JOBS: Javacard-based Online-ticket Booking System

Derek W.M. Sin and Henry C.B. Chan
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{cswmsin, cshchan}@comp.polyu.edu.hk

ABSTRACT

It is expected that the next generation of smart card will be more intelligent, more interactive and more interoperable. We call these the 3i requirements. A smart card wallet system for meeting these 3i requirements is presented. A novel object-oriented framework is proposed, such that everything is stored in the smart card wallet as an object and the extensible markup language (XML) is employed for storing data and facilitating information exchange. Five types of objects are proposed to cater for different requirements. In particular, users can store an agent in a smart card wallet for retrieving the original object over the Internet. This helps to overcome the memory limitations of smart cards. We also present an application called Javacard-based Online-ticket Booking System (JOBS) for purchasing tickets (e.g., film tickets) over the Internet. A simplified prototype has been built to demonstrate its basic functions.

Keywords

Smart card, Java card, e-commerce, e-wallet, e-payment

1. Introduction

Looking like a portable microcomputer, a smart card is a plastic card embedded with a microprocessor chip [1,2]. Emulating the physical wallets, the focus of this paper is to use smart cards as wallets for storing different types of items. People can store digital certificates, electronic cash, etc. in a smart card wallet and carry it to conduct electronic transactions anywhere in a secure and efficient manner through the Web. Integrated with the smart card wallet, the Web becomes an even more effective tool to support business-to-consumer e-commerce. However, current smart cards have two major limitations. First, the memory of a smart card is very limited. Second, smart cards produced by different manufacturers are generally not inter-operable. In this paper, we propose a novel framework of a smart card wallet to address these two issues. To extend the memory of a smart card, we propose that for some objects, an agent of the original object is stored inside a smart card whereas the actual object is stored somewhere on the Internet. Whenever the actual object is required, the agent retrieves the object over the Internet for the user. To develop an inter-operable smart card wallet, we propose an object-oriented framework, and the Java card [3] is used to realize this framework.

2. Next generation smart card wallet

While current smart cards are "smart", future smart cards will be even "smarter". We think that future smart cards will be more “Intelligent” (e.g. multi-functional), “Interactive” (e.g. able to interact with different devices on the Internet) and “Interoperable” (e.g. written once, run many cards). We call these the 3i requirements. To satisfy the 3i requirements, we propose to use an object-oriented framework such that everything is stored inside a smart card as an object. Fig. 1 shows that objects stored inside a smart card wallet can be classified into three categories, namely transferable objects, non-transferable objects and agents. Transferable objects are objects that can be transferred out of the smart card wallet. These objects can be further divided into duplicable objects and non-duplicable objects. Duplicable objects can be duplicated or copied. For transferable and non-duplicable objects, they are totally removed from the original smart card wallet after being transferred to another smart card wallet.

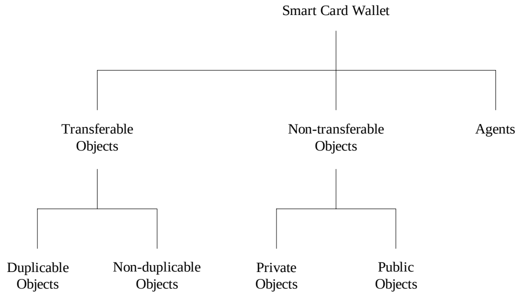

Figure 1: Classification of objects inside a smart card wallet

In contrast to transferable objects are non-transferable objects. As the name implies, these objects cannot be transferred to another smart card wallet. Non-transferable objects can be further divided into public objects and private objects. The third type of object is called an agent. For instance, due to memory constraints, it is impossible to store a large image file in a smart card wallet. However an agent can be stored in the smart card wallet for retrieving the image file over the Internet. This not only overcomes the memory constraint of a smart card wallet but also opens many new applications/services.

Each object has three basic components, namely header, content and methods. The object header and object content are expressed using the XML. The object content is the actual data stored in the object. Finally, the methods allow users/other smart card wallets to operate on the object content. Objects stored inside a smart card wallet are managed by an object manager (OM). The OM itself is a non-transferable and private object inside a smart card wallet. It functions as the interface for accessing the objects stored inside the smart card wallet.

3. Overview of the Javacard-based Online-ticket Booking System (JOBS)

In this section, we apply the smart card wallet to design an application called JOBS for purchasing film tickets online and download them to a smart card wallet. At the cinema entrance, the film ticket is retrieved for gaining access to see the film. The protocol operation is divided into three phases: Authentication/Registration phase, Ticket Purchasing phase and Ticket Retrieval phase. For the first phase, the user first enters his/her password. Notice that the password verification is performed at the client's side by matching the data stored in the smart card wallet. Upon successful verification, the user and the cinema exchange their X.509 certificates and hence the public keys for authentication purpose. After the authentication/registration process, the cinema starts the ticket purchase process by sending the film details to the user. The film details are signed by the cinema's private key to ensure content integrity. The user then selects the film details and sends the "TicketPurchase" message to the cinema. Upon receiving and verifying the message, the cinema sends a digitally signed acknowledgement to the user and the user returns a digitally signed confirmation to the cinema. Then the cinema delivers the digitally signed ticket to the customer using the digital envelope method used in SET [4]. After verifying the digital signature on the ticket, the smart card wallet stores the ticket. Alternatively, the smart card wallet can store the ticket using the agent approach. After receiving the ticket, the customer sends a digitally signed payment instruction to the cinema using the digital envelope method. The payment instruction can be used to initiate a credit card payment. Alternatively, the smart card wallet may pay for the ticket using a micropayment method such as Millicent [5]. In this case, the electronic token is sent to the cinema using the digital envelope method. Finally, the cinema issues a signed receipt to the customer. At the cinema entrance, the customer inserts the smart card wallet into the card reader and the ticket would be presented for the user to gain access to see the film.

4. Prototype System: i-Cinema

To demonstrate the basic function of JOBS, a simplified prototype called i-Cinema has been built.

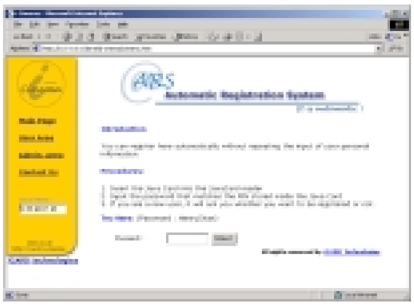

Figure 2: Main page of i-Cinema

The main page of i-Cinema is shown in Fig. 2. To enter the system, a user needs to enter his/her password. User authentication is performed at the client side. Therefore unlike the server-side username/password verification systems, pre-registration with the server-side is not necessary. For a new visitor, the user will be asked whether he/she wants to register with the i-Cinema. If so, the registration process will be done automatically by transferring data to the server's database. After successful login to the i-Cinema system, the customer will be forwarded to the Ticket Purchase screen. Here, he/she can input the film details and then submit the request to the cinema by pressing the button. The cinema then transfers the film ticket to the smart card wallet. For the ticket retrieval process at the cinema entrance, the user inserts his/her smart card wallet into the smart card reader and the system will retrieve the ticket and verify its content against the cinema's database. Finally, the ticket will be deleted from the smart card wallet.

JOBS can also be deployed to facilitate e-retailing in general. With e-commerce, a PULL model is now possible in which consumers can pull the goods down a demand chain with the assistance of an electronic broker (e-broker) [6]. Through the Web, the consumer can explore the goods of different suppliers and place orders accordingly. For each order, the corresponding ticket is downloaded into his/her smart card wallet. The e-broker then notifies the orders to the respective suppliers and the suppliers transport the goods to the respective local distributors. At a local distributor, the consumer inserts his/her smart card wallet into the card reader and obtains the goods based on the tickets stored inside the smart card wallet.

5. Conclusion

We have presented the basic framework of an object-oriented smart card wallet. Five types of objects are proposed and XML is employed to store data and facilitate information exchange. In particular, a user can store an agent inside a smart card wallet for retrieving the original object over the Internet. Integrated with the smart card wallets, the Web becomes an even more effective tool to support B2C e-commerce. An application called JOBS is presented to illustrate how the smart card wallet can be used to purchase and download tickets over the Internet. JOBS can also be deployed to facilitate e-retailing in general in which a consumer can play an active role in pulling goods down a demand chain.

6. ACKNOWLEDGMENTS

This project is supported by an Innovation Technology Fund/Teaching Company Scheme of the Government of the Hong Kong Special Administrative Region under account number UIT/8 and by Macroview Telecom.

7. REFERENCES

[1] W. Rankl and W. Effing, *Smart Card Handbook*, John Wiley & Sons Ltd., Chichester, 1997.

[2] International Organization for Standardization (ISO). Draft International Standard ISO/IEC 7816: Integrated circuit(s) cards with contacts.

[3] JavaCard Forum [http://www.javacard.forum.org]

[4] Secure Electronic Transaction [http://www.setco.org/]

[5] Millicent Protocol for Inexpensive e-commerce [http://www.millicent.com/works/details/papers/millicent-w3c4/millicent.html]

[6] E. Turban, J. Lee, D. King and H. M. Chung, *Electronic commerce: a managerial perspective*, Prentice Hall, 2000.