed by ticket to service provider.

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

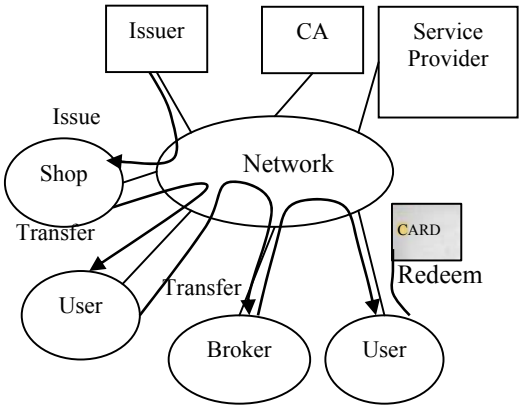

Fig. 1. Ticket circulation model [8]

Later, Fujimura discovered a conventional wallet usually contains many entities, such as cash, credit card, membership card, gift certificate, coupon, admission ticket, loyalty point, plane ticket, and so on, but only the three former ones have been digitalized. Therefore, he defined "Digital Right (DR)" as a digital representation of the right to claim the service and goods, which can be issued by different issues, presents various types of rights, and may be invalidate when it is redeemed or transferred. A digital right defined by Fujimura contains four elements, including issuer, promise, owner and validity-condition. Using the digital right concept, he believed the rest entities of a wallet can be digitalized in the future. To provide a common infrastructure, which can assist any party to issue various digital rights and support consumers to use and transfer their digital rights; Fujimura further proposed a Digital Right Trading Infrastructure (DRTI) [10]. In Fujimura's DRTI, four parties are involved, including an on-line ownership management system (OOMS), issuer, user and service provider.

In 2003, Fujimura and Eastlake extended their discussion to crediting loyalty points and collecting digital coupons or gift certificates [12]. They used the "voucher" concept to represent above activities. They also designed a voucher trading system. Certainly, after Fujimura clarified the definitions of digital ticket and digital right, it is obvious that the digital right can represent more complex services or rights than the digital ticket does. In the following sections, we will discuss digital rights instead of digital ticket. Based on our observation, most of the current digital ticket circulation models and trading systems focus on how to apply digital right concept to different applications and design diverse models, frameworks or systems to help the issuer to issue various types of digital rights, and to support consumers to transfer or redeem their digital rights. Few of them further enhance digital right's function to solve the potential problems caused by digitalization of the paper-based ticket or right, or to solve the paper-based ticket or right's weakness. Two examples are demonstrated as follows to declare our opinions. The first one is an issuer usually issues limited coupons to promote his products. Once an issuer authorizes his brokers to distribute or sell his coupons. Issuers have to print out paper-based coupons and deliver coupons to brokers. Therefore, it is easy to prevent brokers from over-selling coupons. However, it is difficult to prevent brokers from over-selling e-coupons because duplication is quite easy and costless. The other one is current paper-based gift coupon is fixed value. If a consumer uses a coupon to buy a good that is less than the value of the coupon, he may suffer a loss because shop will not return him the price difference. The former one describes the potential problem of digital ticket/ right, and the latter one presents the weakness of traditional paper-based gift coupon. To conquer above problems and to enhance the function of existing digital right, we apply cryptographic techniques to propose two flexible digital right schemes in this paper, one for limited issue and the other for flexible division.

The rest of this paper is organized as follows. In Section 2, we shall briefly review Matsuyama and Fujimura's rights trading system [15]. Our proposed flexible digital right schemes are presented in Section 3. Then, the security analyses are shown in Section 4. Finally, we draw some conclusions in Section 5.

2. A Review of Matsuyama and Fujimura's Rights Trading System

In this section, we shall briefly review Matsuyama and Fujimura's rights trading system [15]. Basically, they proposed the ticket-token management protocols to solve digital ticket transfer problem. In their system, there are four entities: issuer, user, ticket-token manager and service provider involved. Their protocols can be divided into three transactions: issuance transaction, transference transaction and redemption transaction. The detailed descriptions are given as follows.

Issuance transaction

1. User U_0 sends his request and payment to the issuer.
2. The issuer sends his certificate to user U_0.
3. User U_0 sends his certificate to the issuer.
4. The issuer sends a ticket T to user U_0.
5. User U_0 generates a new ticket key K_0 and computes an issue request $R_0 = (h(T), h(K_0))$, where $h(\ )$ is a one-way hash function, and then sends R_0 back to the issuer.
6. The issuer generates the ownership information $IO_0 = (h(T), nil, h(K_0))$ first. Next, he registers IO_0 with the ticket-token manager, where K_0 is the ticket-token for T.
7. The ticket-token manager makes IO_0 public. Hence, user U_0 can evaluate the ownership by verifying IO_0 using K_0.

Transference transaction

Assume user U_0 wants to transfer T to U_1, four steps of transference transaction will be performed as follows.

1. User U_0 sends T to user U_1.
2. User U_1 generates a new ticket-token K_1, and creates a transfer request $R_1=(h(T), h(K_1))$. At last, user U_0 sends R_1 back to U_0.
3. User U_0 generates a new ownership information $IO_1=(h(T), K_0, h(K_1))$ and sends IO_1 to the ticket-token manager.
4. The ticket-token manager compares $h(K_0)$ in IO_0 with the hashed value of K_0 in IO_1. If they are equal, the ticket-token manager replaces the ticket-token K_0 with K_1.

Redemption transaction

If user U_n wants to fulfill his ticket, three steps will be conducted as follows.
1. User U_n presents his ticket T and ticket-token K_n to the service provider.
2. The service provider presents the ownership information $IO_{n+1}=(h(T), K_n, nil)$ to the ticket-token manager.
3. The ticket-token manager checks whether $h(K_n)$ in IO_n is equal to the hash value of K_n in IO_{n+1} given by user U_n. If they are equal, the ticket-token manager deletes all information on ticket T and notifies the service provider that the ownership information is valid. Otherwise, the ticket-token manager will notify the service provider to reject user's redemption.

Matsuyama and Fujimura applied ticket-token to implement transference transaction and verification of the digital ticket ownership [15]. Their idea is simple and their implementation is easy; however, their system does not satisfy the divisible requirement. That means if an issuer adopts Matsuyama and Fujimura's system to implement an e-coupon (e.g., gift coupon) circulation environment, consumers may suffer a loss when what they buy is of less value than the e-coupon's value. In addition, their system neither supports the complex digital ticket circulation model nor discusses the brokers' overissue problem.

To support two additional requirements: limited issue and flexible division, we propose our proposed digital right scheme for limited issue in Subsection 3.3. In Subsection 3.4, we will propose one variant with flexible division property based on our digital right scheme presented in Subsection 3.3. The detailed descriptions of our proposed digital right schemes will be presented in the following section.

3. The Proposed Flexible Digital Right Schemes

Although many scholars treat digital right and digital ticket as the same thing, their functions are not exactly the same. According to Fujimura's definitions [10], digital ticket only contains three items: issuer, promise, owner; but digital right consists of four elements: issuer, promise, owner, and validity-condition. Since digital right can represent more complex services than digital ticket does, we shall adopt digital right in our digital rights trading model, and then we further propose our digital right schemes based on the model shown in Fig. 2.

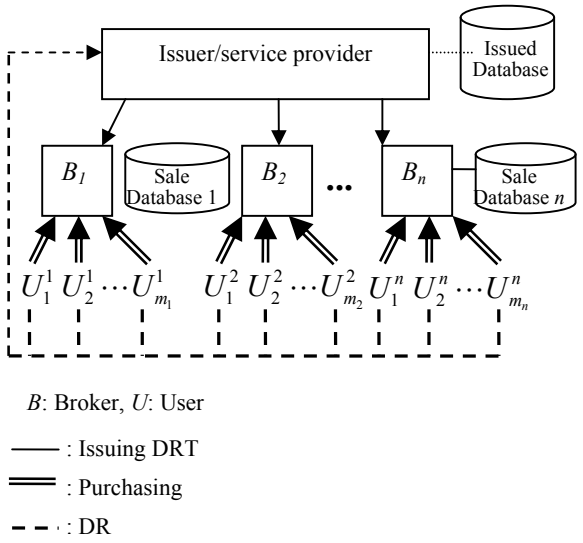

Fig. 2. Our proposed digital rights trading model

In our digital rights trading model, there are only three parties involved: users, service provider and brokers, because the service provider also serves as an issuer in our schemes. In Fig. 2, B_x denotes the broker and U_j^i denotes the user U_j registered at the broker B_i. Three databases are included in our proposed digital rights trading model. The service provider is responsible for issued database, and the broker is in charge of the sale database. Issued database contains *Issued_DRT* table, which stores DRTs issued by the issuer. *Issued_DRT* table is composed of six fields: (1) the identity of the broker ID_B, (2) digital right template *DRT*, (3) the initial serial number of DR that is issued by the issuer *SN_Start*, (4) the initial serial number of DR that is issued by the issuer *SN_End*, (5) the issue date *Issue_Date,* and (6) the last issued serial number *SN*. *Sale Database* is maintained by the broker and records the sold DR's information. Basically, sale database is composed of three tables: *Customer* table, *Sale* table and *DRT* table, shown in Fig. 3.

DRT Table: (ID_{SP}, *DRT, SN_Start, SN_End, Issue_Date*)
Sale Table: (*SN,* ID_U, *DR, Sale_Date*)
Customer Table: (ID_U, $Cert_U$, k_{ij})

Fig. 3. *Sale Database*'s Data Items

In Fig. 3, *Sale_Date* is the sale date. ID_U is the identity of the user. $Cert_U$ is the certificate of customer. k_{ij} is the shared key between user U_i and broker B_j after U_i registered at B_j.

Moreover, each proposed digital right scheme consists of five phases: initialization, issuance, purchasing, redemption and transference phases. In our schemes, we assume that

Public Key Infrastructure (PKI) has existed in the network already; each entity has her/his owner public and private key pair and certificate. Our first scheme is designed to conquer the broker's overissue problem. The second scheme is the variant of our first one to achieve flexible division property. Both of them are based on our proposed digital rights trading model. Therefore, in the following subsections, we first explain our notations in Subsection 3.1. Next, we describe the components of our digital rights in Subsection 3.2. In Subsection 3.3, we shall introduce our first digital right scheme. In the Subsection 3.4, the variant one based on our first scheme is presented.

3.1 The Notations

For convenience, we list the notations in the following.
U_i: The user i.
B_j: The broker j.
IS_k: The issuer k.
ID_{IS}: The identity of the issuer/service provider.
ID_B: The identity of the broker.
ID_U: The identity of the user.
$Cert_x$: Certificate of x entity.
k_{ij}: The shared key between user U_i and broker B_j after U_i registered at B_j.
SN: Serial number.
SN_Start: The initial serial number of DR that is issued by the issuer.
SN_End: The end serial number of DR that is issued by the issuer.
α, β: Two large random numbers.
Curr_Date: Current date.
Issue_Date: Issued date.
Sale_Date: Sale date.
Expi_Date: Expiration date.
Valid_Period: The valid period that is equal to the difference between *Expi_Date* and *Sale_Date*.
Sale_Amount: The limited amount of DR that is determined by the issuer.
$Sign_x(m)$: Using x's private key to sign the message m.
$H^x(m)$: Applying the one-way hash function $H()$ x times to message m.
$E_k(m)$: Using the key k to encrypt the message m.
DRT: Digital right template, which is issued by the issuer. The DRT defines the issuer, promise and validity conditions of the digital right. Each DRT contains an issuer's signature to prove its validity.

3.2 Components of Our Digital Right

In our proposed schemes, each service provider has to determine how many services he would like to provide first. Then, the issuer designs his digital right template (*DRT*) for each service. Each *DRT* contains four components, including ID_{IS}, P_i, V_i, $Sign_{IS}(H(ID_{IS}, P_i, V_i))$. P denotes promise, which is promise or services guaranteed by the issuer *IS*. V denotes the validity conditions defined by the issuer for each service or promise. For example, if McDonald wants to generate one kind of e-coupon to allow his customers to buy one drink with a fifty percent discount during a specific period, e.g., January 2005. McDonald has to generate a *DRT* first. In the DRT, McDonald is the issuer and is the service provider, so ID_{IS} is McDonald's identification. P indicates fifty percent discount for each drink, and V indicates January 2005.
After generating *DRT*s for different services, the issuer further authorizes some brokers to generate their digital rights (*DR*s) according to issuer' *DRT*s. The authorized brokers will sell *DR*s to customers later. Basically, a *DR* contains five components: digital right's serial number *SN*, digital right template *DRT*, customer's purchase date *Sale_Date*, the digital right's expiration date *Expi_Date*, and a hash value $H^{Valid\_Period}(\alpha)$. *DR* does not contain any information related to its owner. Only *DRO* represents the ownership of a digital right. Therefore, when a customer purchases a digital right, the broker has to use his secret key to generate a *DRO* $Sign_{Bj}(H(DR,H(ID_{Ui})))$ for the customer. Only legal owner of a digital right can present a valid *DRO*. In our proposed schemes, the customer has to present his *DRO* and *DR* together to prove his ownership of his digital right when he wants to redeem or transfer his digital right.

3.3 The Proposed Digital Right Scheme with Limited Issue Property

In this subsection, we present the proposed scheme with limited issue property. The proposed scheme is divided five phases: initialization, issuance, purchasing, transference and redemption. In our proposed scheme, users purchase their digital rights first. Then, they can decide to transfer their digital rights to others or redeem their digital rights for specific services or goods. Each *DR* is only allowed to be redeemed once. The details of five phases are described in the following.

Initialization Phase

In this phase, issuers define their digital right templates *DRT*s, and record them in their Issued databases. Brokers record their authorized *DRT*s in their *DRT* tables. User U_i takes the following steps to register at the broker B_j before he wants to buy digital rights.
Step 1. User U_i generates a session key *Key* first. Next, user U_i encrypts his identity, *Key* and certificate $Cert_i$ using brker B_j's public key. At last, he sends them to broker B_j for registration.
Step 2. After B_j receives above message, B_j decrypts it using his private key first. Then, B_j verifies user's $Cert_i$, and checks whether U_i exists in his *Customer Table* using user's $Cert_i$. If he is not being, B_j generates a unique ID_{Ui} and a shared key k_{ij}. Then, he encrypts ID_{Ui} and k_{ij} using the session key *Key*. Finally, B_j sends encrypted data to U_i; and

stores ID_{Ui}, $Cert_i$, k_{ij} in his *Customer Table*. Otherwise, B_j will inform user U_i that he is already registered.
Step 3. After receiving above message, U_i decrypts it first. Next, he stores (ID_{Ui}, k_{ij}) in his smart card.

Issuance Phase

In this phase, the broker sends a request to an issuer for being an agency for selling digital rights. If the issuer authorizes a broker to be his agency, he has to decide the issue quantity of the authorized digital rights. In other words, the issuer has to determine how many digital rights will be sold by the authorized broker. This phase can de divided into four steps. All messages transmitted in the following steps are encrypted by the receiver's public key to achieve data confidentiality.
Step 1. Broker *Bj* sends his ID_B and request to the issuer IS_k for being an agency to sell the digital rights of DRT_i.
Step 2. Issuer IS_k determines that B_j can sell n units of *DRs*, then IS_k generates *SN_Start* and *SN_End* which contains n serial numbers for broker B_j .
Step 3. Issuer IS_k sends DRT_i:{ID_{IS}, P_i, V_i, $Sign_{IS}$($H(ID_{IS}, P_i, V_i,)$)} and (*SN_Start, SN_End*) to B_j. Meanwhile, *IS* stores {ID_{Bj}, DRT_i, *SN_Start, SN_End*, *Issue_Date*} in his *Issue_DRT Table* for later tracing.
Step 4. After receiving the above messages, broker B_j stores them into his *DRT Table* and sends an acknowledgement to issuer IS_k.

Purchasing Phase

In this phase, the registered users purchase digital rights *DR*s from broker B_j and verify D*R*s' validity. This phase can be broken down into six steps as follows. We briefly illustrate them in Fig. 4.

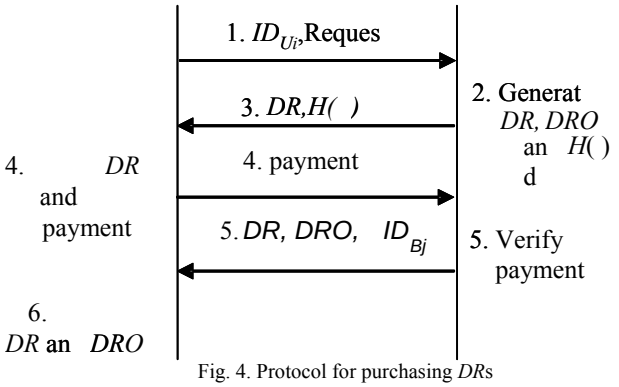

Fig. 4. Protocol for purchasing *DR*s

Step 1. User U_i determines which *DRT* he wants to buy. Next, user U_i sends his ID_{Ui} and request to broker B_j.
Step 2. Broker B_j generates a new *SN* and checks the *DRT Table* to see whether *SN* is less than or equal to *SN_End*. If *SN* is larger than *SN_End*, B_j has to reject U_i's request. Broker B_j generates a random number α. Then, B_j generates *DR* and *DRO* pair according to the user U_i's choice:
DR:{*SN*,DRT_i, *Sale_Date*, *Expi_Date*, $H^{Valid\_Period}(\alpha)$},
DRO:{$Sign_{Bj}$($H(DR,H(ID_{Ui}))$)}.
Step 3. Broker B_j sends *DR* and $H(\alpha)$ to user U_i.
Step 4. User U_i checks *DR* to see whether it meets his request or not. If it does, user U_i sends payment instrument to broker B_j.
Step 5. After receiving user's payment, broker B_j verifies its validity. If it is valid, broker B_j sends *DRO* to user U_i. Meanwhile, broker B_j records (*SN*, ID_{Ui}, *DR*, *Sale_Date*) in his *Sale Table* and sends (*DR*, *DRO*, ID_{Bj}) to U_i.
Step 6. After receiving the above messages, user U_i computes $H(DR,H(ID_{Ui}))$ and compares it with decrypted *DRO* to verify the integrity of his *DR*. If they are equal, he stores ($H(\alpha)$, *DR*, *DRO*, ID_{Bj}) into his smart card for later redemption.

Redemption phase

In this phase, the user U_i wants to redeem his *DR*s to the issuer *IS* for getting services or goods. Before accepting user's *DR*, issuer IS_k checks whether the user is a legal owner of *DR*. Next, issuer IS_k checks *SN* to make sure the *DR* is not double spending. Five steps will be conducted as follows. We demonstrate them in Fig. 5.

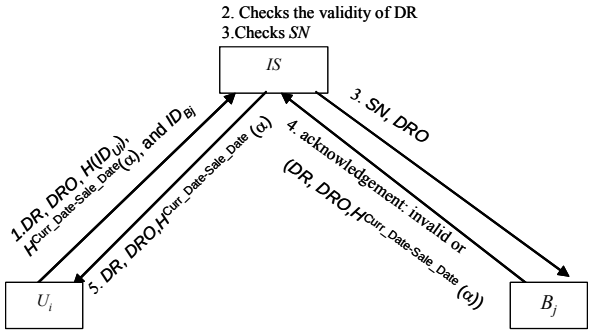

Fig.. 5 Protocol for redemption

Step 1. *User* U_i sends [*DR*, *DRO*, $H(ID_{Ui})$, $H^{Curr\_Date-Sale\_Date}(\alpha)$, ID_{Bj}] to issuer IS_k to get the related goods or services.
Step 2. After receiving the above message, issuer IS_k decrypts *DRO* using broker B_j's public key first. Next, issuer IS_k computes $H(DR, H(ID_{Ui}))$ using *DR* and $H(ID_{Ui})$ provided by user U_i and compares with decrypted *DRO*. If they are equal, the digital right's ownership is confirmed. Finally, issuer IS_k calculates $H^{Expi\_Date\ -Curr\_Date}(H^{Curr\_Date-Sale\_Date}(\alpha))$, and checks whether it is equal to $H^{Valid\_Period}(\alpha)$ or not. If they are equal, that means the digital right is not expired.
Step 3. Issuer IS_k checks whether *SN* of *DR* is between *SN_Start* and *SN_End* in the *Issu_DRT Table* through indexing by ID_{Bj}. If it holds, *IS* sends {*SN, DRO*} to broker B_j to perform on-line verification for double spending.
Step 4. Broker B_j retrieves the record of *Sale Table* according to his received *SN* first. Next, broker B_j verifies the validity of his received *DRO*. If it is valid, B_j marks this record in the *Sale Table* to note that *SN* has been redeemed and updates the status of *DR* as (*DR, DRO,* H^{Curr_Date-}

$^{Sale\_Date}(\alpha)$). Finally, B_j sends the last status of *DR* to issuer IS_k. Otherwise, B_j notifies *IS* that *DR* is invalid.

Step 5. If broker's acknowledgement is positive, issuer IS_k provides user U_i goods or services and returns the recipient, $Sign_{IS}(H(DR, H^{Curr\_Date\text{-}Sale\_Date}(\alpha))$, to user U_i. Otherwise, issuer IS_k rejects user's request.

Transference transaction

Assume U_i and U_k are registered users. If user U_i wants to transfer his *DR* to user U_k, eight steps will be performed as follows. The protocol for transference transaction is shown in Fig. 6.

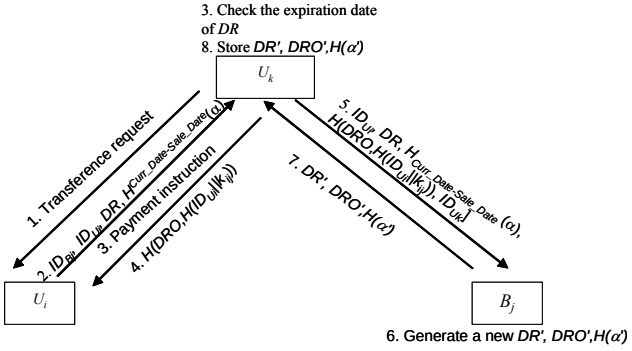

Fig. 6. Protocol for transference transaction

Step 1. User U_k sends a digital right transference request to user U_i.

Step 2. User U_i sends $[ID_{Bj}, ID_{Ui}, DR, H^{Curr\_Date\text{-}Sale\_Date}(\alpha)]$ to user U_k.

Step 3. User U_k computes $H^{Expi\_Date\ -Curr\_Date}(H^{Curr\_Date\text{-}Sale\_Date}(\alpha))$, and checks whether the result is equal to $H^{Valid\_Period}(\alpha)$ of *DR* or not. If it is valid, user U_k sends a payment instruction to user U_i.

Step 4. If the payment instruction is correct, user U_i sends $[H(DRO,H(ID_{Ui}||k_{ij}))]$ to user U_k. Otherwise, the transaction is terminated.

Step 5. User U_k sends $[ID_{Bj},ID_{Ui}, DR, H^{Curr\_Date\text{-}Sale\_Date}(\alpha), H(DRO,H(ID_{Ui}||k_{ij})), ID_{Uk}]$ to broker B_j.

Step 6. After receiving the above messages, broker B_j performs the following substeps.

Step 6.1 Broker B_j rertieves data from *Sale Table* according to his received *SN* and computes $\lambda=\{Sign_{Bj}(H(DR,H(ID_{Ui})))\}$. Next, he retrieves the (ID_{Ui}, k_{ij}) from his *Customer Table* by indexing ID_{Ui} to compute $H(\lambda, H(ID_{Ui}||k_{ij}))$. Broker B_j compares it with his received $H(DRO,H(ID_{Ui}||k_{ij}))$. If they are equal, the *DR* and identity of U_i are verified.

Step 6.2 Broker B_j computes $H^{\text{Expi\_Date-Curr\_Date}}(H^{\text{Curr\_Date-Sale\_Date}}(\alpha))$ and compares it with $H^{\text{Valid\_Period}}(\alpha)$. If they are equal, broker B_j marks the record of *Sale Table* to note that it is transferred.

Step 6.3 Broker B_j generates new *DR* , *DRO'* and a new random number α' for user U_k as follows:

Sale_Date'=Curr_Date, and Valid_Period'=Expi_Date-Curr_Date.

DR':{*SN, DRT*$_i$, *Sale_Date'*, *Expi_Date*, $H^{Valid\_Period'}(\alpha')$}, *DRO'*:{$Sign_{Bj}(H(DR',H(ID_{Uk})))$}.

Step 7. Broker B_j stores (*SN*, ID_{Uk}, *DR', Sale_Date'*) into *Sale Table* and sends (*DR', DRO',H(α')*) to user U_k.

Step 8. After receiving the above messages, user U_k computes $H(DR,H(ID_{Uk}))$ and compares it with decrypted *DRO* to verify the integrity of his *DR*. If they are equal, he stores (*DR', DRO', H(α')*, ID_{Bj}) in his smart card for later redemption.

In our proposed scheme, the issuer determines the issue quantities of his digital rights for his authorized brokers. Each broker assigns a unique serial number for his issued digital rights. In other words, each digital right contains a unique serial number, which can be checked by the issuer during the redemption phase. Brokers can not overissue digital rights without being discovered by the issuers. Therefore, our proposed scheme can help issuers to issue limited digital rights. Moreover, the fields related to limited issue are optional. The steps related to checking the issued number are also optional. The issuer can only record *Start_SN* and *End_SN* in his issued database for his *DRT* with limited issue. That means our proposed scheme also can support issuers to issue digital rights without limited quantity.

3.4 The Proposed Digital Right Scheme with Flexible Division Property

Although our proposed scheme presented in Subsection 3.3 can help the issuer to issue limited digital rights, it could be damage user's interest when it is applied to issue e-gift coupons, because our proposed scheme does not have the flexible division property.

In the existing paper-based gift coupon systems, if the good's price is less than the value of gift coupon, users have two choices. One is that users try to buy more goods and make sure the total price is equal to the value of gift coupon. The other one is that users pay for goods by using their gift coupons and their interests are damaged. To conquer the weakness of paper-based gift coupon, we try to extend our proposed scheme to achieve the flexible division and limited issue properties simultaneously.

In the variant scheme, we assume each *DR* has a fixed value. The structure of digital right DR is modified as *DR*:{*SN, DRT*$_i$, *Sale_Date*, *Expi_Date*, $H^{Valid\_Period}(\alpha)$, $H^{DR\_Value}(\beta)$}. *DR_Value* is the fixed value of each *DR*. In addition, the *Sale Table* maintained by the broker is modified to provide flexible division function, shown in Fig. 7. The balance equals *DR_Value* minus *Pay_Value*, where *Pay_Value* is the value paid by the user for some services or goods.

SN	ID_U	*DR*	*Balance*	*Sale_Date*

Fig. 7. The modified *SaleTable*

Our variant scheme also consists of initialization, issuance, purchasing, and redemption phases. It also can support users to transfer their digital rights. Basically, the initialization phase and issuance phase are the same as our proposed scheme presented in Subsection 3.3. In the following paragraphs, we shall introduce the rest phases: purchasing phase, redemption phase and transference transaction of our variant scheme.

Purchasing Phase

In general, users conduct the following steps to purchase digital rights from the broker. The protocol for our variant's purchasing phase is presented in Fig. 8.

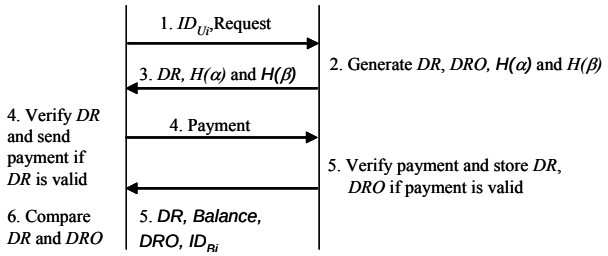

Fig. 8. Protocol for our variant's purchase phase

Step 1. User U_i determines which *DRT* he wants to buy and sends his request to broker B_j.

Step 2. Broker B_j generates a new *SN* and checks the *DRT Table* to see whether *SN* is less than or equal to *SN_End* or not. If *SN* is larger than *SN_End*, B_j has to reject U_i's request. Broker B_j generates two random numbers α and β. Then, B_j generates *DR* and *DRO* according to the user U_i's choice: DR:{SN, DRT_i, $Sale\_Date$, $Expi\_Date$, $H^{Valid\_Period}(\alpha)$, $H^{DR\_Value}(\beta)$}, DRO:{$Sign_{Bj}(H(DR,H(ID_{Ui})))$}.

Step 3. Broker B_j sends *DR*, $H(\alpha)$ and $H(\beta)$ to user U_i.

Step 4. User U_i checks *DR* to see whether it meets his request or not. If it does, user U_i sends payment instrument to broker B_j.

Step 5. After receiving user's payment, broker B_j verifies its validity. If it is valid, broker B_j sends *DRO* to user U_i. Meanwhile, broker B_j records (*SN*, ID_{Ui}, *DR*, *Sale_Date*) in his *Sale Table* and sets *Balance* as *DR_Value*. Finally, broker B_j sends (*DR*, *Balance*, *DRO*, ID_{Bj}) to U_i.

Step 6. After receiving the above messages, user U_i computes $H(DR,H(ID_{Ui}))$ and compares it with decrypted *DRO* to verify integrity of *DR*. If they are equal, he stores ($H(\alpha)$, $H(\beta)$, *DR, Balance*, *DRO*, ID_{Bj}) in his smart card for later redemption.

Redemption Phase

In this phase, U_i redeems a part of *DR* value to issuer IS_k. Issuer IS_k notifies broker B_j to check the *Balance* of user's *DR*. If the *Balance* is enough, then issuer IS_k will permit user U_i's redemption. Since our variant scheme can allow user to redeem a part of his digital right DR, we assume user U_i wants to redeem *Pay_Value* and *Pay_Value* is less than *Balance* of his *DR*. The Protocol of variant scheme's redemption phase is shown in Fig. 9.

Step 1. User U_i sends {*DR, DRO*, $H(ID_{Ui})$, $H^{Curr\_Date-Sale\_Date}(\alpha)$, $H^{Balance}(\beta)$, $H^{Balancee-Pay\_Value}(\beta)$, *Pay_Value*, ID_{Bj}} to issuer IS_k for redemption *Pay_Value* of his *DR*.

Step 2. After receiving the above message, issuer IS_k performs the following substeps.

Step 2.1 Issuer IS_k first computes $H(DR, H(ID_{Ui}))$ and compares it with the decrypted *DRO*. If they are equal, the ownership of digital right is confirmed.

Step 2.2 Issuer IS_k checks $H^{Expi\_Date-Curr\_Date}(H^{Curr\_Date-Sale\_Date}(\alpha))$ to see whether it is equal to $H^{Valid\_Period}(\alpha)$ or not. If they are equal, the digital right is not expired.

Step 2.3 Issuer IS_k uses his stored *Balance* to check $H^{DR\_Value-Balance}(H^{Balance}(\beta))$ to see whether it is equal to $H^{DR\_Value}(\beta)$. If they are equal, the $H^{Balance}(\beta)$ is correct.

Step 2.4 Issuer IS_k checks $H^{Pay\_Value}(H^{Balance-Pay\_Value}(\beta))$ to see whether it is equal to $H^{Balance}(\beta)$. If they are equal, the *Pay_Value* is verified.

Step 3. Issuer IS_k sends {*SN*, $H^{Balance}(\beta)$, $H^{Balance-Pay\_Value}(\beta)$, *Pay_Value*} to broker B_j to perform on-line validation for double spending.

Step 4. After receiving the above messages, broker B_j performs the following substeps.

Step 4.1 Broker B_j retrieves *Balance* and $H^{DR\_Value}(\beta)$ from his *Sale Table* by indexing his received *SN*.

Step 4.2 Broker B_j uses his received $H^{Balance}(\beta)$ to compute $H^{DR\_Value-Balance}(H^{Balance}(\beta))$. If it is equal to $H^{DR\_Value}(\beta)$ that is retrieved from *DR* in *Sale Table*, the validity of $H^{Balance}(\beta)$ is confirmed.

Step 4.3 Issuer IS_k uses his received *Pay_Value* and $H^{Balance-Pay\_Value}(\beta)$ to compute $H^{Pay\_Value}(H^{Balance-Pay\_Value}(\beta))$. If it is equal to $H^{Balance}(\beta)$ that is derived from Step 4.2, broker B_j updates the *Balance* in *Sale Table* as (*Balance-Pay_Value*). Otherwise, broker B_j rejects the transaction, and notifies issuer IS_k that the *DR* is invalid.

Step 5. If the notification is positive, issuer IS_k provides goods or services for user U_i, and sings [*DR*, $H^{Curr\_Date-Sale\_Date}(\alpha)$, $H^{Balance}(\beta)$, $H^{Balance-Pay\_Value}(\beta)$, *Balance*, *Pay_Value*] as a receipt for user U_i.

Step 6. Issuer IS_k sends receipt to user U_i.

Step 7. User U_i updates his *Balance* as (*Balance-Pay_Value*) in his smart card, and keeps his receipts.

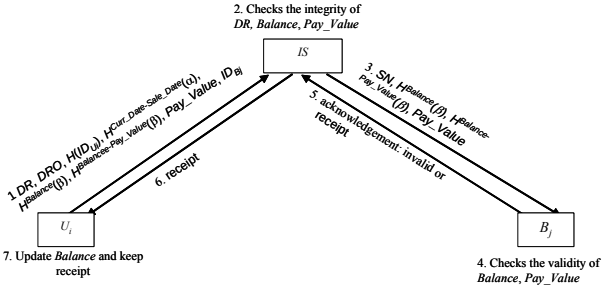

Fig. 9. Protocol for variant's redemption phase

Transference Transaction

In general, the transference transaction of our variant scheme is similar to our proposed scheme presented in Subsection 3.3. The major difference between them is the buyer in the variant scheme not only checks the validity of *DR* but also has to check the balance of *DR*. Assume user U_i wants to transfer his *DR* to user U_k. Both of them are registered users. The transference transaction can be broken down into eight steps. The protocol for variant scheme's transference transaction is shown in Fig. 10.

Step 1. User U_k sends a digital right transference request to user U_i.

Step 2. User U_i sends [ID_{Bj}, ID_{Ui},*DR*, $\tau=H^{\text{Curr\_Date-Sale\_Date}}(\alpha)$, $\upsilon=H^{Balance}(\beta)$, *Balance*] to user U_k.

Step 3. User U_k checks $H^{\text{Expi\_Date-Curr\_Date}}(H^{\text{Curr\_Date-Sale\_Date}}(\alpha))$ to see whether it is equal to $H^{\text{Valid\_Period}}(\alpha)$ of *DR* or not. If they are equal, *DR* is not expired and user U_k sends a payment instruction to user U_i.

Step 4. If the payment instruction is correct, user U_i sends [$H(DRO,H(ID_{Ui}||k_{ij}))$] to user U_k. Otherwise, the transaction is terminated.

Step 5. User U_k sends [ID_{Bj}, ID_{Ui},*DR*, $\tau=H^{\text{Curr\_Date-Sale\_Date}}(\alpha)$, $\upsilon=H^{Balance}(\beta)$, *Balance*, $H(DRO,H(ID_{Ui}||k_{ij}))$, ID_{Uk}] to broker B_j.

Step 6. After receiving the above messages, broker B_j performs the following substeps.

Step 6.1 Broker B_j retrieves data from his *Sale Table* according to *SN* and computes $\mu=\{Sign_{Bj}(H(DR,H(ID_{Ui})))\}$. Next, he retrieves the (ID_{Ui}, k_{ij}) from his *Customer Table* by indexing ID_{Ui}, computes $H(\mu, H(ID_{Ui}||k_{ij}))$ and compares it with his received $H(DRO,H(ID_{Ui}||k_{ij}))$. If they are equal, *DR* and the identity of U_i is verified.

Step 6.2 Broker B_j computes $H^{Expi\_Date-Curr\_Date}(\tau)$ and $H^{DR\_Value-Balance}(\upsilon)$ and compares them with $H^{Valid\_Period}(\alpha)$ and $H^{DR\_Value}(\beta)$ of *DR*, respectively. If they are all equal, broker B_j marks the record of *Sale Table* as transferred.

Step 6.3 Broker B_j generates two random numbers α', β' and then generates new *DR'* and *DRO'* for user U_k as follows:

Sale_Date'=*Curr_Date*, *Valid_Period'*=*Expi_Date-Curr_Date*.

DR':{*SN*,DRT_i, *Sale_Date'*, *Expi_Date*, $H^{Valid\_Period'}(\alpha')$, $H^{DR\_Value}(\beta')$},

DRO':{$Sign_{Bj}(H(DR',H(ID_{Uk})))$}.

Step 7. Broker B_j stores (*SN*, ID_{Uk}, *DR', Balance, Sale_Date'*) into his *Sale Table* and sends (*DR', DRO', Balance, H(α'), H(β')*) to user U_k.

Step 8. After receiving the above messages, user U_k computes $H(DR',H(ID_{Uk}))$ and compares it with decrypted *DRO* to verify the integrity of his *DR*. If they are equal, he stores ($H(\alpha')$, $H(\beta')$, *DR', Balance*, *DRO'*, ID_{Bj}) (*DR', DRO',H(α')* , ID_{Bj}) into his smart card for later redemption.

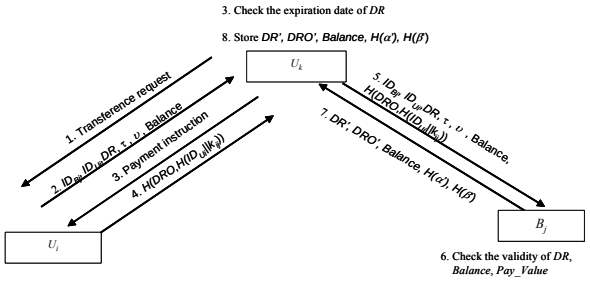

Fig. 10. Protocol for variant's transference transaction

4. The Security Analysis

In this section, we are going to show that our proposed schemes are secure. First, we discuss the security of our first scheme in Subsection 4.1. Then, we will analyze the security of our variant scheme in Subsection 4.2. In security analyses, we summarize security issues, such as confidentiality, anonymity, verifiability, preventing forgery, preventing alternation, preventing duplicate-redemption, preventing reproduction, non-repudiation and trust manageability, proposed by Fujimura and Nakajima [8], and Fujimura and Eastlake [12].

4.1 The Security of Our First Scheme

In the following, we are going to show how our first proposed scheme meets the following security requirements.

1. Confidentiality:

In our proposed scheme, we assume PKI exists. Since each party can easily find out others' certificates and get their public keys, each transmission is performed through the secure channel. Even in our initialization phase, the user can generate a symmetric session key when he wants to register at the broker. Broker can use user's session key to encrypt data and sends them back. When registration is completed, the broker will get user's certificate. In other words, the broker can use user's public key to encrypt transmitted data later. Therefore, in our proposed scheme, the confidentiality is guaranteed.

2. Anonymity:

In purchasing phase, the broker will generate a unique identity for each user. In redemption phase, the user presents [*DR*, *DRO*, $H(ID_{Ui})$, $H^{Curr\_Date-Sale\_Date}(\alpha)$, ID_{Bj}] to the issuer. Issuer *IS* can compute $H(DR, H(ID_{Ui}))$ using his received data, and compares it with that of *DRO*. If they are equal, the *DR*'s ownership is confirmed. Since user only presents his identity in a hashed value, issuer can verify *DR*'s ownership and he does not know who the owner is. User's anonymity is achieved in our scheme.

3. Verifiability:

In our proposed scheme, there are three items, which need to be verified: *DR*, the ownership of *DR* and *DR*'s expiration date. Since the broker will sign each digital right

DR, any party can use broker's public key to verify *DR*'s validity. The verifiability of *DR*'s ownership can be achieved by using the corresponding *DRO*, because *DRO* consists of *DR* and hash value of user's identity. Once a user presents his *DR* and the hash value of his identity, the issuer and broker can easily verify the ownership of user's *DR*. The expiration date of *DR* also can be verified easily, because the user has to present [*DR*, *DRO*, $H(ID_{Ui})$, $H^{Curr\_Date\text{-}Sale\_Date}(\alpha)$, ID_{Bj}] to the issuer in the redemption phase. Issuer just simply checks whether $H^{Expi\_Date\text{-}Curr\_Date}(H^{Curr\_Date\text{-}Sale\_Date}(\alpha))$ is equal to $H^{Valid\_Period}(\alpha)$ or not. If they are not equal, it means *DR* is expired. In transference transaction, buyers also can use the same way to check whether their *DR*s are valid or not. To sum up, the verifiability is achieved in our proposed scheme.

4. Preventing forgery:

In our proposed scheme, the broker signs each *DRO*. Since *DR* is one component of *DRO*, and only the broker has his private key, no one can forge *DR* or *DRO* without being discovered.

5. Preventing alternation:

If user tries to alter the valid period of his digital right, or modify the promise of his digital right, he has to get the broker's private key first. However, user has no chance to get broker's private key. That means user cannot alter his digital right without compromising his digital right's integrity and validity.

6. Preventing duplicate-redemption:

In the redemption phase, user has to present [*DR*, *DRO*, $H(ID_{Ui})$, $H^{Curr\_Date\text{-}Sale\_Date}(\alpha)$, ID_{Bj}] to the issuer. Issuer checks whether *SN* of *DR* is between the *SN_Start* and *SN_End* in the *Issu_DRT Table* through indexing by ID_{Bj}. If *SN* of *DR* is valid, issuer sends {*SN*, *DRO*} to broker B_j to perform one-line verification. Broker B_j further checks his *Sale Table* according to {*SN, DRO*}. If *SN* exists in broker's *Sale Table* and is not marked, then *DR* can be redeemed for goods or services. Otherwise, the *DR* has been spent and broker will inform issuer to reject user's request. Since *SN* is unique in each *Sale Table*, the double-redemption can be prevented.

7. Preventing reproduction:

In our proposed scheme, user has less intention to reproduce his *DR* and transfer the reproduced *DR* to the other users, because he also has to provide his $H(ID_{Ui})$ to make the reproduced *DR* can pass issuer's verification in the redemption phase. However, it may make user unable to redeem his *DR* and damage his own interests. Therefore, our scheme can prevent reproduction indirectly. If a user redeems his *DR* first, and transfers his reproduced *DR* to the other users later, the reproduced *DR* will be discovered as duplicate-redemption in the redemption phase. Therefore, in our proposed scheme, users can not reproduce their *DR*s without being discovered.

8. Non-repudiation:

In our proposed scheme, issuer signs his *DRT*, and broker signs the *DRO*s. They can not claim that they do not issue *DRT*s and *DRO*s. In the transference phase, once user U_i agrees to transfer his *DR* to user U_k, user U_i sends [ID_{Bj}, ID_{Ui},*DR*, $H^{Curr\_Date\text{-}Sale\_Date}(\alpha)$, *Balance*, $H(DRO,H(ID_{Ui}||k_{ij}))$] to user U_k. Since the shared key k_{ij} is a secret shared between user U_i and the broker for a *DR*. If user U_k can prevent a valid $H(DRO,H(ID_{Ui}||k_{ij}))$, user U_i can not deny that he promises to transfer his *DR* to user U_k.

9. Trust manageability:

In the transference phase, the broker is in charge of the transference transaction and checks transferred *DR* status for seller and buyer. If a dispute occurs, seller and buyer can ask the broker to provide evidence. Therefore, our proposed scheme can achieve trust manageability.

4.2 The Security of our variant scheme

In this section, we shall focus on the security issue related to the flexible division property of *DR*. Basically, our variant scheme may suffer from some attacks as follows.

1. User U_i wants to modify *Balance* value of his digital right:

In the redemption phase, user has to present [*DR, DRO*, $H(ID_{Ui})$, $H^{Curr\_Date\text{-}Sale\_Date}(\alpha)$, $H^{Balance}(\beta)$, $H^{Balancee\text{-}Pay\_Value}(\beta)$, *Pay_Value*, ID_{Bj}] to issuer. Issuer can retrieve $H^{DR\_Value}(\beta)$ from his received *DR*, computes $H^{DR\_Value\text{-}Balance}(H^{Balance}(\beta))$, and checks whether they are equal. If they are not equal, issuer will treat *DR* as invalid, and reject user's request. If user wants to pass issuer's verification, he has to modify $H^{DR\_Value}(\beta)$ of *DR*. Since *DRO* contains the original *DR*, and *DRO* is signed by the broker. User has no chance to modify his *DR*'s balance value and forge broker's signature without being discovered by the issuer.

2. Issuer may want to forge $H^{Balance}(\beta)$ in order to get more benefits.

For example, a user still has ten units of his digital right, but the issuer claims that the user only has five units left. In this case, the user can present his receipt to broker B_j or the judge and prove the issuer is cheating. Since the receipt is signed by the issuer in the redemption phase, and the receipt contains $H(DR, H^{Curr\_Date\text{-}Sale\_Date}(\alpha), H^{Balance}(\beta), H^{Balance\text{-}Pay\_Value}(\beta)$, *Balance*, and *Pay_Value*). By chaining each receipt, the broker can find out which redemption transaction is incorrect and ask the issuer to correct it.

5. Conclusions

In this paper, we first develop a digital right scheme that helps an issuer to issue limited quantity of his digital rights. Then, we extend our proposed scheme to achieve limited issue and flexible division properties at the same time. All of them can satisfy confidentiality, anonymity, verifiability, preventing forgery, preventing alternation, preventing

duplicate-redemption and related security requirements. In addition, digital rights can be transferred fairly among users in both schemes.

Our schemes also can support issuer to issue digital right without limited quantity by slight modification. In general, our schemes extend the applications of digital rights. Nevertheless, our computation cost is high in both schemes due to adopt public key system to achieve data confidentiality. In the future, we will try to reduce the computation cost to apply our schemes to a mobile commerce environment.

References

[1] F. Bao, "A Scheme of Digital Ticket for Personal Trusted Device," *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, Vol. 4, pp. 3065-3069, 2004.

[2] D. Chaum, "Blind Signatures for Untraceable Payment," *Proceedings of Advanced in Cryptology-CRYPTO'82,* New York, pp. 199-203, 1983.

[3] D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash," *Proceedings of Advanced in Cryptology-CRYPTO'88, LNCS 403*, Springer-Verlag, pp. 319-327, 1989.

[4] D. Chaum and S. Brands, "Minting Electronic Cash," *Spectrum, IEEE,* Vol. 34, Issue 2, pp. 30-34, Feb. 1997.

[5] E-Stamp Corporation, "E-Stamp," http://www.e-stamp.com/.

[6] C. I. Fan, W. K. Chen and Y. S. Yeh, "Date Attachable Electronic Cash," *Computer Communications*, Vol. 23, Issue 4, pp. 425-428, Feb. 2000.

[7] A. O. Freier, P. Karlton, and P. C. Kocher, "The SSL Protocol Version 3.0," *IETF Internet Draft*, 1996, http://wp.netscape.com/eng/ssl3/draft302.txt.

[8] K. Fujimura and Y. Nakajima, "General-Purpose Digital Ticket Framework," *Proceedings of the 3rd USENIX Workshop on Electronic Commerce,* Boston, Massachusetts, USA, pp. 177-186, Aug. 1998.

[9] K. Fujimura, H. Kuno, M. Terada, K. Matsuyama, Y. Mizuno, and J. Sekine, "Digital-Ticket-Controlled Digital Ticket Circulation," *Proceedings of the 8th USENIX Security Symposium,* Washington D.C., USA, pp. 229-238, Aug. 1999.

[10] K. Fujimura, "Digital-Right Trading Infrastructure (DRTI)," *Proceedings of the 46th Internet Engineer Task Force (IETF),* Washington D.C., USA, Nov. 1999, http://www.ietf.org/proceedings/99nov/slides/trade-drti/index.htm.

[11] K. Fujimura, M. Terada and J. Sekine, "A World Wide Supermarket Scheme Using Rights Trading System," *Proceedings of 7th IEEE International Conference on Parallel and Distributed Systems Workshops,* pp.289- 294, Jul. 2000.

[12] K. Fujimura and D. Eastlake, "Requirement and Design for Voucher Trading System (VTS)," *IETF Internet Draft, 2003,* http://www.faqs.org/rfcs/rfc3506.html.

[13] K. Fujimura and M. Terada, "XML Voucher: Generic Voucher Language," *IETF Draft,* 2005, http://www.ietf.org/internet-drafts/draft-ietf-trade-voucher-lang-07.txt.

[14] Gold & Silver Reserve, Inc., "E-Gold," http://www.e-gold.com/.

[15] K. Matsuyama and K. Fujimura, "Distributed Digital-Ticket Management for Rights Trading System," *Proceedings of the 1st ACM Conference on Electronic Commerce,* pp.110-118, Nov. 1999.

[16] T. Okamoto and K. Ohta, "Universal Electronic Cash," *Proceedings of the Advances in Cryptology-Crypto'91*, Springer-Verlag, pp.324-337, 1992.

[17] R. Song and L. Korba, "How to Make E-cash with Non-repudiation and Anonymity," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04),* Vol. 2, pp. 167 – 172, Las Vegas, Nevada, USA, Apr. 2004.

[18] M. Terada and K. Fujimura, "Voucher Trading System Application Programming Interface (VTS-API)," *IETF Internet Draft,* 2004, http://www.ietf.org/internet-drafts/draft-ietf-trade-voucher-vtsapi-06.txt.

Chia-Chen Lin received her B.S. degree in information management in 1992 from the Tamkang University, Taipei, Taiwan. She received both her M.S. degree in information management in 1994 and Ph.D. degree in information management in 1998 from the National Chiao Tung University, Hsinchu, Taiwan. Dr. Lin is currently an associate professor of the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image hiding, mobile agent, electronic commerce and digital rights management.

Chia-Chi Wu was born in 1967 Taoyuan, Taiwan, Republic of China (ROC.). He is current a Ph.D. candidate in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Chin-Chen Chang received his B.S. degree in Applied Mathematics in 1977 and his M.S. degree in Computer and Decision Sciences in 1979 from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.d. in Computer Engineering in 1982 from National Chiao Tung University, Hsinchu, Taiwan. From 1983 to 1989, he was the faculty at the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. Since August 1989, he was a professor of the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. Dr. Chang is a Fellow of IEEE and a Fellow of IEE. He is also a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, and the Phi Tau Phi Society of the Republic of China. His research interests include database design, computer cryptography, and data compression.