A Study on Implementation and Design of Scheme to Securely Circulate Digital Contents*

디지털콘텐츠의 안전한 유통을 위한 구조 설계 및 구현에 관한 연구

Yong Kim**
EunJeong Kim***

ABSTRACT

With explosive growth in the area of the Internet and IT services, various types of digital contents are generated and circulated, for instance, as converted into digital-typed, secure electronic records or reports, which have high commercial value, e-tickets and so on. However, because those digital contents have commercial value, high-level security should be required for delivery between a consumer and a provider with non face-to-face method in online environment. As a digital contents, an e-ticket is a sort of electronic certificate to assure ticket-holder's proprietary rights of a real ticket. This paper focuses on e-ticket as a typical digital contents which has real commercial value. For secure delivery and use of digital contents in on/off environment, this paper proposes that 1) how to generate e-tickets in a remote e-ticket server, 2) how to authenticate a user and a smart card holding e-tickets for delivery in online environment, 3) how to save an e-ticket transferred through network into a smart card, 4) how to issue and authenticate e-tickets in offline, and 5) how to collect and discard outdated or used e-tickets.

초 록

개방형 네트워크인 인터넷의 확산과 웹(Web) 기술이 발전함에 따라 다양한 형태의 디지털 콘텐츠가 생성, 응용되고 있다. 이와 같은 디지털 콘텐츠는 디지털 정보 자체가 재화적(財貨的) 가치를 가짐으로 인하여 개방형 네트워크 상에서 소비자와 제공자와의 비대면(Non Face-to-Face) 방식으로 전달하기 위해서는 고도의 보안성이 요구되어진다. 이와 같은 재화적 가치 또는 보안성이 요구되는 콘텐츠로서 내용의 공개가 어려운 전자기록물, 경제적 가치가 높은 보고서, 재화적 가치를 내용에 포함하고 있는 전자티켓 등이 있다. 전자티켓은 디지털 콘텐츠의 일종으로서 전자티켓(Electronic Ticket)은 티켓 소지자의 소유권을 보장하는 전자적 인증서(Electronic Certificate)이다. 위에서 언급한 디지털 콘텐츠 중에서 본 논문에서는 실질적인 재화적 가치를 포함하고 있는 디지털 콘텐츠로서 전자티켓을 중심으로 인터넷 온-라인 환경에서의 안전한 전자티켓의 생성과 전달 그리고 스마트 카드를 이용한 오프-라인에서의 안전한 사용을 위해 1) 전자티켓 서버에서의 티켓 생성; 2) 온라인 환경에서의 전자티켓의 전달을 위한 사용자 및 카드 인증; 3) 네트워크를 통하여 전달된 전자티켓의 스마트 카드로의 저장; 4) 오프라인에서의 스마트 카드에 저장된 전자티켓의 정당성 인증 및 발권; 5) 사용된 전자티켓의 수집 및 폐기 등을 제안한다.

Keywords: e-ticket, electronic certificate, anonimity, integrity, digital cash
전자티켓, 전자인증서, 익명성, 무결성, 전자화폐

---

* This paper was supported by research funds of ChonBuk National University in 2008.
** Associate professor, Dept. of Library and Information Science, ChonBuk National Univ (yk9118@chonbuk.ac.kr)
*** Senior Researcher, KT Service Incubation Office(hama1@kt.com)
▪ Received : 11 May 2009 ▪ Revised : 15 May 2009 ▪ Accepted : 29 May 2009
▪ Journal of the Korean Society for Information Management, 26(2): 27-41, 2009.
〔DOI:10.3743/KOSIM. 2009.26.2.027〕

1. Introduction

With explosive growth in the area of the Internet and IT services, various types of digital contents are generated and circulated, for instance, as converted into digital-typed, secure electronic records or reports, which have high commercial value, e-tickets and so on. However, because those digital contents have commercial value, high-level security should be required for delivery between a seller and a buyer with non face-to-face method in online environment. As a digital contents, an e-ticket is a sort of electronic certificate to assure ticket-holder's proprietary rights of a real ticket. Libraries and information centers should consider how to control and manage those contents. Because it is possible to generate and duplicate digital goods, perfect protection or management technologies such as DRM (Digital Rights Management), access control, usage control, and contents control are required for its circulation on open networks. Especially, some of digitalized information sources have high economic value or secure documental content which should be securely distributed or circulated. Therefore, those contents should be securely controlled. We propose a protection mechanism for secure generation and circulation of e-tickets, which is one of the digital contents mentioned above, in online environment.

Conceptually, electronic tickets or e-tickets are the digital counterpart of paper tickets such as movie or airplane tickets. Generally, an e-ticket system consists of ticket agents, service providers, and customers.

A ticket agent can sell an e-ticket to a customer (ticket issuance). The customer can take it to a service provider and get the services indicated in the e-ticket (ticket consumption). The e-ticket can also be transferred between customers (ticket transfer). When the e-ticket is used up, it will be returned to the ticket agent to be voided (ticket collection). It is obvious from <Figure 1> that for the secure circulation of e-tickets, the e-ticket system should be able to provide four cryptographic protocols to accomplish the above four types of transactions, i.e., ticket issuance, ticket consumption, ticket transfer, and ticket collection.

The circulation steps can be similar in the area of electronic documents which includes administrative policies, diplomatic agreements and so on. That is, e-documents have a lifecycle such as production, usage, collection, preservation, and so on. As mentioned above, there are so many security issues relating digital contents including e-document, e-ticket, e-journal and so on.

There are several useful properties that one might expect from e-tickets. Among others, we will address four fundamental properties, which are anonymity, revocable anonymity, divisibility, and transferability(Fujimura et. al 1999). Anonymity means that in the course of e-ticket issuance or consumption, the identity of an e-ticket holder should not be traceable by anyone. On the contrary, revocable anonymity means that if the e-ticket holder overspends his e-ticket, then his identity will be revealed, that is to say, his anonymity should

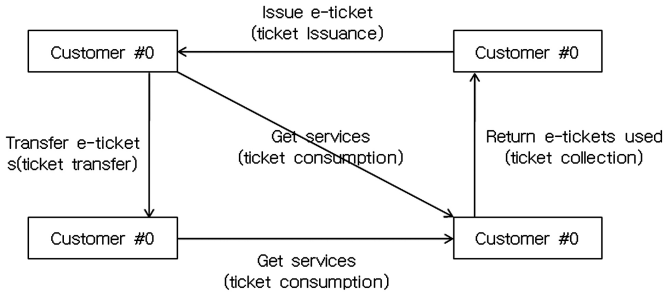
be revocable. That an e-ticket is divisible implies that it can be used in some specified number of times. Finally, a transferable e-ticket can be transferred to other customers off-line without contacting a central authority such as ticket agents.

Indeed, active research on e-tickets has begun since Fujimura-Nakajima (1998) proposed a general framework of e-tickets in late 1990s. A number of different e-ticket methods have since been published in the literature. However, most of them are abstract in the description of the technical details of e-ticket mechanism or are limited in the provision of major e-ticket functions.

To overcome those problems, Quercia-Hailes (2005) recently presented a general purpose e-ticket scheme with anonymity, revocable anonymity, and divisibility properties. Although their scheme supports many good features, one critical drawback is that it does not support transferability property. Therefore, arises a strong demand to extend Quercia-Hailes' method to include transferability property.

In order to add transferability into Quercia-Hailes' scheme, we examine some of transferable digital cash techniques, because there has been little work on transferable e-tickets. Chen-Pedersen (1995) proposed a transferable digital cash by using dummy blind signatures. Each time a digital cash is transferred, one dummy blind signature is inserted into the digital cash. Instead of dummy blind signatures, Pagnia-Jansen (1997) suggested the use of anonymous public keys. Motivated by the fact that group signatures are anonymous, Jeong, Lee and Lim(2001) also presented another transferable digital cash mechanism based on group signatures. Recently, Saxena, Soh and Zantidis (2005) introduced a notion of additive NIZK (Non-Interactive Zero Knowledge) and used it to propose an off-line transferable digital cash.

Among the above methods, we choose Saxena-Soh-Zantidis' approach for the realization of transferability upon Quercia-Hailes' scheme, because it is easier to implement and requires less communication overhead than any other methods. Therefore, in this paper, we aim to seamlessly and securely integrate both Quercia-Hailes and Saxena- Soh-Zantidis' schemes and propose a new e-ticket scheme that provides transferability as well as ano-



〈Figure 1〉 Typical circulation model of e-tickets

nymity, revocable anonymity, and divisibility.

This paper is organized as follows. Section 2 describes some preliminaries related to this paper. Section 3 gives a detailed explanation to the new e-ticket scheme proposed in this paper. Section 4 provides the requirement analysis of the proposed e-ticket scheme. Lastly, Section 5 concludes this paper with a short remark.

2. Preliminaries

2.1 Blind Signature

A blind signature is a protocol in which a signer signs a message without seeing the message.

Brands' scheme (1994) is based on Schnorr signature scheme, whereas Chaum's scheme (1985) is based on RSA public-key cryptography. We briefly describe Chaum's blind signature scheme here. Assume that Alice wants Bob to sign a message 'm' blindly and that Bob has a private key 'x', a public key 'y', and a public modulus 'n'. First, Alice chooses a random value 'b', called a blinding factor. She blinds 'm' by computing $t^x = mb^y$. Bob signs 't' by computing $t^x = (mb^y)^x \bmod n$. Now Alice unblinds t^x by computing $s = t^x / b \bmod n = m^x \bmod n$. The resulting message, $s = m^x \bmod n$, is the blind signature by Bob on 'm'.

2.2 Quercia-Hailes's Approach

In the e-ticket scheme suggested by Quercia and Hailes, a customer splits his identity string into two different parts in such a way that one part alone does not give any information about the identity, but the combined use of two parts can reveal the identity. The customer prepares a 2D array of such pairs. A ticket agent then uses Chaum's cut-and-choose protocol (1989) to blindly signs the 2D array, thereby generating the 2D array I_{bs} containing blindly signed identity strings. The array I_{bs} is actually embedded into an e-ticket and plays a crucial role in providing three important features of Quercia-Hailes' scheme, namely, anonymity, revocable anonymity and divisibility properties.

Anonymity: Customer identity information in the array I_{bs} is not known to ticket agents or service providers by virtue of the blind signature property, which means that customer anonymity is always preserved.

Revocable anonymity: In the case of overspending, an overspender can be identified by collecting the parts of the split identity string from two different service providers and combining them together. Therefore, the anonymity of dishonest customers can be revocable .

Divisibility: Whenever the customer spends the e-ticket with a service provider, one column of the array I_{bs} is used to verify his identity. So, the number of permissible uses of the e-ticket is determined by the number of columns in the array I_{bs}.

2.3 Saxena-Soh-Zantidis' Approach

2.3.1 Additive NIZK Proof System

In an NIZK proof system, a prover convinces a verifier of the validity of a statement in a non-interactive way. To retain the zero knowledge property, most NIZK proof systems require a random reference string that is shared between the prover and the verifier. On the contrary, an identity based NIZK proof system, in which the identity of the prover is included in the statement, does not need such a common random string, since the proof is verifier-independent.

An additive NIZK proof system, an extension of the identity based NIZK system, enables a verifier to build a new identity based NIZK proof from an old proof. This proof building process can proceed many times, resulting in a long chain of trust relationships between the participants. For example, suppose that Alice knows a secret 'u'. If so, she can create an identity based NIZK proof p_A for the statement s_A = Alice knows the value of u," and can send $\{s_A, p_A\}$ to Bob. Then Bob verifies the proof and builds a new identity based NIZK proof p_B from p_A for the statement s_B= "Bob know that the statement s_A is true." Now he can keep p_A secret and send $\{s_A, s_B, p_B\}$ to the next recipient.

2.3.2 Associative One-Way Function

AOWF (Associative One-Way Function) was first introduced by Sherman and has been used in digital signatures and key agreement(1997).

Let $M = \{0,1\}^n$ be a message space for some positive integer 'n'. An AOWF on M is any binary function $o : M \times M \rightarrow M$ that is associative and one-way. The function o is associative if $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in M$. The function $\circ$ is one-way if o is easy to compute but hard to invert. Besides, The function $\circ$ is strongly non-invertible if the function o is difficult to invert, even if either one of its inputs is given. The function $\circ$ is non-commutative if $aob \neq boa$. The function $\circ$ is called a strong non-commutative AOWF if The function $\circ$ is associative, one-way, strongly non-invertible, and non-commutative.

2.3.3 Digital Cash Protocol

Saxena, Soh, and Zantidis' transferable digital cash is based on the additive NIZK proof system. Assume that the first customer C_o withdraws a digital coin, which has a secret value that is known only to C_o. To transfer the digital coin to the next customer C_1, C_o\$ has to prove that he is the legitimate owner of the coin. Thus, C_o generates an identity based NIZK proof about the coin's secret value by using a strong non-commutative AOWF, and presents the proof to C_1. If C_1 wants to transfer this coin to C_2, then C_1 generates a new NIZK proof that he is the current owner of the digital coin, and sends the resulting proof to C_2. In this manner, a customer C_i can transfer the digital coin to the next customer C_{i+1} securely.

3. Proposed Scheme

3.1 Overview

To understand a complicated system, it is important to grasp some underlying ideas of the system. So, before we look into the details of the proposed e-ticket scheme, we briefly outline some important aspects of it below.

Each customer creates a vector of his identity strings similarly as in Quercia-Hailes' method, and also makes a private key and a matching public key. The customer keeps the private key, but sends the identity vector and the public key to a ticket agent. The ticket agent then blindly signs both the identity vector and the public key, and returns them to the customer. Using the blindly signed identity vector, the customer can obtain e-tickets from ticket agents or spend e-tickets with some service providers anonymously.With a pair of the private key and the blindly signed public key, the customer can sign a message in an anonymous way.

The blindly signed identity vector is verified via a well-known challenge-response protocol. To begin with, the customer sends the ticket agent the identity vector as a witness. On receiving the identity vector, the ticket agent creates a random binary string as a challenge, and sends it to the client. According to the random binary string, the client computes a response about his identity and returns it to the ticket agent. The ticket agent then computes a check value by using the customer's response and verifies whether the computed check value is equal to the check value derived from the received identity vector.

Customer identities are blindly embedded into e-tickets in order to identify overspenders. Specifically, if a customer overspends an e-ticket with more than one service provider, more than one pair of a random binary vector and its response are reported to the ticket agent. Since it is very likely that any two random binary vectors have different binary values in at least one position, the ticket agent is able to obtain two different responses about the overspender's identity. These two different responses are actually used to identify the overspender, because the customer identity is split into two parts in a way that one part alone doest not disclose the identity, but two parts together can reveal the identity.

In preparation for ticket transfer, the first customer C_0, who purchases an e-ticket, creates a zero-knowledge signature z_0 about the purchased e-ticket by using a strong non-commutative AOWF function $\circ$. C_0 also uses the function o to compute r_0 which contains information about the next intended recipient C_1. C_0 now sends z_0, r_0 to C_1. Based on the received information, C_1 verifies if C_0 is a valid owner of the e-ticket and if C_0 himself is designated as the new ticket holder. If the e-ticket is transferred again from C_1 to C_2, then C_1 computes a new zero-knowledge signature z_1 by using the old signature z_0, and computes r_1 as well. C_1 keeps z_0 secret (as an evidence) and sends $\{z_1, r_0, r_1\}$ to C_2. Similar process can continue as many times

as the e-ticket transfer happens.

Regarding the internal structure of the e-ticket used in this paper, it consists of several components, including a certificate signed by a ticket agent, a list of services requested, a value used to check if i-th spending request is valid, a list of anonymous public keys, a list of e-ticket holders, and some cryptic values for zero-knowledge proof.

Some important symbols and notations used throughout this paper are defined as follows.

- TA, SP_i, C_i: a ticket agent, i-th service provider, and i-th customer, respectively.
- x_i, y_i: signer $i's$ private key and public key, respectively.
- $BS_i(m)$: blind signature by signer i on a given message m.
- $E_{key}(m), D_{key}(m)$: encryption and decryption of a given message m by a key key, respectively.
- $Sign_i(m)$: digital signature over a given message m by the private key x_i of signer i, that is, $Sign_i(m) = E_x(h(m))$, where $h(\bullet)$ is a hash function.
- $\circ$: strong non-commutative AOWF.
- $\oplus$: bitwise XOR operator.

In general, an e-ticket system should provide the four cryptographic protocols for ticket issuance, ticket consumption, ticket transfer, and ticket collection. Now we will give a detailed description of how each of the four protocol works in the proposed scheme.

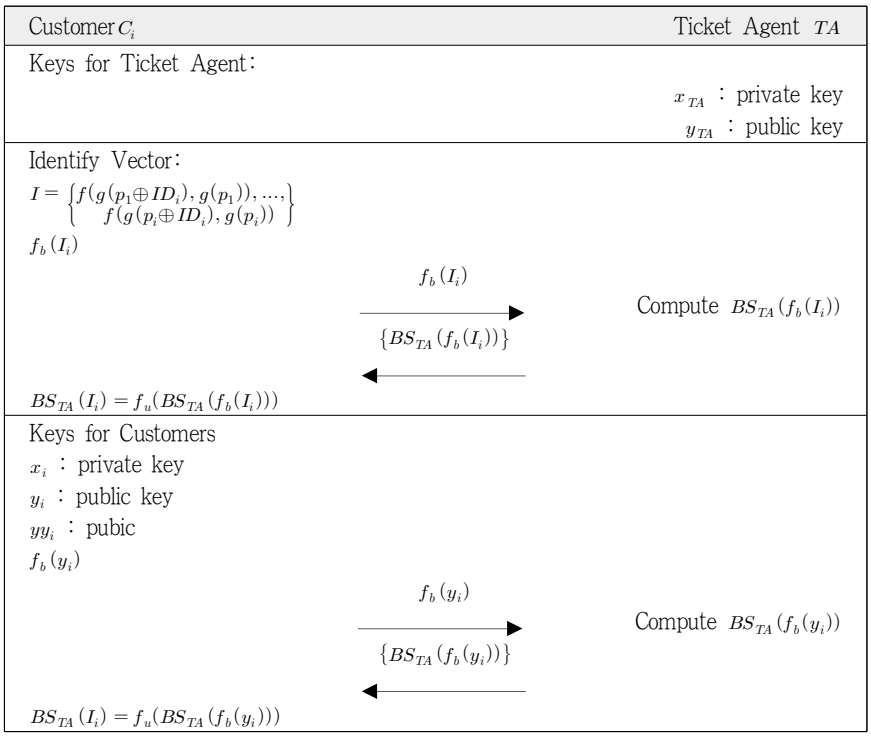
3.2 Initial Registration

There are three entities in our e-ticket system: a ticket agent TA, service providers SP_i and customers C_i. In this initial setup, our e-ticket system produces keys for TA and SP_i, and creates keys and identity vectors for customers C_i as follows (see <Figure 2>).

TA creates a private key x_{TA} and a public key y_{TA}. Similarly, each service provider SP_i creates a private key x_{SP_i} and a public key y_{SP_i}.

Each customer C_i divides his identity string into two parts such as, $ID_i = (p \oplus ID_i, p)$, where ID_i is $C_i's$ identity string and p is a random number. Each part on its own is not meaningful, but when the two parts are combined together, they reveal $C_i's$ identity. Each part can be further hidden with a commitment scheme, e.g., $g(p \oplus ID_i), g(p))$, where g is a hash function. These two commitments are given to a binary hash function f to produce a single value $f(g(p \oplus ID_i), g(p))$. C_i prepares a vector I_i of k such values, $I_i = f(g(p_1 \oplus ID_i), g(p_1)), \ldots, f(g(p_k \oplus ID_i), g(p_k))$ Now is the time to start a blind signature procedure. To begin with, C_i blinds I_i with a blinding function f_b and sends $f_b(I_i)$ to TA. Then TA blindly signs $f_b(I_i)$ through some interactions with C_i and returns $BS_{TA}(f_b(I_i))$ to C_i. At last, C_i unblinds $BS_{TA}(f_b(I_i))$ with an unblinding function f_u and obtains the blindly signed identity vector $BS_{TA}(I_i) = f_u(BS_{TA}(f_b(I_i)))$.

C_i makes a private key x_i and a public key y_i. C_i gets the blindly signed public key $BS_{TA}(y_i)$ via the same procedure as above, and computes



〈Figure 2〉 Initial process for setting up some keys and identify vectors for a ticket agent TA and a customer C_i

a value $yy_i = x_i \circ y_i$ which will be used later in our e-ticket transfer protocol. C_i then makes yy_i.

3.3 E-ticket Issuance

In this stage, the customer C_0, who is the first e-ticket holder, interacts with TA and purchases an e-ticket from TA. To provide the e-ticket with the divisibility property, we use the notion of hash chain. A hash chain consists of a series of elements $\{chain_0, chain_1, \ldots, chain_n\}$, satisfying the following properties:

Initially, $chain_n$, called the root of the hash chain, is set to be $h(seed)$, where h is a hash function and $(seed)$ is a random number.

Two adjacent elements $chain_{j-1}$ and $chain_j$ are related by $chain_{j-1} = h(chain_j)$, $j \in [1, n]$.

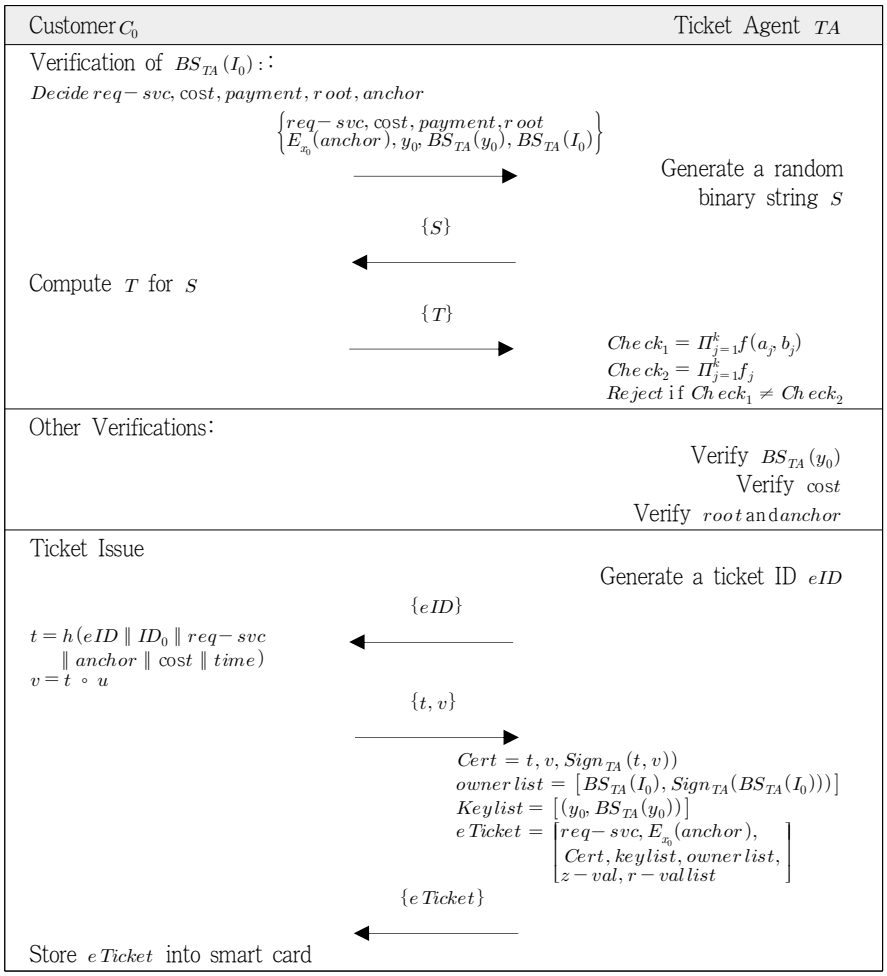
The anchor of the hash chain, $chain_0$, is computed by recursively applying the above relationship. Therefore, the whole hash chain can be completely constructed with the knowledge of the root.

In our scheme, one chain element should be revealed each time a customer uses an e-ticket. For example, assume that the current e-ticket holder reveals $chain_{I}$ at the $i-th$ use of the e-ticket. If he wants to use the e-ticket one more time, then he should be able to reveal the next chain element $chain_{I+1}$. The validity of $chain_{I+1}$ is easily

verified by checking if $(I+1)$ applications of h to $chain_{I+1}$ are equal to the anchor $chain_0$.

Note, however, that it is infeasible to compute $chain_{I+1}$ with the knowledge of $chain_I$ due to the nature of the hash function h. Now we describe the detailed procedure for e-ticket issuance (see <Figure 3>).

C_0 constructs a hash chain and gets $E_{x0}(anchor)$ by encrypting its anchor, called $anchor$, with his private key x_0. C_0 also makes a list of services that he wants to access, which is denoted as $req-svc$. Now C_0 prepares a message $\{req-svc, \mathrm{cost}, payment, root, E_{x0(anchor)}, y_0, BS_{TA}(y_0)BS_{TA}(I_0)\}, C_0$, where cost is an e-ticket price, $payment$ retains some payment information, and $root$ denotes the root of the hash chain. C_0 sends the message to TA.



〈Figure 3〉 e-Ticket issue protocol between a ticket agent TA and the first e-ticket holder C_0

Remember that the length of the vector $BS_{TA}(I_o)$ is k. To check the validity of $BS_{TA}(I_o)$, TA generates a random binary string s of length k and sends it to C_0 as a challenge.

- For each binary value s_j in $S(j \in [1,k])$, C_0 yields a pair of values T_j:

$$T_j = (p_j, g(p_j \oplus ID_o)) \text{ if } s_j = 0$$
$$(p_j \oplus ID_o, g(p_j)) \text{ if } s_j = 1$$

- C_0 sends the resulting pairs $[T_1, \cdots, T_k]$ to TA as the response to s.
- TA now starts to verify the validity of $BS_{TA}(I_o)$ as follows.
- Using $[T_1, \cdots, T_k]$, compute a_j and b_j for all $j \in [1,k]$.

$$a_j = T_{j,2}, b_j = g(T_{j,1}) \text{ if } s_j = 0$$
$$a_j = g(T_{j,1}), b_j = T_{j,2}) \text{ if } s_j = 1$$

where $T_{j,1}$ and $T_{j,2}$ denote the first and the second element of T_j, respectively. If all $T_1, \cdots, T_k$ are valid, then a_j is supposed to equal $g(p_j \oplus ID_o)$ and b_j is supposed to equal $g(p_j)$, irrespective of the value of s_j. Compute a check value $check_1 = \Pi^{k_{j=1}} f(a_j, b_j)$.

Let $f_j = f(g(p_j \oplus ID_0), g(p_j))$. In terms of f_j, $BS_{TA}(I_o)$ can be represented as $BS_{TA}(I_o) = [E_{xTA}(f_1), \cdots, E_{xTA}(f_k)]$. Now compute another check value from $BS_{TA}(I_o)$, that is, $check_2 = \Pi^{k_{j=1}} D_{yTA}(E_{xTA}(f_j)) = \Pi^{k_{j=1}}(f_j)$. If $check_1$ and $check_2$ are equal, then C_0's $BS_{TA}(I_o)$ is proven to be valid. Otherwise, terminate the current ticket issuance procedure.

Subsequently, TA verifies whether the blindly signed public key $BS_{TA}(y_o)$, which is equivalent to $E_{xTA}(y_o)$, is valid by using the received y_o. TA also checks if $cost$ is as much as the total cost of services requested in $req-svc$. Further, it checks if $anchor$ is correctly derived from $root$ within some specific number of applications of the hash function h. After successfully finishing the above verification steps, TA generates a unique e-ticket ID eID and sends it to C_0.

In preparation for e-ticket transfer, C_0 produces the following information:

Using eID together with other information, compute a message digest t about the e-ticket to be newly issued, that is, $t = h(eID \| ID_0 \| req-svc \| anchor \| cost \| time)$, where $\|$ is a string concatenation operator and $time$ is the current time.

Generate a random secret signing key u, which will be known only to C_0, and derive a public verification key $v = to\, u$ keeps u secret and sends $\{t, v\}$ to TA.

On receiving $\{t, v\}$, TA carries out some final tasks as follows:

Generate a certificate $Cert$ to bind the ticket digest t to its verification key v. Thus, $Cert$ will include information like $(t, v, Sign_{TA}(t, v))$.

Make an e-ticket owner list called $Owner list$ and put a pair of C_0's identity vector and its signature into $Owner list$, so $Owner list = [BS_{TA}(I_0), Sign_{TA}(BS_{TA}(I_0))]$. This list will continue to grow as a new owner appears as a result of e-ticket transfer.

Make an anonymous public key list called $Key list$ and initialize it as $Key list = [(y_0, BS_{TA}(y_0)]$.

Allocate the storages for two values $z-val$ and $r-val-list$. These two values together with $Cert$ and $Owner list$ will play an important role during e-ticket transfer.

Finally, create an e-ticket $eTicket$ including the following components, C_0 the amount of $Cert$ according to the information in $payment$, and sends $eTicket$ to C_0.

$$eTicket = \begin{bmatrix} req-svc, E_{x0}(anchor), \\ Cert, key-list, owner-list, \\ z-val, r-val-list \end{bmatrix}$$

Upon receiving $eTicket$, C_0 stores it into his smart card.

3.4 E-ticket Consumption

Any customer C_i who presently holds $eTicket$ can bring his smart card to a service provider SP_j and spend $eTicket$ there. Before rendering the services, SP_j should verify if $eTicket$ is genuine and legitimate. The whole process of e-ticket consumption proceeds as follows.

C_i sends SP_j the message $\{eTicket, np, chain_{np}\}$, np is the next non-spent position in the hash chain and $chain_{np}$ is the corresponding hash chain element.

Upon receiving the message, SP_j performs three verification procedures:

Verify whether the certificate $Cert$ in $eTicket$ is valid.

Verify whether $chain_{np}$ is correct. In other words, check if np applications of the hash function h to $chain_{np}$ is equal to the hash chain anchor stored in $eTicket$. Actually, the anchor can be computed by decrypting $E_{x0}(anchor)$ by the public key y_0.

Verify the identity of the current e-ticket holder C_i. $eTicket$'s identity can be found at the end of $Ownerlist$. Specifically, if the identity is $(BS_{TA}(I_i), Sign_{C_{i-1}}(BS_{TA}(I_i))$, SP_j checks the genuineness of the signature part $Sign_{C_{i-1}}(BS_{TA}(I_i))$, as well as the validity of the identity vector $BS_{TA}(I_i)$ by following the same procedure in steps 2, 3, and 4 of the previous e-ticket issuance protocol.

If C_i passes all the above verifications, then he will be granted the services specified in the list $req-svc$. Finally, in case that overspending may happen, SP_j sends TA all the transactional data including $e-Ticket, np, chain_{np}, S_{np}$, and $(T_1, ..., T_k)$, where S_{np} is the random binary string at np-level, $(T_1, ..., T_k)_{np}$ is the response to S_{np}. Note that $\$S_{np}$ and $(T_1, ..., T_k)$ are both generated in the course of verifying $BS_{TA}(I_i)$ in the previous step.

3.5 E-ticket Transfer

In our scheme, $eTicket$ can be transferred many times to other customers without contacting TA or central authority. Assume that $eTicket$ has been transferred among customers $(C_1, ..., C_k)$ in this order. Our protocol to transfer $eTicket$ from the customer C_i to the next customer C_{i+1} proceeds as follows (see <figure 4>).

C_i, the current e-ticket holder, computes two values z_i and r_i:

$$z_i = ((z_{i-1} \circ y_{i-1}) \circ x_i \quad (1)$$
$$r_i = ((t \circ yy_{i+1}) \circ x_i \quad (2)$$

where z_{i-1} is read from $z-val$, y_{i-1} is read from $keylist$, and t is extracted from $cert$. In case of C_0, z_0 is computed by $z_0 = u \circ x_0$, where u is the secret signing key generated in step 7 of

our e-ticket issuance protocol. Conceptually, z_i signifies the zero-knowledge signature of C_i on $eTicket$, whereas r_i holds information that the next intended recipient is C_{i+1}.

Now C_i prepares a message $\{Cert, keylist, onerlist, z_i, (r-val-list, r_i)\}$, where $r-val-list$ carries the values $(r_0, ..., r_{i-1})$. C_i sends the message to C_{i+1}. Upon receiving the message, C_{i+1} executes the verification procedure that consists of the following tests:

Verify the certificate $Cert$, and get t and v from $Cert$.

Verify if each pair $(y_j, BS_{TA}(y_j))$ in $keylist$ is valid for all $j \in [0, i]$ by using the fact that $BS_{TA}(y_j)$ is equivalent to $E_{x\,TA}(y_j)$

Check if $((t \circ yy_{j+1}) \circ yy_j) = (r_j \circ y_j)$, for all $j \in [0, i]$

Each element in $ownerlist$ has the form of $(BS_{TA}(I_j), Sign_{C_{j-1}}(BS_{TA}(I_j)))$ for all $j \in [0, i]$. Verify if all those elements are valid by using the keys in $keylist$.

If all the tests succeed, then C_{i+1} sends his identity vector $BS_{TA}(I_{i+1})$ and the anonymous public key $(y_{i+1}, BS_{TA}(y_{j+1}))$ to C_i.

Finally, C_i performs the following.

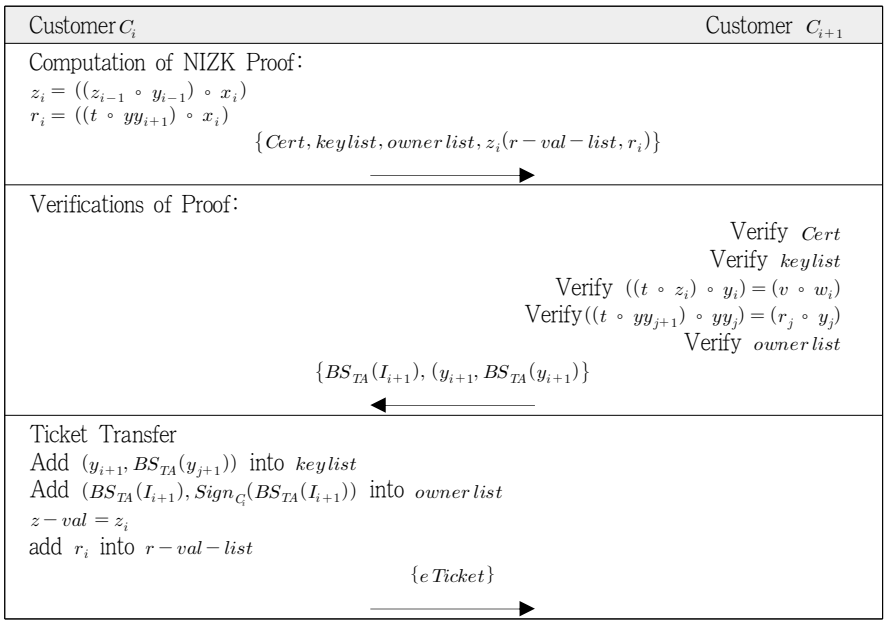
Put the received public key $(y_{i+1}, BS_{TA}(y_{j+1}))$ into the end of $keylist$.

Sign the received identity vector and appends the vector-signature pair $(BS_{TA}(I_{i+1}), Sign_{C_i}(BS_{TA}(I_{i+1}))$ into $ownerlist$.

Replace $z-val$ with z_i.

Add r_i into the end of $r-val-list$.

Now, $eTicket$ and its related resources in C_i's smart card are transferred to C_{i+1}'s smart card by using a ticket transfer protocol such as TTP (Ticket Transfer Protocol).



〈Figure 4〉 A protocol to transfer e-ticket from the customer to the next customer

3.6 E-ticket Collection

$eTicket$, when it has reached its maximum limit (e.g., the maximum number of uses), will be collected by TA. At this point, TA can check for the misuse of $eTicket$ by examining all the transaction data saved in the database.

For example, assume that a particular customer C_d double-uses $eTicket$ with two different service providers SP_a and SP_b. According to our e-ticket consumption protocol, SP_a then generates a binary random string S_a of length k and receives its response $T_a = [T_1^a, T_2^a, ..., T_k^a]$ from C_d. Likewise, SP_b has two quantities S_b and $T_b = [T_1^b, T_2^b, ..., T_k^b]$.

The double spending of $eTicket$ can then be detected by the fact that two binary random strings S_a and S_b are very likely to have different binary values in at least one position. For instance, if $S_a = (..., 1, ...)$ and $S_b = (..., 0, ...)$ have different values at the $j-th$ position, then $T_j^a = (p_j \oplus ID_d, g(p_j))$ and $T_j^b = (p_j, g(p_j \oplus ID_d))$. Therefore, TA can compute C_d's identity by $T_{j,0}^a \oplus T_{j,0}^b = (p_j \oplus ID_d) \oplus (p_j)) = ID_d$.

4. Requirements Analysis

As mentioned in Section 1, there are four useful requirements for e-tickets, which are anonymity (untraceability for honest users), revocable anonymity (traceability for overspenders), divisibility, and transferability. In this section, we analyze the proposed e-ticket scheme from the viewpoint of these four requirements.

Anonymity: In the initial registration phase, each customer C_i acquires a blindly signed identity vec-

tor $BS_{TA}(I_i)$ and a blindly signed public key $BS_{TA}(y_i)$ from a ticket aget TA. C_i uses them to purchase e-tickets anonymously, spend e-tickets with service providers anonymously, or transfer e-tickets to other customers anonymously. In other words, during the above all actions, C_i's identity cannot be discovered by anyone, even TA, due to the nature of a blind signature. The blind signature is anonymous, so all e-ticket related actions are anonymous, which means that the proposed scheme is anonymous as well.

Revocable anonymity: In our e-ticket scheme, the identity string ID_i of the customer C_i is a secret. This secret string is carefully divided into two parts such that one part alone does not disclose the secret, but the combination of two parts can reveal the secret. This technique is usually called secret splitting. As shown in e-ticket collection phase, our scheme can employ this secret splitting technique to detect the overspenders and reveal the identities of those dishonest customers. However, our scheme does not provide any mechanisms to figure out the identities of honest customers.

Divisibility: In our scheme, the divisibility function is achieved by virtue of a hash chain. The number of times an e-ticket can be used is determined by the length of the hash chain. For example, the hash chain of length n means that the e-ticket can be used only n times. So, one non-spent chain element of the hash chain is presented to a service provider whenever a customer spends his e-ticket with the service provider. The service provider can easily check the validity of the presented chain element by using the anchor of the hash chain that is securely stored within the e-ticket.

Transferability: In Equation (1), $z_0 = u \circ x_0$ indicates the zero-knowledge signature of the first e-ticket holder C_0 on the purchased e-ticket $eTicket$. z_0 cannot be computed without the knowledge of the secret signing key u and the private key x_0, both of which are only known to C_0. The next e-ticket holder C_1 can add more information to z_0 by computing $z_1 = ((z_0 \circ y_0) \circ x_1$ in such a way that z_1 still retains the zero-knowledge. Therefore, each z_1 is an NIZK proof that only C_0 knows u and information was added i times to z_0. This additive property makes $eTicket$ transferable virtually unlimited number of times. In the e-ticket transfer protocol, Step 2-c is necessary to check that this additive proof is correctly constructed. Meanwhile, r_i in Equation (2) contains information on the next intended user of $eTicket$. Step 2-d of the e-ticket transfer protocol is necessary to check that all those r_i are correctly computed according to the order of e-ticket holders appearing in $ownerlist$. The proofs of two equations in Steps 2-c and 2-d are shown below.

Proof of

$$((t \circ z_i) \circ y_i) = (v \circ w_i) : t \circ z_i \circ y_i :$$

$$= t \circ (z_{i-1} \circ y_{i-1} \circ x_i) \circ y_i$$

$$= t \circ (z_{i-1} \circ y_{i-1}) \circ yy_i$$

$$= t \circ (z_0 \circ y_0) \circ yy_1 \circ yy_2 \circ \cdots \circ yy_i$$

$$= t \circ (u_0 \circ x_0 \circ y_0) \circ yy_1 \circ yy_2 \circ \cdots \circ yy_i$$

$$= (t \circ u) \circ yy_1 \circ yy_2 \circ \cdots \circ yy_i$$

$$= (t \circ u) \circ w_i$$

$$= v \circ w_i$$

Proof of

$$((t \circ yy_{j+1}) \circ yy_j) = (r_j \circ y_j) : t \circ yy_{j+1} \circ yy_j :$$
$$= t \circ yy_{j+1} \circ (x_j \circ y_j)$$
$$= (t \circ yy_{j+1} \circ x_j) \circ y_j)$$
$$= r_j \circ y_j$$

5. Conclusion

With increasing interests in e-tickets, various e-ticket schemes have been developed. However, most of them are abstract in the description of the technical details of e-ticket mechanismor are limited in the provision of major e-ticket functions. To overcome those drawbacks, Quercia and Hailes recently presented an e-ticket protocol that supports anonymity, revocable anonymity, and divisibility properties, but not transferability property. Thereforer, in this paper, we extend Quercia-Hailes' protocol and propose a new e-ticket scheme that provides transferability as well as anonymity, revocable anonymity, and divisibility. Actually we adapt Saxena-Soh-Zantidis' digital cash scheme to provide the transferability property. The requirement analysis shows that our e-ticket scheme supports the above four e-ticket properties.

References

Brands, S. 1994. "Untraceable offline cash in wallet with observers," *LNCS*, 773: 302-318.

Chaum, D. 1985. "Security without identification: transaction systems to make big bother obsolete," *Communications of the ACM*, 28(10): 1030-1044.

Chaum, D., A. Fiat, and M Naor. 1989. "Untraceable electronic cash," *Proceedings of 8th Annual International Cryptology Conference on Advances in Cryptology,* 403: 319-327.

Chen, L. and T. Pedersen. 1995. "New group signature schemes," *LNCS*, 950: 171-181.

Fujimura, K. and Y. Nakajima. 1998. "General-purpose digital ticket framework," *Proceedings of 3rd USENIX Workshop on Electronic Commerce*, 177-186.

Jeong, I., D. Lee, and J. Lim. 2001. "Efficient transferable cash with group signatures," *LNCS*, 2200: 462-474.

Fujimura, K., Hiroshi Kuno, Masayuki Terada, Kanzuo Matsuyama, Yasunao Mizuno, and Jun Sekine. 1999. Digital-Ticket-Controlled Digital Ticket Circulation," *Proceedings of the 8th USENIX Security Symposium*, 8.

Kuramitsu, K., T. Murakami, H. Matsuda, and K. Sajamura. 2000. "TTP: secure ACID transfer protocol for electronic ticket between personal tamper-proof devices," *Proceeding of 24th Annual International Computer Software and Applications Conference*, 87-92.

Pagnia, H. and Jansen, R. 1997. “Towards multiple-payment schemes for digital money,” *LNCS*, vol.1318, 203-216.

Quercia, D. and S. Hailes. 2005. “MOTET: mobile transactions using Electronic Tickets,” *Proceedings of 1st IEEE International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 374-383.

Rabi, M. and A. Sherman. 1997. “An observation on associative one-way functions in complexity theory,” *Information Processing Letters*, 64: 239-244.

Saxena, A., B. Soh, and D. Zantidis. 2005. “A digital cash protocol based on additive zero knowledge,” *LNCS*, 3482: 672-680.