The Design, Implementation and Evaluation of an Internet Payment System

Term 4 report

prepared by

Chong Ka Lung
(98080070)

Supervisors:

Prof. Michael, R. T. Lyu
Prof. Y. S. Moon

Markers:

Prof. Ada, W. S. Fu
Prof. John, C. S. Lui

April 19, 2000.

Department of Computer Science and Engineering
The Chinese University of Hong Kong

Abstract

With the gaining popularity of electronic commerce nowadays, a secure payment system plays a significant role in the Internet. In this report, we propose an Internet payment system which uses a payment gateway to handle the credit card payment transaction between customers, merchants and banks. To test and evaluate the payment system, we build an online travel agency called TravelNet, which simulate an real-life E-commerce application. On-line travel services including flight reservation, selling of travel accessories, tour guides, and hotel reservation are provided in TravelNet. TravelNet makes use of the proposed payment system to handle the payment transferred between customers and merchants. We implement the payment model as well as TravelNet, and conduct performance evaluation on the payment system. The performance results show that our payment system is easy-to-use, secure, and cost-effective. To improve the creditability of our performance evaluation, SET will be simulated to compare with our performance evaluation on the payment model.

Contents

1. Introduction ____ *4*

1.1 Background ____ 4

1.2 Current Payment Systems ____ 5

2. Payment Model ____ *7*

3. TravelNet ____ *10*

3.1 The architecture ____ 11

3.2 Features ____ 12

4. Implementation of our payment model ____ *12*

4.1 Merchant class ____ 13

4.2 PGate class ____ 13

4.3 Acquirer class ____ 13

4.4 Issuer class ____ 14

5. Simulation ____ *14*

5.1 Customer behaviors ____ 14

5.2 Payment system ____ 14

5.3 Assumptions ____ 15

5.4 Simulation flow ____ 16

6. System Evaluation ____ *16*

6.1 Qualitative Analysis ____ 16

6.2 Performance Measurement ____ 17

7. Future work ____ *20*

8. Conclusions ____ *20*

9. REFERENCES ____ *20*

1. Introduction

1.1 Background

Editors of the National Geographic Traveler Magazine [1] elect cyberspace to be one out of fifty places that travelers should visit in their life time. They conclude that Cyberspace is one of the world wonders and a place easy to access for travelers. From the economical point of view, the electronic market size is huge, and Internet can provide advantages to all kinds of economic activities. Therefore, many different electronic commerce applications are created every moment worldwide.

Internet invents a new style of life. It breaks the physical barriers of time and space, so that people can go around the world without leaving home. Most importantly, customers can buy goods or make monetary transactions through the World Wide Web, using mouse and keyboard for buying instead of physically visiting a shop.

Four major elements are associated with payment systems: (1) the parties involved; (2) the means of payment; (3) the medium of exchange; and (4) the infrastructure handling transactions. The parties involved can range from banks or financial institutions, individuals, non-bank corporations, to computer software providers. The means of payment include currency, credit and bank deposit. The medium of exchange includes cash, credit cards, checks, or bills. The infrastructure handling the transaction can be ATMs and POSs, check and bill clearing systems, and Internet banking. These elements are common to most of the payment systems.

For the transactions performed in Internet in an electronic form, we need a secure Internet payment system to handle the transactions. The following criteria should be considered when an Internet payment system is introduced:

- Security: The major concern in the payment system used in the Internet is security. As the communication networks are not secure enough, intruders can steal personal information from customers and make use of the information illegitimately. To prevent fraud and disputes, the system should incorporate entity authentication of the parties, message integrity protection, and non-repudiation of payment order. The number of parties involved in the payment process is also a factor to affect the security of the system.
- Cost: The revenue of payment orders should be larger than the expense of the payment system. Cryptography is used to encrypt critical information before it

is transmitted to the network. Higher security is achieved by a complex cryptographic algorithm at a higher cost. As a result, higher security is used only for higher transaction cost as the cost for complex cryptography algorithms can then be justified.

- Time: The time of the payment process should be reasonably fast so that customers are not kept waiting impatiently. The efficiency of the payment system, on the other hand, depends on the computation time of the cryptographic algorithm, the payment mechanism, and the number of parties involved in the payment process.
- Capacity: The capacity of the system is regarding the number of people who can use it concurrently. In other words, it refers to the maximum number of people who can use the system for on-line purchase without system failure due to overloading.

1.2 Current Payment Systems

Secure Electronic Transaction (SET) [2] was incorporated by MasterCard and Visa. It supports electronic commerce security based on Certificate Authority (CA). SET protocol includes a payment section which is able to deal with different credit cards, and it applies an acquirer payment gateway which is able to authorize the usage of existing bankcard networks. In the authorization request sent by the merchant to the acquirer, the purchase instruction of the customer enables the acquirer to verify that the merchant and the customer agree as to what is purchased and how much is authorized. SET is a well-known secure electronic commerce payment protocol where five parties, namely, (1) customer, (2) merchant, (3) payment gateway (same as acquirer), (4) certificate authority and (5) issuer, are involved in the payment process. Consequently, much computation time is required for producing, encrypting, decrypting and verifying signatures for all parties involved. Although SET is secure for making online electronic transactions, it is not recommended for micro-payment because it is too time-consuming. Besides, all parties may have to authenticate themselves, for security reasons, introducing more performance penalties.

Secure Socket Layer (SSL) Protocol [3] was developed by Netscape Communications Corporation. It provides privacy and reliability between two communicating applications. The protocol is composed of two layers. At the lowest

level, developed on top of some reliable transport protocols (e.g., TCP), is the SSL Record Protocol which receives uninterpreted data from higher layers in non-empty blocks of arbitrary size. The SSL Record Protocol is used for the encapsulation of various higher level protocols. One such encapsulated protocol, the SSL Handshake Protocol, allows the server and client to authenticate each other and to negotiate an encryption algorithm with its associated cryptographic keys before the application protocol transmits or receives its first byte of data. One advantage of SSL is that it is independent of an application protocol. A higher level protocol can be built on top of the SSL Protocol transparently. For online communications, SSL allows traffic between a Web server and a client (i.e., the browser) to be strongly encrypted, using the public key technology. When compared with SET Protocol for online electronic transactions, the major disadvantage of SSL is that it cannot prevent personal information from being stolen. Furthermore, the merchant can examine or tamper this information. Comparisons between SET and SSL can be found in [4].

Qudro-way Internet Payment Protocol (QIPP) [5] is a simple yet secure electronic payment for the electronic market on the World Wide Web. The Protocol imitates the conventional payment in a shop. There are four parties involved: customer, merchant, payment gateway and certificate authority. It is different from most other payment systems as the payment is initiated by the customer and the merchant is not directly involved in the payment process. One resulting benefit is that the merchant is virtually excluded from being able to attack the customer's bank account.

In this report, we present an Internet payment system satisfying the criteria for security requirements. The remainder of this report is organized as follows. Section 2 presents the proposed Internet payment model which is designed for an online electronic commerce application. Section 3 presents TravelNet, an online Web application integrated with the proposed payment model. Section 4 presents the implementation of our payment model which is implemented by Java programming language. Section 5 presents the simulation on different payment systems and they are used to conduct experiments for quantitative comparisons. Section 6 presents an evaluation of the security and performance of our payment system. Section 7 presents the future work. Section 8 presents the conclusion of this report.

2. Payment Model

In this section we propose an Internet payment system. The proposed system simulates the buying behavior of customers who use a credit card or cash card to buy goods. The procedure of buying goods in our payment system is the same as that in a real life.

The system provides credit card and cash payment. Customers must own a credit card or a cash card in order to make online transactions. The conventional shopping choice is preserved. That is, customers first browse the goods in the merchant shop (which resides in a Website), then they pick up those wanted goods (put them into a virtual cart). They bring the goods to the cashier (charge out and triggers the payment process) and the transaction is complete after they pay for the goods (perform online payment transaction). Our target users, however, are those who want to browse products and conduct shopping in World Wide Web.

There are four major entities involved in our system. They are customers, merchants, a payment gateway and banks. The Certificate Authority will manage the certificate and those public keys required for the entities. RSA public-key cryptography is used for authentication and encryption purposes. A pair of private/public keys is generated by the customer or by a trusted third party, i.e. the Certificate Authority.

Our main focus is on the purchasing part (how customers interact with merchants) and the payment process (how money is settled down). Other traditional security issues such as how the keys will be managed and distributed to the users are not our major concern. Besides, there is an general assumption that it is secure from attacks in the communications network between the payment gateway and the existing banking system.

Before we describe our payment system, we introduce the conventions that are used in the message content.

- address: The mailing address of the customer.
- amt: The total amount of the purchased goods.
- card_name: The name of the credit card holder.
- card_no: The credit card number of the customer.
- card_type: There are three types of credit card: MasterCard (MC), VISA (VS), and American Express (AE).
- e_date: The expiry date of the customer's credit card.

- p_opt: There are two payment options: using credit card (CC), and using electronic coins (EC).
- prod_id: An identification number for different products.
- quan: The total quantity of the purchased goods.
- receipt: An unique number recording the transaction for future retrieval when needed.
- RESULT: An acknowledgement from acquirer to merchant, and also from merchant to customer, stating whether the transaction is completed or aborted.
- SIG: The digital signature of a message. It uses the sender's private key to sign on message digest.
- X_cert: A public-key certificate of different parties, denoted by X. It is composed of the acquirer's name, the public-key, trusted third party's name. X = Payment Gateway (pg) or bank (bank).
- X_id: An 8-digit unique number for different parties X. X = bank (bank) or merchant (m).
- X_name: The name of party X. X = customer (cust), or merchant (m).
- X_priv: The private key of party X. X = PG (pg), bank (bank), customer (cust), or merchant (merc).
- X_pub: The public key of party X. X = PG (pg), bank (bank), customer (cust), or merchant (merc).

Table 1: Conventions used in the message content of our payment system

The mechanism of the payment model is shown in Figure 1. The payment process is described in four steps, and the details of the information flows are as follows:

i. The customer first goes to the merchant's homepage and browses products, and puts the selected goods into a virtual basket. After the customer finishes choosing the products, the payment process is triggered by clicking a button. A secure connection between the customer and the merchant is established using SSL protocol for communications. The customer then enters personal information and credit card information into the browser. In addition, the product information and the total amount will be included in the message which is sent to the merchant. The message content (MC1) in this step is

MC1: $\{card\_name, card\_no, e\_date, card\_type, address, prod\_id, quan, amt, p\_opt\}_{by\ SSL}$

ii. Upon the receipt of message MC1, the merchant can get the personal information and credit card information of the customer. The merchant then requests payment authorization and validation of credit card from cardholder's financial institution by composing a message (MC2) which consists of the customer's personal and credit card information, together with the total amount and the merchant's name. This message will be encrypted by the merchant's private key to serve as an authentication. A header, which contains the merchant identification number and a number, denoting the payment option the customer chose, is attached to the message. The whole message is encrypted with the payment gateway's public key to prevent eavesdropping and message tampering. At this step, the merchant will send out the message packet to the PG as

MC2: $\{\{card\_name, card\_no, e\_date, card\_type, amt, m\_name\}_{merc\_priv}, m\_id, SIG, p\_opt\}_{pg\_pub}$

iii. When the PG receives the message (MC2) from the merchant, the PG first uses the private key to decrypt the message to get a decrypted message and a header. The PG will notice the message is sent by a specific merchant but only the merchant's public key can decrypt the header message. Next, PG will communicate with the issuer (the bank issue customer's credit card) and the acquirer (the bank where merchant's account resides) through an existing banking network which is assumed secure. After the PG receives the response from the issuer and the acquirer, the PG will compose a message (MC3) including the response (whether the credit card is valid and the purchase is within the credit limit) and a receipt to the merchant for record purposes. It is then encrypted by the PG's private key for authentication. In addition to the message, the PG's certificate is adhered to the message. The whole message is encrypted by the merchant's public key for privacy and security purpose.

MC3: $\{\{RESULT, receipt, m\_name\}_{pg\_priv}, SIG, pg\_cert\}_{merc\_pub}$

iv. Upon the receipt of the PG's message, the merchant will decrypt the message using the private key and then using PG's public key to obtain the original message. After checking the result, the merchant will compose a message (MC4) to inform the customer if the purchase is successful or not. The message

will be displayed as an html document for the customer. The message can be decrypted by the SSL for the privacy purpose.

MC4: *{RESULT, receipt, prod_id, quan, card_name, address}*$_{by\ SSL}$

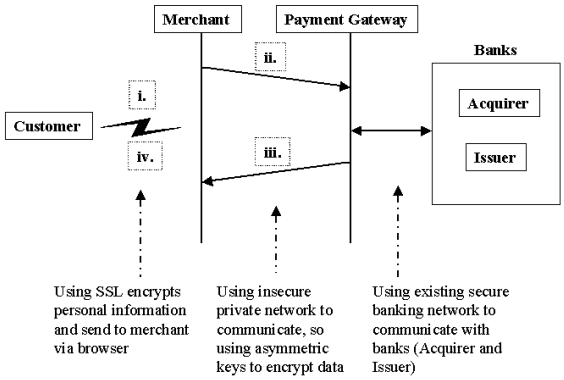

Fig. 1: The Payment System and Its Payment Process Flows

After this confirmation message is sent to customer, the payment process is said to be complete.

We have described a light-weight payment system for E-commerce applications. The system is faster yet secure to handle the personal information from being stolen by malicious users. Descriptions on how the system is secure from attacks are given in section 7. It prevents from eavesdropping, message tampering and masquerading attacks. The payment system is implemented and incorporated into an online travel agent system called TravelNet for testing and performance evaluation.

3. TravelNet

TravelNet is a project that simulates a real life E-commerce application, i.e. an online travelling agency. There are similar E-commerce applications in the web, for example, Expedia [6] and Travelocity [7]. TravelNet is an Web application [8] and it provides services like flight reservation, travel accessories selling, tour guides, and hotel reservation. Secured payment [9] will be done by the payment system mentioned in the previous section and the credit card payment is provided by TravelNet.

The followings briefly describe the architecture and features of TravelNet, and how it cooperates with the payment system through a Payment Gateway (PG).

3.1 The architecture

The overall architecture of TravelNet is shown in Figure 2. Details of the information flow are described as follows:

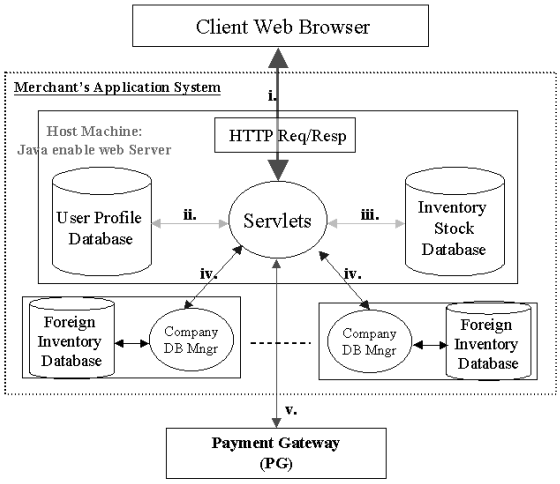

Fig. 2: The Overall Architecture of TravelNet

i. A client communicates with a merchant server through HTTP on the SSL layer. Information from the client like orders and user authentication will be passed to the merchant. Through this channel, the merchant can push responses back to the client.

ii. The merchant server accesses local user profile database for authentication, updating, inserting new users, etc. We implement the merchant server by Servlets [10, 11]. The main advantages of Java Servlet are the great concurrent performance and platform independent nature. Since all the applications only expose object code, users are not able to view the source code of the programs (for cracking or hacking purposes). Consequently, security is provided on the server.

iii. The merchant server accesses its local inventory stock database for getting product information or updating inventories.

iv. The merchant server consults foreign companies (e.g. flight companies in TravelNet) for product information query, booking, ordering, etc.

v. Connected to the payment gateway (PG), the merchant server requests a payment from a specific credit card. Message to PG will be encrypted by an agreed public key of PG and TravelNet's private key will be used for

authentication (MC2). An acknowledgement of a successful or unsuccessful transaction will be encrypted by TravelNet's public key and send back from PG to TravelNet (MC3).

3.2 Features

A number of features are provided based on the architecture of TravelNet. They include:

1). User Registration and Profile Management: Users should register for membership before using TravelNet online reservation and shopping service. After being a member of TravelNet, the users can login the system to use the service and change their profile whenever they want.

2). Flight Search: Users can search for available flights by means of one of three methods, i.e., the one-way search, the round-trip search and the multiple destination search. Once they have searched for a suitable flight, they can add the item in the Itinerary for reservation of tickets.

3). Itinerary Management: Users can view and modify their itineraries with this service. They can also confirm the reservation and trigger the payment service for the reservation.

4). Travel Accessories Shop: Users can buy travelling accessories like luggage, maps and travel guide books in this online shop. They can add and remove items in a shopping basket during their shopping time. Finally, they can check out the shopping basket and request the payment gateway for payment.

5). Travel Guides: Users can obtain tourist information of the cities covered by TravelNet so that it will be convenient for them to plan for their trips.

4. Implementation of our payment model

We implement our payment model by Java programming language version 1.2.2. An additional logi.crypto [12] Java package is served as a tool for developing the encryption and authentication in our payment model.

Four classes (Merchant, PGate, Acquirer and Issuer) are implemented to describe the behaviors of the merchant, the payment gateway, the acquirer and the issuer respectively. The Acquirer class and Issuer class simulate the behavior of the

banking system. The payment authorization and the payment capture are handled at the same time in our model, whereas in the real-life practices, they can be handled separately at different times.

The customer's credit card account and the merchant's account are stored in a Oracle 8*i* [13] database. Verifying the account is the same as retrieving the data from the database for checking.

4.1 Merchant class

It is embedded in the merchant web server. The aim of this class is to ask for payment authorization of the customer's credit card account from the issuer and payment capture from the acquirer. The request is sent to the issuer and acquirer via the payment gateway. When it receives the reply from the payment gateway, it will inform the customer by posting the information on the customer's web browser.

When this class is initiated, it first makes a connection to the Payment Gateway with a specified Internet address and port number by socket programming. After the connection is established, they can communicate to complete the payment order.

4.2 PGate class

It is a gateway for other systems to communicate with banking network. It picks out the relevant information to accomplish the payment authorization or the payment capture and forwards them to the Issuer or Acquirer respectively.

This class is implemented by Java Thread programming. Therefore, it allows multiple merchants to communicate with the Payment Gateway concurrently. Besides, all activities are recorded in a log file called "pgate.log".

Moreover, the PGate class has to be operated continuously to handle the requests from Merchant class and responses from Acquirer class and Issuer class.

4.3 Acquirer class

The aim of this class is to handle the payment authorization. It will verify whether the customer's credit card account is correct and the credit limit is not over. If both are correct, it will debit customer's account and sends acknowledgement to the merchant via payment gateway; otherwise, negative acknowledgement is sent instead.

Besides, all activities are recorded in a log file called "acquirer.log". It records the time and the SQL statements which are committed.

4.4 Issuer class

The aim of this class is to handle the payment capture. It will verify whether the merchant's account corresponds to the right merchant. If it is authenticated, it will credit merchant's account and sends acknowledgement to the merchant via payment gateway; otherwise, negative acknowledgement is sent instead.

Besides, all activities are recorded in a log file called "issuer.log". It records the time and the SQL statements which are committed.

5. Simulation

In this section, we describe the simulation of existing payment systems. We first describe the behaviors of customer in the simulation. Hence we present the payment system, SET protocol, to be simulated. Then we list the assumptions used throughout the simulation and describe the overall picture of the simulation flow in the last part.

5.1 Customer behaviors

For each customer, he/she has two states in the simulation run, either he/she roams around the web page or pays the goods in the web page. In simple words, there are two routes for the customer to behave in the web page. One route is from roaming → paying → leaving the web page; the other route is from roaming → leaving the web page.

The customer arrival time (when he/she requests the web page) variable and the roaming time (total time used in the web page other than paying goods) variable are randomly generated using a exponential distribution.

5.2 Payment system

We simulate the SET payment system and compare with our payment model. We will simulate the purchase request, the payment authorization and the payment capture processes only. The cardholder registration and the merchant registration

processes are ensured to take place before the customer plays it to proceed the payment transaction to the merchant.

We define a set of time variables to act as the time for different operations. The operations are generating symmetric key, encrypting message, decrypting message, verifying digital certificate, generating digital signature, generating digital envelope, and verifying integrity.

We introduce a loading time variable to acquire a more realistic simulation. When there is a huge number of payment transactions coming in the payment system, the loading of the payment system will increase dramatically and the service time will be longer. Adding this variable into the simulation can provide a real picture to that situation.

We will not only simulate the SET protocol, other payment system such as QIPP protocol is to be simulated. Our payment model will be compared with other payment systems in the performance and security levels.

5.3 Assumptions

Here we list all assumptions on the simulation of different payment systems:

i. All time variables are in the unit of seconds

ii. All time variables are with mean μ seconds and standard deviation σ seconds generated from distribution functions (Exponential, Uniform, Normal, Poisson, etc.).

iii. The time of requesting the web page and responding the customer are included in the roaming time variable.

iv. Each customer has two behaviors only. Either they will pay for the goods at most once (one payment is processing per customer) or they will roam around another web page without buying goods.

v. No time is counted into the initializing of wallet software in the customer side.

vi. There is a loading variable which will be added to the payment processing time when there is a multiple of twenty people making payment.

vii. The payment processing time will be counted by a combination of time variables (encryption, decryption, database connection, database retrieving and message composing).

viii. Different payment systems have a bit different implementation in the area of payment processing. However, there will be no changes on the behavior of the customer.

5.4 Simulation flow

In our simulation, we use JavaSim [14], a set of Java packages for building discrete event process-based simulation, and also Java Development Kit version 1.2.2. to implement our simulation model.

The emphasis in our simulation is on the purchase request stage in the payment system. The stage includes customers roam around the web page for favorite goods, customers pay for the goods by electronic means, merchants ask for the payment authorization and the payment capture, and the acknowledgement from the merchant to the customer. The payment authorization and the payment capture are handled by the middle party (Payment Gateway) and/or the banking systems. We obtain the average time occupied in running the simulation for the evaluation of the different payment systems.

6. System Evaluation

In this section, we evaluate the design of our payment model which is incorporated in TravelNet. There are two aspects to evaluate: qualitative and quantitative analyses. In qualitative analysis, we discuss the security attacks that may occur to TravelNet, and how our defensive mechanism works. In the quantitative analysis, we evaluate the performance of the TravelNet payment system.

6.1 Qualitative Analysis

For a system to be secure from potential attacks, it should handle the attacks on eavesdropping, message tampering and masquerading. TravelNet and the payment system are secure from those attacks.

1). Eavesdropping: Attackers cannot see the contents of the message (MC1 and MC4) on the personal information throughout the payment process. The customer's information is encrypted by the SSL protocol. The merchant's message (MC2) sent to PG is encrypted by the PG's public key. Besides, the

message (MC3) sent back to the merchant from PG is encrypted by the merchant's public key. Hence, no one can understand the message except the one who owns the corresponding private key for message decryption.

2). Message tampering: Any encrypted message cannot be tampered with, since it will not be possible to decrypt it after it has been changed. By using message digests, a digitally signed message cannot be tampered with. In MC2 and MC3, for example, digitally signed messages are used to prevent message tampering attack.

3). Masquerading: TravelNet system gets a server certificate from a trust third party for authentication purpose. Masquerading is consequently prevented on the system. Moreover, messages are authenticated with a digital signature to prevent masquerading. As a digital signature uses an owner's private key, no other people owns the private key except the owner.

Comparing with the performance of the SET protocol, our system is faster in terms of the number of symmetric keys generated. In the SET protocol, the number of symmetric keys generated is six while our proposed system does not require any. The time required for creating the symmetric keys increases dramatically when the number of key generation increases. Therefore, we can conclude that our system will get a much better performance than the SET protocol while a satisfactory level of security can be achieved.

6.2 Performance Measurement

In our experiments, the server always allows concurrent users to request a payment and all the requests can be executed concurrently. The merchant, however, can specify the type of execution scenario, either sequential or concurrent. For a single request, the total checkout time in TravelNet is between 1.7 seconds and 2 seconds. The time could be as long as 10 seconds in the worse scenario. To filter out noises, we perform 5 executions to obtain the average time measure for each data point in every experiment.

The performance measurement is based on two different models: Multiple-threaded model and single-threaded model. In the multiple-threaded model, requests are processed in parallel. Each request will obtain only a portion of the server resources, which is reversely proportional to the number of requests. For example,

when there are 10 concurrent users requests, each client process will be on the average 10 times slower than each executing alone, as each of them only grasp 10% of the server resources. The time of overlapping processes will consequently be longer. There is also an extra task switching overhead that is very significant when the number of tasks becomes large. As displayed in Figure 3, the payment process time increases as the number of concurrent users increases. We can also see in Figure 3 that the total payment process time is divided into two parts: time spent on the Merchant client, and time spent on the Payment system server. In terms of the portion of time spent for the total checkout process, payment server contributes over 80%.

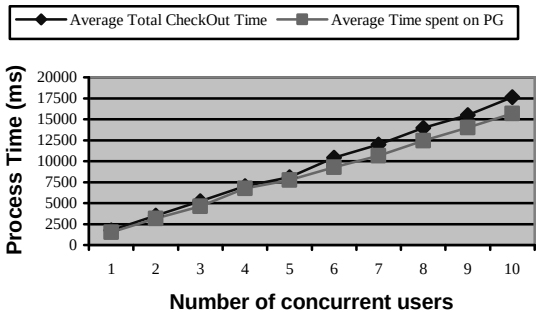

Fig. 3: Payment Transaction Time in Multiple-Threaded Model

In the single-threaded model, TravelNet clients request in a first-come-first-serve manner. Every request waits for all the previous requests to be finished before it can gain access to the server resources. Figure 4 shows the average total process time and the time spent on PG for the single-threaded model. As a comparison, we can see from Figure 5 that its average process time is much shorter than that of the multiple-threaded model. The main reason is due to database resource conflict for the multiple-threaded model when the multiple concurrent processes access the PG, which currently has only one merchant, namely, TravelNet. As the PG server resources have to be shared among the multiple requests, the requests will hold resource (e.g., lock a data item) and compete with each other, thus delaying the complete time. In the single-threaded model, server resources are not shared among the requests and only a task-switching time is necessary between each request. As the response time is quite important in such an interactive application, the single-threaded model behaves better than the multiple-threaded model. It is noted, however, that if we have multiple merchants in the PG which handles different requests with independent merchants, the multiple-threaded model would be significantly improved.

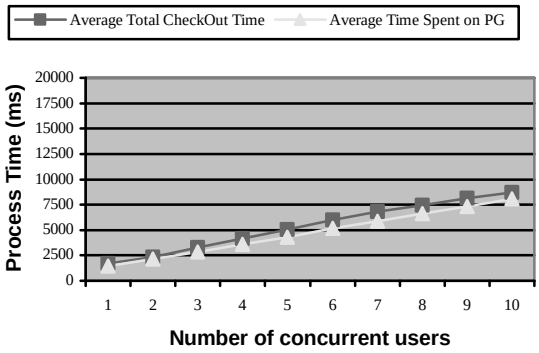

Fig. 4: Payment Transaction Time in Single-Threaded Model

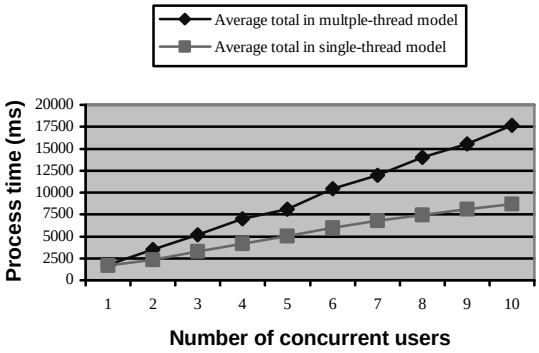

Fig. 5: A Comparison for Single-Threaded and Multi-Threaded Model

The payment processing time can be divided into two parts as well: the time required to perform cryptography algorithms (including message encryption and decryption), and the time required to transmit messages and handle payments. Figure 6 shows the comparison on the payment process time on the PG regarding the overhead due to cryptography. We found that when the number of concurrent users increases, the gap showing the difference on the process time between using cryptographic algorithms and without using them becomes larger. This overhead indicates that for a more secure payment system, there is a tradeoff on the time to handle payment transactions. This tradeoff is quantitatively provided in TravelNet for a detailed analysis.

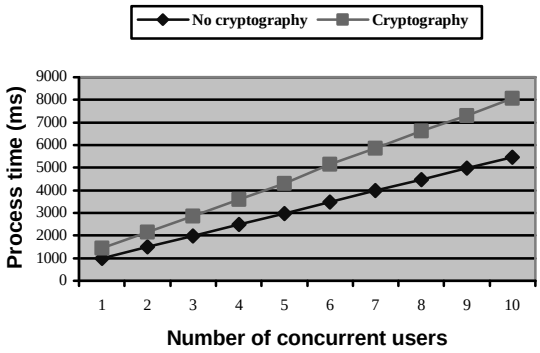

Fig. 6: Single-Threaded Model on the Payment Transaction Time on PG

7. Future work

In the coming days, we will conduct experiments on simulating the SET protocol and other payment systems with respect to our payment model. Quantitative analysis in different simulated payment systems will be compared with each other on the performance and security.

8. Conclusions

Payment system is an essential component in the Internet electronic commerce. We propose a light-weight payment system comparing to the complicated SET protocol. To test and evaluate the payment system, we build an online travel agency called TravelNet, which simulate a real-life E-commerce application.

We implement both the payment model and TravelNet, and conduct performance evaluation on the payment system. The performance results show that our payment system is secure and cost-effective.

Moreover, in order to measure speed-up of our proposed system when compared with SET, we simulate the SET protocol and other existing payment systems in TravelNet and evaluate the performance between these approaches.

9. REFERENCES

1. *50 places of a lifetime,* National Geographic Traveler Magazine, Special 15th Anniversary Issue
2. MasterCard International - What is SET?, http://www.mastercard.com/shoponline/set/set.html

3. Alan O. Freier, Philip Karlton, Paul C. Kocher, *The SSL Protocol Version 3.0*, Internet Draft, March 1996, http://home.netscape.com/eng/ssl3/3-SPEC.HTM
4. Y. S. Moon, H. C. Ho, *Secure Transport Protocol for E-Commerce - SET versus SSL*, Multimedia Information Systems in Practice, edited by Wing S. Chow, Springer, 1999.
5. J. Zhao, C. Dong and E. Koch, *Yet Another Simple Internet Electronic Payment System*, Proc. of the IFIP 1996 World Conference – Mobile Communications (Canberra, Australia, Sept. 1996).
6. *Expedia.com*, http://expedia.msn.com
7. *Travelocity*, http://www.travelocity.com
8. *Web Application Development*, http://www.winwinsoft.com/articles/wad.html
9. *Security in Internet Transaction*, http://www.holt.ie/text/security.html
10. *Web Application Development*, http://www.winwinsoft.com/articles/wad.html
11. C. Darby, *Developing 3-Tier Database Apps w/ Java Servlets*, Java Developers Journal, Feb 1998, http://www.sys-con.com/java/
12. Logi.crypto Java Package, http://logi.org/logi.crypto/
13. Oracle 8*i* Database, http://www.oracle.com/database/oracle8i/index.html
14. JavaSim Homepage, http://javasim.ncl.ac.uk/