Application of Electronic Ticket to Online Trading With Smart Card Technology

Shirley Siu Weng In Guo Zhen Sheng
fstwis@umac.mo fstzsg@umac.mo
Faculty of Science and Technology
University of Macau

Abstract

The concept of electronic ticket is an abstraction of customer rights or services purchased. It is presented as the certification of ownership, which can be issued over the Internet from seller to buyer in the electronic marketplace. Online delivery is a new type of demanded service of E-Commerce, which requires secure communication among merchant, customers and service providers. This paper proposes an electronic ticket management architecture with an electronic ticket structure design, secure ticketing communication protocol, and an electronic ticket wallet using smart card technology. This paper also presents design and implementation details of the electronic ticket management architecture.

1. Introduction

With the ever growing of E-Commerce in today's business, demand on quality of online services is getting more restricted. Online delivery of purchased items is one of them. Due to the physical constraints, purchased items are usually delivered to the customer's site by traditional mail. However, it has been overlooked that the rights of accessing certain kinds of services, such as event, transportation, theaters, membership services that are usually represented by cards or tickets, can actually be encoded and delivered electronically over the Internet. The electronic ticket is a certificate of customer rights or services, which guarantees that the ticket owner has the right to claim the services written on the ticket [1]. Just as its name implies, the media for storing ticket data should not be a piece of paper but some electronic means, that should meet the requirements of secure, portable and durable etc. Smart cards are ideal media for electronic tickets. Both ticket purchase and ticket consumption process can be done via a smart card read-write device (known as Card Accepting Device, CAD).

The digital ticket concept in the area of E-Commerce was first proposed by K. Fuijimura and Y. Nakajima in [1]. Their work laid the foundation of the so-called digital rights, its data schema and processing architecture. Similar applications were also emerged such as Internet coupons [2], software licenses distribution over the Internet [3], etc. Much simpler form of e-ticket services emerge recently, but the ticket is never delivered to customer over the Internet, instead it is stored in central servers. Customer is requested to show identification documents when using the service purchased online.

This paper describes the trading mechanism using electronic tickets in the Internet, the problems encountered, and the approaches to solving them by adopting secure communication protocols and smart card technology. The system that proposed here is the Electronic Ticket Management System (ETMS). The prototype has been implemented that demonstrates feasibility and efficiency of the concepts.

2. Applying Electronic Tickets to E-Commerce

2.1 Definition of Electronic Ticket

Electronic ticket is a piece of data showing that the user is entitled to certain rights. The ticket must be readable and writeable electronically. The ticket may have associated with it a number of conditions, such as validity period, quality of service, just as real world tickets do.

2.2 Trading Mechanism using Electronic Ticket

The basic service trading mechanism consists of three phases: purchase transaction, ticket issuance and ticket consumption, as shown in Figure 1. A trading scenario involves at least two participants: merchant (the service seller) and customer (the service buyer). Most of the time, merchant is also the one who provides the traded services to the customer. However, for businesses that involve intermediaries such as selling agents and retailers, the intermediaries would probably be the service providers. To be more general, the role of ticket issuance (merchant) is separated from the service-providing deputy (service provider). Client-server architecture is adopted here, the entity that handles processing and communications in merchant side acting as server and the one in customer side acting as client.

Stage 1. Purchase and payment transaction

In the first stage, the customer selects the target items and sends the purchase request to the merchant. If the server requires the client to be authenticated, it can either accept the client’s certificate as valid or have it validated by its Certification Authority (CA). Similarly, if the client requires the server’s public key certificate then it can be supplied signed by a CA trusted by the client. If payment were required, after both parties agreed the deal, the customer would be requested to pay by some suitable payment methods [5] (e.g. electronic cash, electronic check or secure credit card transaction). The online payment issues are out of the scope of the paper.

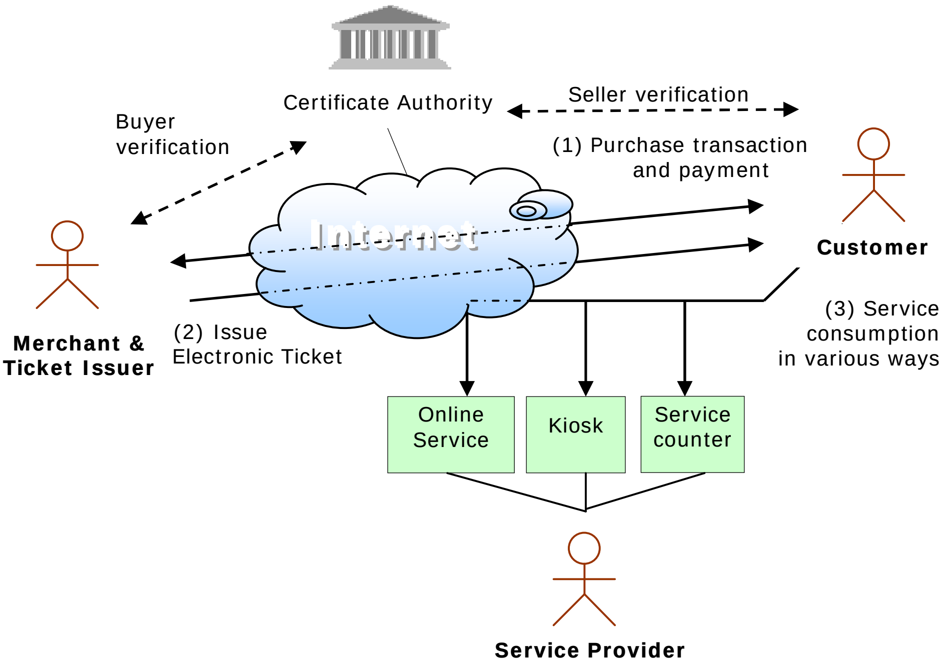


Figure 1. Basic service trading mechanism using Electronic Ticket

Stage 2. Ticket issuance

The merchant composes a ticket from the chosen service and handed it over to the customer. In the server side, a new ticket which contains the service details and most importantly the certificate of the right of ownership, is generated. To ensure the ticket arrives intact, a secure session is established. Then the ticket is encrypted, signed by the merchant and sent over the secure channel. In the client side, the received ticket shall be decrypted, verified and assured of its integrity. Then, the entire ticket should be stored safely. The medium of storage that is proposed here is smart card.

Stage 3. Ticket consumption

As soon as the ticket is issued, the customer becomes the owner of the ticket. When the traded service is being accessed, the ticket would be presented for both verification and validation. Usually, the service provider can provide services in several ways: online services to be accessed over the Internet or offline services to be accessed from the specially made kiosks, storefronts, service counters, entrance gates, etc.

2.3 Requirements of Electronic Ticket and the Management System

Electronic ticket is an abstraction of rights and values, the design of the ticket structure should be general such that it can be applied to many E-Commerce applications. It should also be portable for user to access both online and offline services easily, because some of the services are still required the physical access of customers. Depending on the types of ticket, the electronic ticket should support both applications that require anonymity and non-anonymity. It is expected to be more durable than traditional ticket because the ticket information can always be updated electronically. Most important of all, the electronic ticket should be secure in terms of integrity and non-repudiation to protect the rights of merchant and customer. The duplicated ticket must be detected and denied access to services, while the original ticket should still be effective. In order to detect fraud ticket right away, some secret information should be used to construct a valid ticket.

Aimed to meet the requirements mentioned, the proposed Electronic Ticket Management System (ETMS) defines a generic structure of ticket, which supports both online and offline services, provides anonymity and non-anonymity features and reflects ticket consumption state. Security strategies are designed into the ticket itself, the ticket storage (so called ticket wallet) and the communication protocols such that the tickets are guaranteed to be genuine (i.e. not a forgery) and original (i.e. not a duplicate).

3. The ETMS Approach

3.1 Data Model of Electronic Ticket

The electronic ticket is defined as the 5-tuple of **(M, S, P, C, L)**. The **merchant (M)** issues and underwrites this electronic ticket. The ticket carries the rights of using certain kinds of **service (S)** provided by the authorized **service provider (P)**. The service component includes all the information about the service purchased such as value, valid period, quality of service and general information about the service. **Customer (C)** is the holder of the ticket and also the buyer of the service. As decided by the merchant, each ticket should be specified with the usage restrictions, known as **properties list (L)**, e.g. anonymity, transferability, etc. The ticket must be correctly validated and digitally signed by the merchant:

$$\mathbf{ET = Sign_M(M, S, P, C, L)}$$

When the ticket is signed, the integrity and non-repudiation properties of the ticket are assured.

3.2 Trusted Computing Base for Ticket Wallet

3.2.1 Smart Card Technology

The smart card, an intelligent token, is a credit card sized plastic card embedded with an integrated circuit chip. It provides not only memory capacity, but computational capability as well [8]. Smart card is a tamper-resistant device for its highly secure feature. It offers facility to process cryptographic operations entirely inside the card. Data in smart card can be protected with the PIN (or password), biometrics locks and electronic digital signature [6]. Other benefits of smart cards are small size, low cost, portable and more durable that makes it possible for individual to carry certain amount of data in their wallets [7]. All these features and strengths make the smart card the ideal container of electronic tickets.

3.2.2 Design of the Secure Ticket Wallet

The smart card has been abstracted as a ticket wallet, in which many kinds of ticket are stored. As shown in Figure 2, there are two separate programs, Ticket Wallet program for ticket operations and Private Data Locker program to store customer information. Four types of data are stored in the smart card: the electronic tickets, personal information of the customer, information for access authentication and ticket operation's credentials.

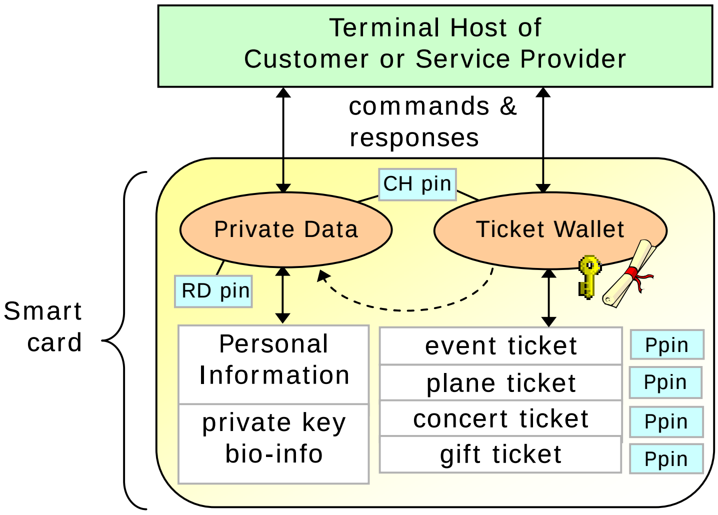


Figure 2. Design of Ticket Wallet

3.2.2.1 Ticket Wallet

The Ticket Wallet program receives operation requests and ticket information from the host. It could read, write, modify or remove tickets in the card. In addition, each of them possesses keys and credentials, which uniquely identifies the wallet in the globe. Non-trivial operations that have been completed inside the Ticket Wallet can be manifested to the requestor by signing the result of operation and sending it out.

Primary data in the wallet is the electronic tickets. Multiple tickets can be stored in one card at the same time. Each ticket entry is the realization of ticket data model mentioned in Section 3.1. Since the electronic ticket is officially signed by the merchant, modifying the ticket would automatically invalidate it. To make the ticket state changeable, a consumption index (CI) represents the current consumable value of the ticket is attached to each ticket entry.

To access Ticket Wallet, one must be authenticated to it by giving access PIN. CH pin is the cardholder access PIN. Whoever accesses the wallet using this PIN is considered as the card owner. Hence, he is allowed to do most operations such as list, view, download and delete electronic tickets from the wallet, with exception to modify and consume the ticket. Because each electronic ticket is issued and digitally signed by the merchant, any unauthorized modifications to the ticket would immediately invalidate it. To consume the ticket, service provider who owns the Service Provider PIN (Ppin) of the corresponding ticket could modify its current value by delta-amount.

3.2.2.2 Private Data Locker

The Private Data Locker program is responsible for managing customer’s sensitive information. Operations include add, delete and modify personal data, generate cipher keys, digitally signing data, modify access permissions of those data, etc. Each entry of personal information is classified into two access levels. General personal information like name, age, place of birth, etc. could be managed by customer himself. To read the information only, the Read PIN (RD pin) is required. Highly confidential information such as private key, biometric data, cannot be accessed directly and arbitrarily by anyone, even the cardholder. But only the cardholder can invoke operations on them, such as initialize cryptographic key-pair, generate digital signature, verification of biometric information, etc.

3.3 Secure Ticket Operations

To ensure the secure communication between merchant and the ticket wallet holding by customer, a trusted third party "Card Issuer" is introduced. It guarantees both merchant and ticket wallet the genuineness of one another over the Internet, without the requirement of exposing customer's identity. In essence, secure ticket operations required the use of cryptography to generate valid ticket, checking integrity of ticket after transmission over public network, ensure ticket information privacy and to prove operations inside ticket wallet are completed without falsification. In this section, some abbreviations are used: I for Card Issuer, M for Merchant, C for Customer, P for Service Provider and TW for Ticket Wallet. $\mathrm{Enc}_{\mathrm{Ykey}}\{X\}$ represents message X being encrypted by Ykey, $\mathrm{Sign}_{\mathrm{Ykey}}\{X\}$represents message X is digitally signed with Ykey. Important data flows between interacting parties and operations[1] are illustrated in figures.

3.3.1 Initialization of Ticket Wallet: Card Issuer ⇔ Ticket Wallet

Programs in ticket wallet handle ticket operations and credentials represent the unique identity of the wallet. They must be installed and initialized by card issuer. For later validation, information of the wallet is stored in a repository kept by Card Issuer.

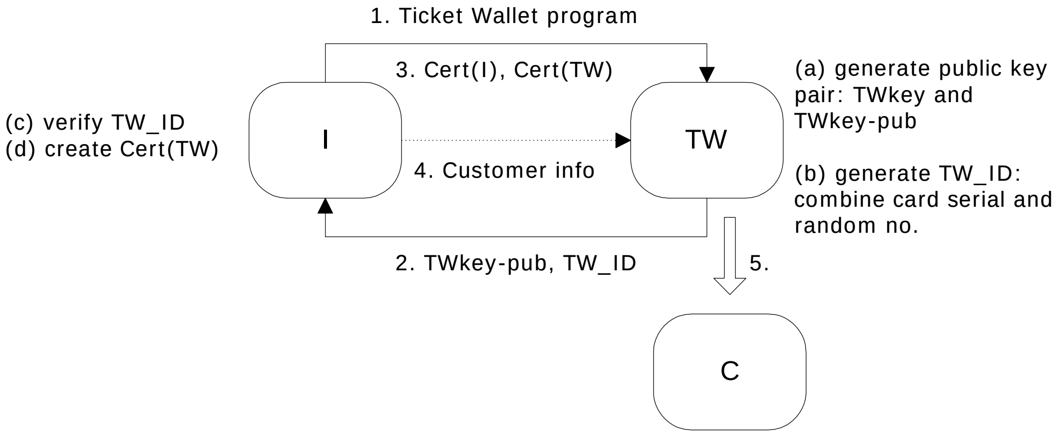


Figure 3. Initialization of ticket wallet

Ticket wallet programs are first sent to the smart card for installation. Installed wallet program is then requested to generate a public key pair (the private key TWkey and public key TWkey-pub) and an identity number (TW_ID). To guarantee the uniqueness of TW_ID, it can be created by combining the card serial number and a random number. After verification by card issuer, a digital certificate of ticket wallet (Cert(TW))is created. This certificate binds the owner of TWkey to TW_ID, and signed by I. Hence, application that trusts I can safely trust the ticket wallet and the operation done by it. Moreover, certificate of I should also be kept by the wallet to achieve two-way authentication. After initialization is completed, the ticket wallet should contain this information: TWkey, Cert(TW) and Cert(I).

[1] Other operations such as ticket transfer and lost ticket management can be referred to [9].

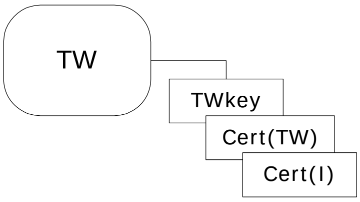


Figure 4. Key and credentials stored in ticket wallet

3.3.2 Ticket Issuance: Merchant ⇔ Customer ⇔ Ticket Wallet

After purchase and payment transaction, both merchant and customer keep a transaction identity number (TID), which acts as an evidence to the online paid transaction. Using this TID, customer can claim for ticket issuance from merchant immediately. Since the issuance procedure is taken place over the Internet, a secure session has to be established between merchant and customer. Messages are encrypted with the shared session key known to both of them; a message digest or hash is also attached for checking integrity.

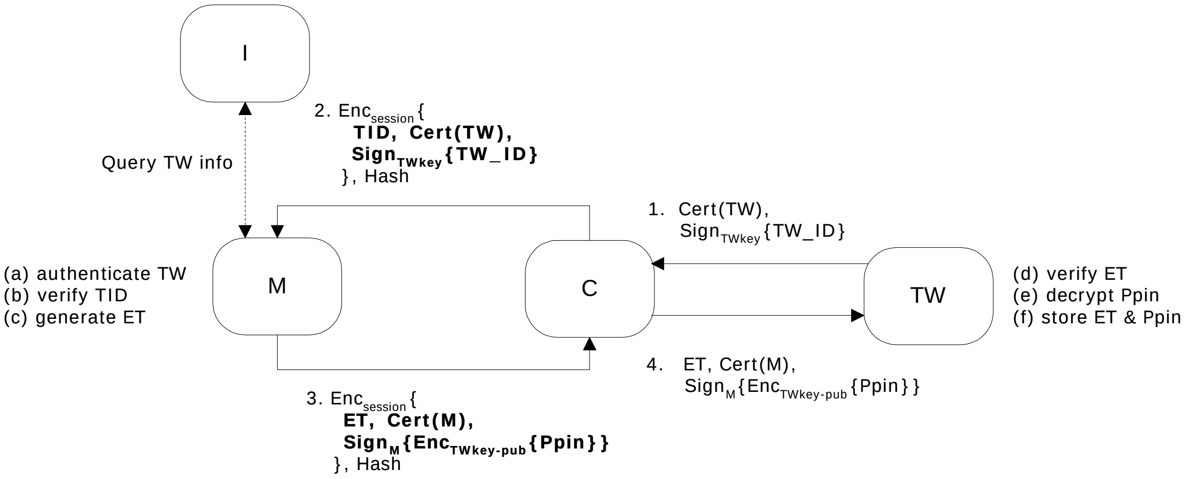


Figure 5. Ticket issuance over Internet

The issuance procedure is as follows: Customer retrieves the wallet certificate and the signed TW_ID from the wallet, together with the TID sent to the merchant. The purpose of including the signed TW_ID is to manifest to merchant that this wallet is the one claimed in the certificate. Merchant has to verify the certificate, which may optionally invoke queries to card issuer. Then, it must also verify the transaction order, service availability, issuance policy etc. Finally, the electronic ticket that binds to TW_ID is generated. This prevents abuse of the ticket in other wallets, except the real one involved in the issuance transaction, because wallet will only accept electronic tickets issued to itself. To enable access of ticket by service provider, access PIN is also supplied and securely encrypted with the wallet's public key. At the other end, customer who acts as an intermediary, only needs to test the integrity of the whole message, and passes the core of it to the wallet. If the ticket is verified successfully, it is stored and the transaction completes.

3.3.3 Ticket Consumption: Ticket Wallet ⇔ Service Provider

Since the connection between the ticket wallet and service provider is direct in offline consumption, the environment is assumed to be secure. Only some verification is carried out:

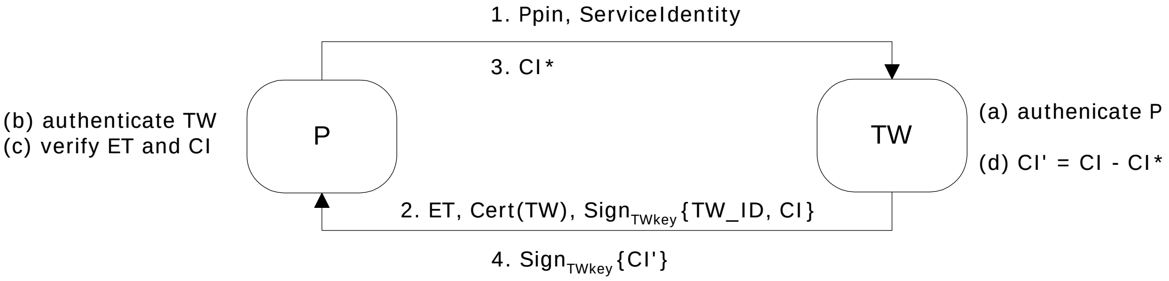


Figure 6. Ticket consumption in secure Service Provider environment

Service provider sends its PIN and service identity to the connected ticket wallet. After authenticated, ticket wallet replies with the electronic ticket that matches the service offered by this service provider and its current consumable value (CI), etc. Ticket verifications by service provider involve verifying merchant's digital signature on ET, matching of TW_ID in ET and the TW_ID indicated in wallet certificate, the ticket information, valid period, etc. If they are correct and the remaining amount is enough, service provider calculates the consumption amount and sends the increment or decrement delta value (CI*) to the wallet. Finally, ticket wallet increases or decreases values from CI and replied the operation result.

When ticket is consumed online, not only two-way authentication is necessary, service provider has also to supply its Ppin to ticket wallet in order to modify the current CI. In Figure 7 (i), after authenticating TW, P signs the ServiceIdentity and the encrypted Ppin that are sent to TW to retrieve the matching ET. In Figure 7 (ii), TW generates a timestamped-random number encrypted with P's public key. This number identifies the one-time consumption request that will be issued by P in next message. The random number, the matching ET and the signed CI are then returned to P. Then, P generates a consumption request consists of CI* and the random number.

TW should only consume the ticket once per such request. Replaying the same request will not cause any effect in the wallet. Finally in Figure 7 (iii), the result is sent to P as confirmation.

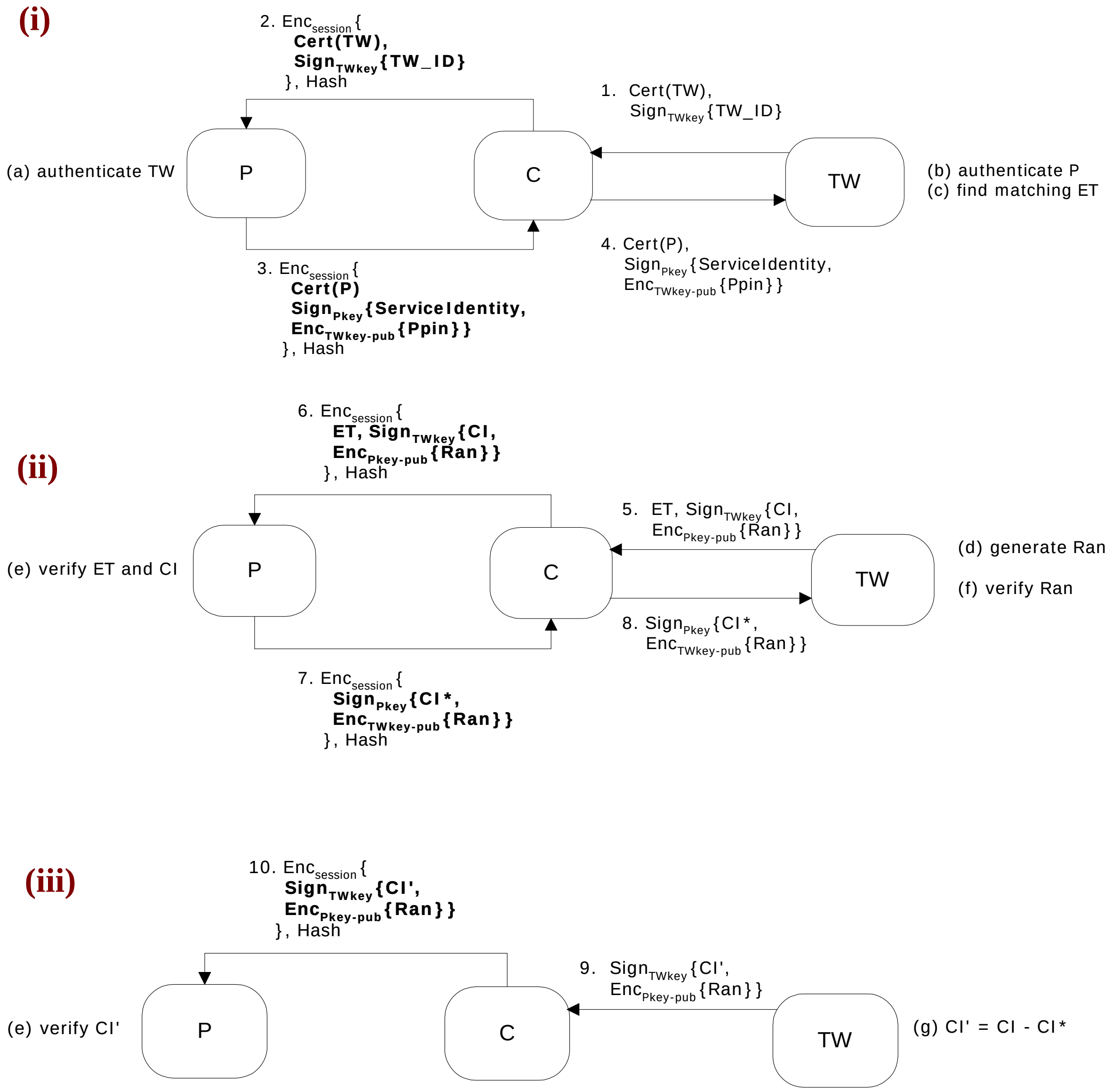

(i)
2. $\mathrm{Enc}_{session}$ { Cert(TW), Sign_{TWkey} {TW_ID} }, Hash
1. Cert(TW), Sign_{TWkey} {TW_ID}
(a) authenticate TW
P
C
TW
(b) authenticate P
(c) find matching ET
4. Cert(P), Sign_{Pkey} {ServiceIdentity, $\mathrm{Enc}_{TWkey\text{-}pub}$ {Ppin} }
3. $\mathrm{Enc}_{session}$ { Cert(P) Sign_{Pkey} {ServiceIdentity, $\mathrm{Enc}_{TWkey\text{-}pub}$ {Ppin} } }, Hash
(ii)
6. $\mathrm{Enc}_{session}$ { ET, Sign_{TWkey} {CI, $\mathrm{Enc}_{Pkey\text{-}pub}$ {Ran} } }, Hash
5. ET, Sign_{TWkey} {CI, $\mathrm{Enc}_{Pkey\text{-}pub}$ {Ran} }
(e) verify ET and CI
P
C
TW
(d) generate Ran
(f) verify Ran
8. Sign_{Pkey} {CI*, $\mathrm{Enc}_{TWkey\text{-}pub}$ {Ran} }
7. $\mathrm{Enc}_{session}$ { Sign_{Pkey} {CI*, $\mathrm{Enc}_{TWkey\text{-}pub}$ {Ran} } }, Hash
(iii)
10. $\mathrm{Enc}_{session}$ { Sign_{TWkey} {CI', $\mathrm{Enc}_{Pkey\text{-}pub}$ {Ran} } }, Hash
9. Sign_{TWkey} {CI', $\mathrm{Enc}_{Pkey\text{-}pub}$ {Ran} }
(e) verify CI'
P
C
TW
(g) CI' = CI - CI*

Figure 7. Online ticket consumption: (i) authentication (ii) request (iii) response

4. Implementation Issues

4.1 System Architecture

ETMS is divided into four components: Card Issuer Subsystem, Merchant Subsystem, Service Provider Subsystem and User Subsystem. Communication between the subsystems is over the public Internet.

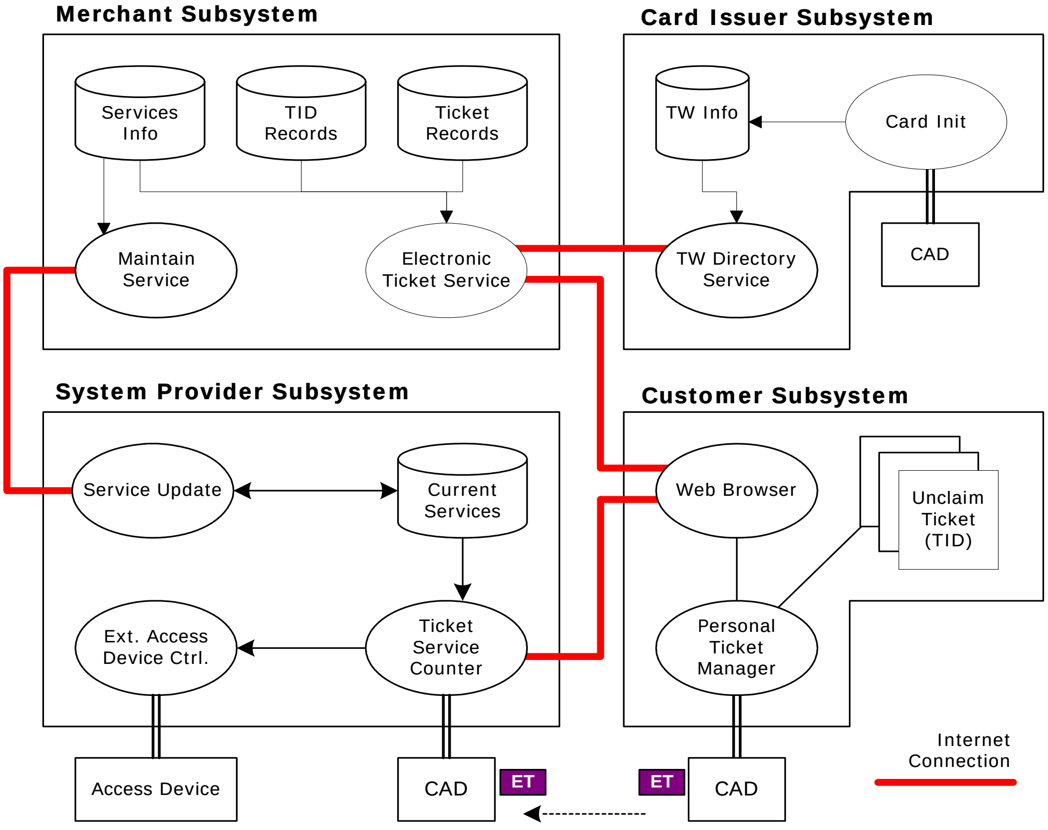


Figure 8. Simplified ETMS system architecture

Card Issuer is the trust authority in ETMS framework. It maintains a Ticket Wallet Directory Service for querying information of ticket wallet issued by it. Query returns the mapping of TW_ID to the public key corresponds to the private key owned by the wallet. In Merchant Subsystem, data repositories for services information, TID records and issued ticket records are provided. Electronic Ticket Service collects all the ticket handling modules – issue ticket, process ticket transfer request, and query information, etc. Since the ticket issuance is completely web-based, it can be seamlessly integrated into existing online shopping procedure. Newly updated service information can be propagated to Service Provider in Maintain Service module. To operate the wallet in customer's site, it is required to install a card reader and obtain the smart card with wallet program from Card Issuer. User interface is browser-based to avoid complex installation procedure, which hindered most of the customers to use new application. Personal Ticket Manager manages unclaimed ticket in local host as well as tickets stored in the wallet. During service consumption, Service Provider examines the ticket in Ticket Service Counter and executes redemption.

4.2 Data Structure of Electronic Ticket

A complete ticket is composed of three parts: header, optional headers and body (see Figure 9), this is in order to achieve the generic design of the ticket. The header provides common information of all tickets. Zero or more optional headers may be included to extend the header by additional features. The body portion contains only business-specific information designed by each merchant. One important design objective of the ticket data structure is to be compact and bit-oriented. It is because the ticket will finally be stored in smart card, in which all data is managed in bits.

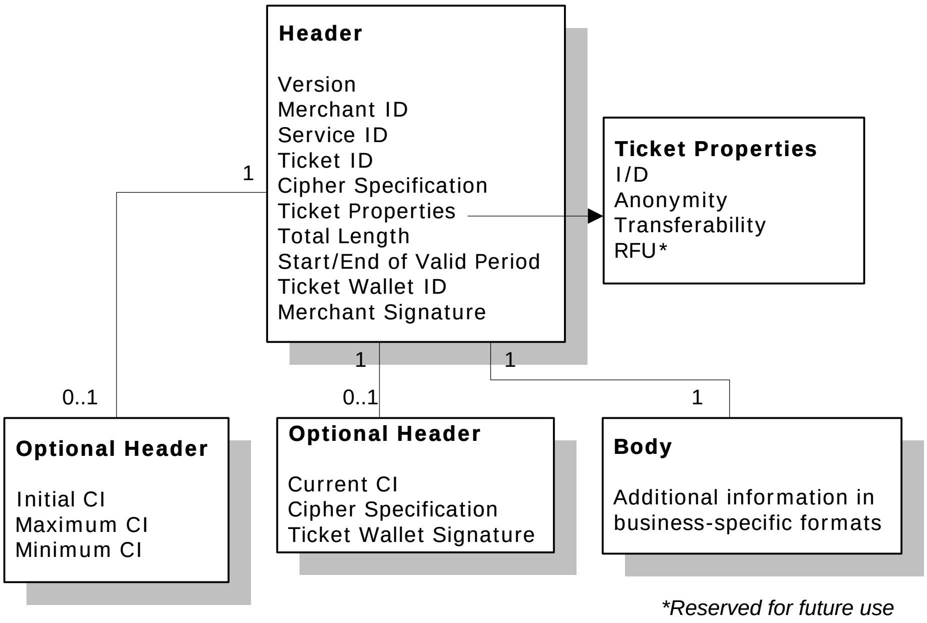


Figure 9. Ticket structure

4.2.1 Default Ticket Header

The ticket is drawn in a way such that each row takes four bytes. The default header consists of 11 fields.

Merchant ID, *Service ID* and *Ticket ID* form the index key of the ticket. *Merchant ID* is assigned by the card issuer when merchant company registered to provide electronic ticket service, while *Service ID* and *Ticket ID* are assigned by the merchant when the ticket is issued. It is merchant’s responsibility to keep the *Service ID* and *Ticket ID* unique. *Cipher Specification* indicates the cryptographic algorithm used to create the digital signature of the ticket. For highly secure ticket, stronger signature algorithm can be chosen and vice versa.

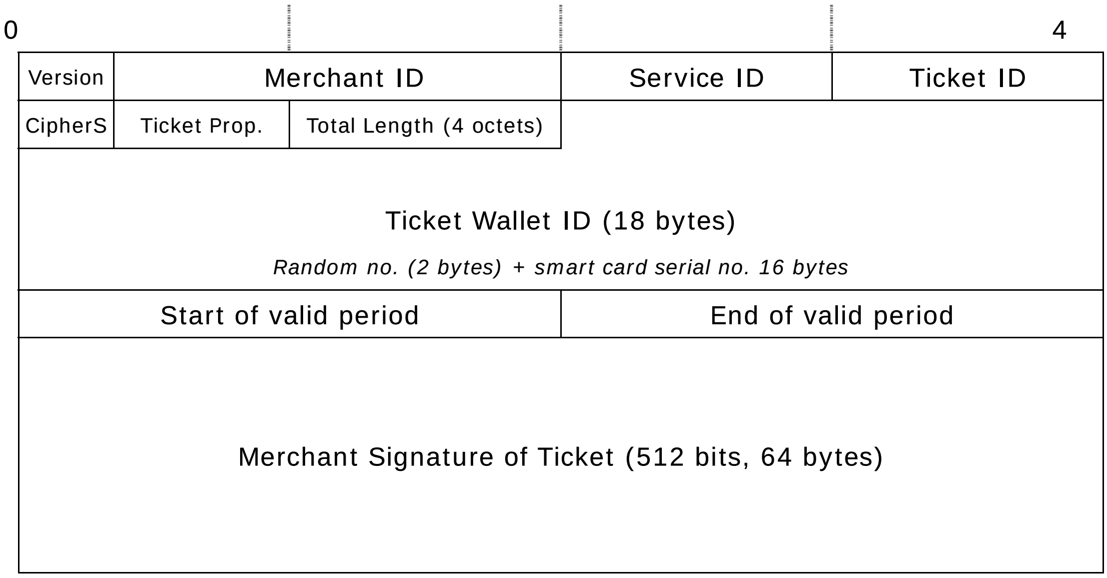

0
4
Version
Merchant ID
Service ID
Ticket ID
CipherS
Ticket Prop.
Total Length (4 octets)
Ticket Wallet ID (18 bytes)
Random no. (2 bytes) + smart card serial no. 16 bytes
Start of valid period
End of valid period
Merchant Signature of Ticket (512 bits, 64 bytes)

Figure 10. Default ticket header format

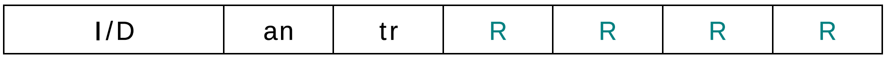


Figure 11. Ticket Properties format

Ticket Properties define common properties of the tickets. Three properties are defined initially: *I/D* is the abbreviation of Increment/Decrement. The ticket value can be either increased, decreased or no change when it is consumed. For example, for traffic ticket, the stored value is always decremented after each ride. On the contrary, the value of bonus-point ticket could be increased after each purchase in a particular shop. In addition, ticket used as membership card will neither add nor remove value from the card. *Anonymity* (*an*) indicates whether customer should be queried for identification when consumption. *Transferability* (*tr*) implies whether the ticket is transferable. The *Length* field contains the total length of the ticket, including header, optional header if present and data. *Ticket Wallet ID (TW_ID)* is the identity number of the wallet in which the ticket will be stored. Therefore, the ticket is bound to one ticket wallet only. For both security and commercial consideration, the validity of ticket is constrained to a period of time. Finally, *Merchant Signature* of the ticket information must be attached for ensuring the integrity of the ticket.

4.2.2 Optional Ticket Headers

For features that are common to some but not all of the tickets, they are put in the optional headers. An optional header is in the format of TLV (Type-Length-Value). *Type* is to identify the information that is carried, S*Len* indicates the length of it, and *Value* contains the information. For example, the optional header in Figure 12 (a) carrying the consumption information contains: the *Initial CI (consumption index)* shows the purchased value of the ticket, *Min/Max CI* governed the maximum and minimum consumption value that can be attained until the ticket must be reloaded. For large optional header that is more than 7 bytes, the length can be encoded in the next byte.

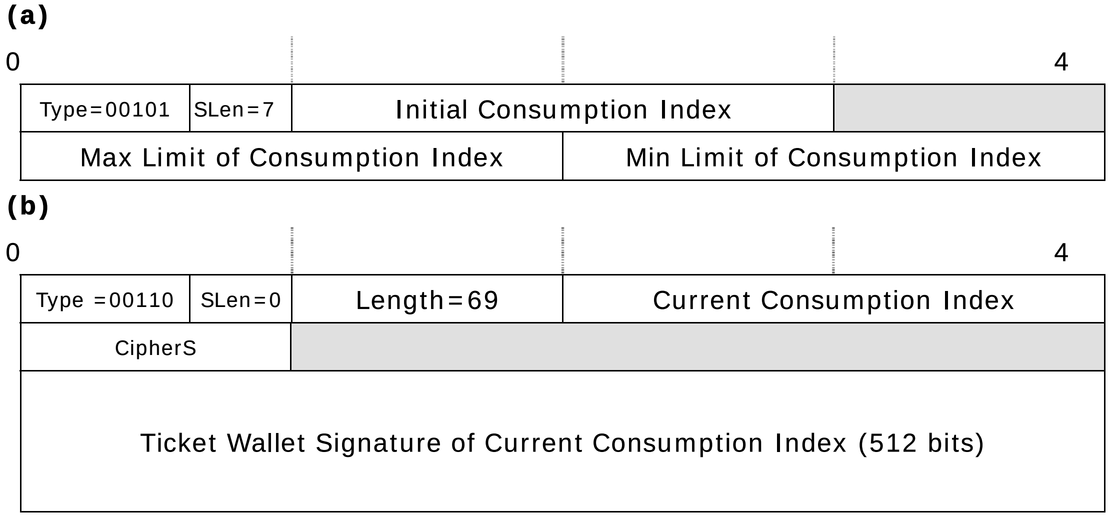

(a)
0
4
Type=00101
SLen=7
Initial Consumption Index
Max Limit of Consumption Index
Min Limit of Consumption Index
(b)
0
4
Type =00110
SLen=0
Length=69
Current Consumption Index
CipherS
Ticket Wallet Signature of Current Consumption Index (512 bits)

Figure 12. Sample optional headers format

4.2.3 Ticket Body

This part contains business-specific information describing the service provided to the ticket owner. To be flexible, the ticket information data structure should be designed by merchants themselves. The recommended format is XML (Extended Markup Language). It supports customized specification of application-specific tags. It is extensible and highly portable. A sample hierarchy is shown below:

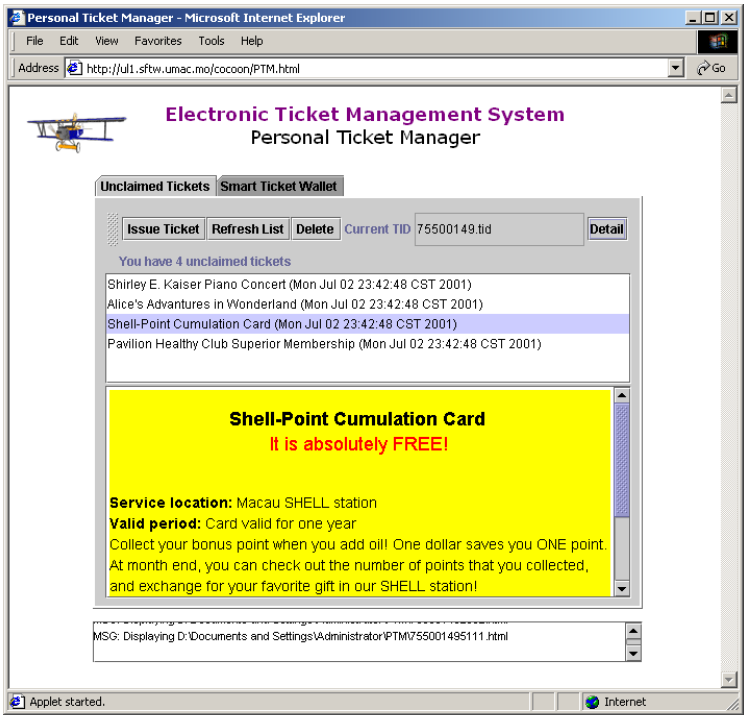
```
<TicketInfo>
    <Merchant>
        <Organizer>Macau Online Ticket Center </Organizer>
        <Contact>+853-5897789</Contact>
        <URL>http://www.ticket-center.mo</URL>
    </Merchant>

    <EventInfo>
```

```
        <Event>International Women Volleyball Match</Event>
        <Program>China vs. Japan</Program>
        <Date>12-07-2001</Date>
        <Time>15:00</Time>
        <Venue>Macau Forum</Venue>
  </EventInfo>
  <PurchaseInfo>
        <Price currency=”MOP”>100</Price>
        <SeatNo>E60</SeatNo>
        <Class>Normal</Class>
  </PurchaseInfo>
</TicketInfo>
```

4.3 Implementation Status

A prototype of ETMS and partial implementation of the ticket operations are being developed. The development language is Java. Merchant server is in a SUN Ultra-1 workstation running Solaris 2.6, while Customer and Service Provider clients are based on Pentium III-800 PC. The ticket wallet is resided in Java Card (one type of smart card) [12]. Digital signature algorithm chosen to use is RSA with MD5, while encryption uses DES. Key length is limited to 512 bits due to Java Card[2] limitation. Communication between client and server is through a normal 56K dial-up connection. Customer is restricted to offline ticket consumption. Card Issuer Subsystem and Private Data Locker are not included in this prototype. Initial evaluation of the system is completed with satisfactory performance. Result of system evaluation is presented in Table 1 to 4. To calculate the timing of each action, it is done by taking the average of 10 trial results. Screen captures of the Customer and Service Provider screens are shown in Figure 13 and Figure 14 respectively.

[2] Gemplus 211/PK International Sample card, with RSA coprocessor


Personal Ticket Manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://ul1.sftw.umac.mo/cocoon/PTM.html
Go
Unclaimed Tickets
Smart Ticket Wallet
Issue Ticket
Refresh List
Delete
Detail
Applet started.
Internet

Electronic Ticket Management System
Personal Ticket Manager

Current TID 75500149.tid

You have 4 unclaimed tickets

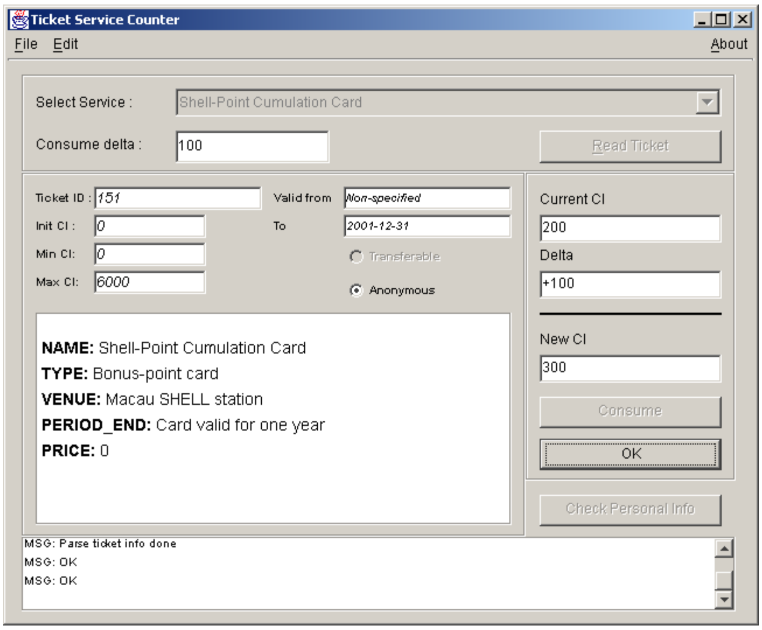
- Shirley E. Kaiser Piano Concert (Mon Jul 02 23:42:48 CST 2001)
- Alice's Advantures in Wonderland (Mon Jul 02 23:42:48 CST 2001)
- Shell-Point Cumulation Card (Mon Jul 02 23:42:48 CST 2001)
- Pavilion Healthy Club Superior Membership (Mon Jul 02 23:42:48 CST 2001)

Shell-Point Cumulation Card
It is absolutely FREE!

Service location: Macau SHELL station
Valid period: Card valid for one year
Collect your bonus point when you add oil! One dollar saves you ONE point.
At month end, you can check out the number of points that you collected, and exchange for your favorite gift in our SHELL station!

```
MSG: Displaying D:\Documents and Settings\Administrator\PTM\755001495111.html
```

Figure 13. User interface for Customer managing unclaimed tickets



Select Service : Shell-Point Cumulation Card
Consume delta : 100

Ticket ID : 151
Init CI : 0
Min CI: 0
Max CI: 6000
Valid from Non-specified
To 2001-12-31
○ Transferable
● Anonymous

Current CI 200
Delta +100
New CI 300

NAME: Shell-Point Cumulation Card
TYPE: Bonus-point card
VENUE: Macau SHELL station
PERIOD_END: Card valid for one year
PRICE: 0

```
MSG: Parse ticket info done
MSG: OK
MSG: OK
```

Figure 14. User Interface for Service Provider execute ticket redemption

Action	Time in sec
Issue Ticket, includes: ● submit TID ● get ticket ● save to wallet	4.5
Retrieve ticket list (10 tickets)	0.9
Retrieve ticket body of one ticket (200 – 300 bytes per ticket)	3
XML-parsing of ticket information	<0.1
Delete a ticket	0.7

Table 3. Customer - Ticket management in Wallet

Action	Time in sec
Retrieve ticket and verify	3.1
Consume ticket and return new CI	0.3

Table 4. Service Provider - Ticket consumption

Action	Time in sec
Wallet loaded to smart card	36
Wallet installed in smart card and initialize wallet key	7

* *Using the Gemplus JCard manager to upload and install.*

Table 1. Card Issuer - Wallet initialization

Action	Time in sec
Ticket generation	0.2

**Digital signature algorithm using RSA with MD5*

Table 2. Merchant - Ticket generation

5. Conclusion

This paper proposes the initial infrastructure of the Electronic Ticket Management System for trading services online. Objective of the system is to improve quality of E-Commerce in particular some service industries, such that the online transaction result can be delivered to users right on the web. This paper discussed the requirements of an electronic ticket and its physical realization with the smart card technology. To ensure security, the ticket operations are carefully designed. Finally, the system architecture is proposed to implement the suggested infrastructure.

To put a final word, E-services have already been booming on the web. The connection between these services with the physical business is always neglected. With the smart card technology, the secure electronic ticket can bridge the gap between them by bringing the online traded data to be used offline.

6. References

[1] K. Fujimura, Y. Nakajima, “**General-purchase Digital Ticket Framework**”, the Proceedings of the 3rd USENIX Workshop on Electronic Commerce, Boston, Massachusetts, August 31, 1998.

[2] M. Kumar, A. Rangachari, A. Jhingran, R. Mohan, “**Sales Promotions on the Internet**”, the Proceedings of the 3rd USENIX Workshop on Electronic Commerce, Boston, Massachusetts, August 31, 1998.

[3] T. Aura, D. Gollman, “**Software License Management with Smart Cards**”, the Proceedings of the USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 10-11, 1999.

[4] B. Patel, J. Crowcroft, “**Ticket Based Service Access for the Mobile User**”, the Proceedings of the third annual ACM/IEEE international conference on Mobile Computing and Networking, Budapest, Hungary, September 26-30, 1997.

[5] N. Asokan, P.A. Janson, M. Steiner and M. Waidner, “**The State of the Art in Electronic Payment Systems**”, IEEE Computer, September 1997, pp.28-35.

[6] D. Youd, "**What is a Digital Signature?**", http://www.youdzone.com/signature.html

[7] T. Kalin, G. Kandus, "**Smart-Card- and IP-Based Infrastructure for a Health-Care Information System in Slovenia**", in Proceedings of the INET Conference, 2000.

[8] "**What is a Smart Card?**", http://java.sun.com/products/javacard/smartcards.html

[9] W. I. Siu, Z. S. Guo, “**The Secure Communication Protocol for Electronic Ticket Management System**”, submitted to 8th Asia-Pacific Software Engineering Conference (APSEC2001).

[10] P. McDaniel, P. Honeyman, A. Prakash, "**Lightweight Secure Group Communication**", CITI Technical Report 98-2.

[11] "**Java Card Technology**", http://java.sun.com/products/javacard/

[12] U. Hansmann, M. S. Nicklous, et al., "**Smart Card Application Development Using Java**," Springer, 2000.