Building a Secure Environment for Client-Side Ecommerce Payment System Using Encryption System

Akinyede Rapahel Olufemi, Alese Boniface Kayode, Adewale Olumide Sunday

Abstract- **Client-side computer, network connection between client and merchant web site's server, web site's server and software that allow it to run properly have become the target of attackers. Literature has shown that client-side network of the system has become the target of the attackers, this is because it involves buying, and selling and fund transfer by customers, merchants and financial institutions. Unfortunately, many of the researches that have been done in this area have not provided the required solutions. Hence, the purpose of this paper is to develop a client-side network security that safeguards information stored on the system from individuals that attempt to gain unauthorized access to data. Architecture of the system is designed by using advanced encryption system –AES encryption/decryption algorithm. The symmetric key cryptosystem can protect transaction data such as account numbers, amount and other payment information from alterations. The system uses the most common method of authentication which is user name and password.**

Index Terms: Internet, AES, payment system, goods and services.

I. INTRODUCTION

The most widely used applications, such as instant messaging, email, shopping carts, database programs, security techniques (encryption) and other programs that allows the web to run properly are usually the target of the attackers [1].

Normally at the design stage of any system development, security concerns should be a point of discussion so that it can be addressed properly [2,3] but unfortunately, oftentimes it is not a primary focus when new systems architecture is being developed because of its complexity [2,4].

The purpose of client side security is to safeguard information stored on a system from individuals and malware that attempt to gain unsanctioned access to data. Protection from this type of unauthorized intrusion must be handled by both software and hardware [5, 6].

As a result, we designed and implemented a client/server database for secure client-side e - Commerce payment system.

We made use of MySQL relational database management system (RDBMS) program to serve as a tool for storing and maintenance of data. This will enable users who want to input large amount of information and perform calculations at the same time [7, 8]. However, in e-payment system, server stores records of every transaction.

II. OVERVIEW OF THE SYSTEM

The system was divided into three (3) parts, namely: the merchant server-side scripting, customer-side scripting and payment transaction host-side scripting (figure 1).

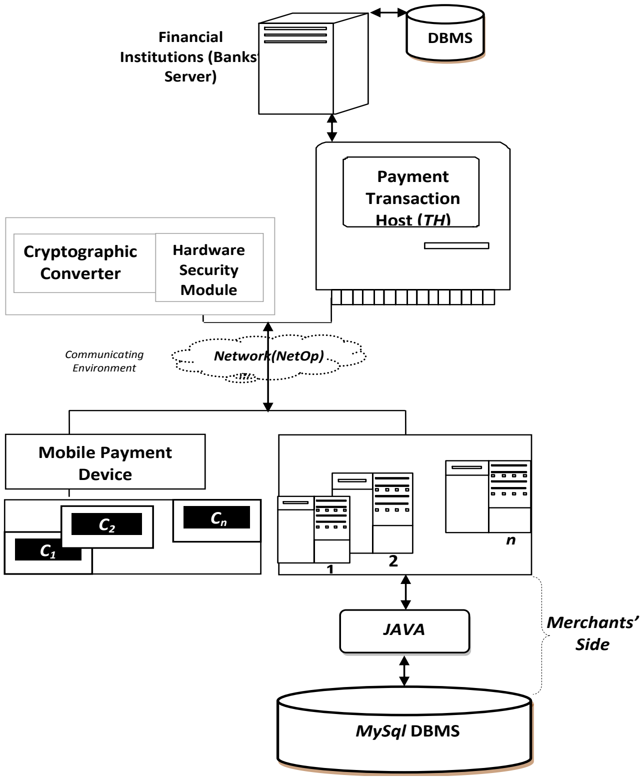


Fig 1: Diagram showing the merchant server-side scripting, customer-side scripting and payment transaction host-side scripting of the designed secure client-side ecommerce payment system.

- **Merchant server-side scripting**: It is a web server technology in which a customer's request is fulfilled by running a script directly on the merchant web server to generate dynamic web pages. The system provides

Manuscript received August 29, 2013; revised on October 11, 2013

R. O. Akinyede is with the Federal University of Technology, Akure, Nigeria, e-mail: femi_akinyede@yahoo.com +2348034702718

B. K. Alese is with the Federal University of Technology, Akure, Nigeria, e-mail: kaalfad@yahoo.com, +2348034540465

O. S. Adewale is with the Federal University of Technology, Akure, Nigeria, e-mail: mxtunde@yahoo.com, +2348033616386

interactive web sites that interface to databases and other data stores.

- **Customer-side scripting**: It is different from merchant server-side scripting in that it allows scripts to run by viewing web browser, usually in JavaScript. The system, which is a client/server architecture in design has three (3) main parts, namely: client module, server module and transaction host module.
  - Client module: The purpose of this module is to pass request made by client to server.
  - Server module: It stores all transaction information in a set of data files.
  - Transaction host module: It processes transactions and transfers fund appropriately.
- **Payment transaction host-side scripting**: It deals with fund transfer.

Figure 1 is secured for clients {i.e. the customers and merchants} because security was considered at the design stage for the needs of the network. The proposed security was AES encryption/decryption scheme.

III. THE PROPOSED SYSTEM PROTOCOL DATAFLOW DIAGRAM

A. Client registration: The proposed protocol introduces a preliminary registration process in order to prevent eavesdropping; and this is important because from experience, we discovered that some merchants are dishonest while lots are incompetent at protecting customers' sensitive data; as a result, we must develop a system that will be able to ensure customers' confidentiality.

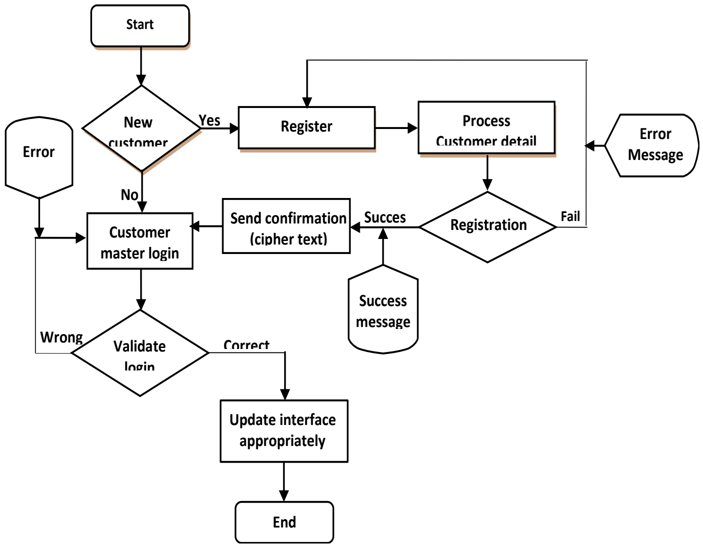


Fig 2: Data Flow Diagram that checks whether the Customer has registered or not

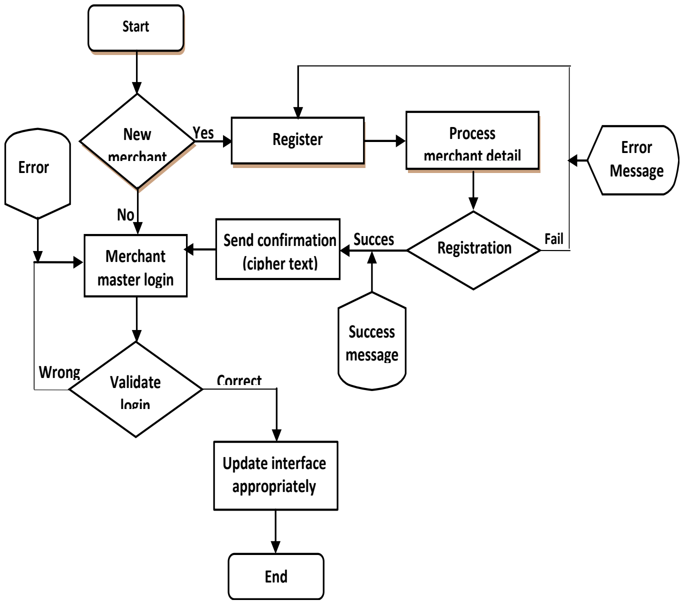


Fig 3: Data Flow Diagram that checks whether the Merchant has registered or not

It was also observed that some fraudsters might want to supply either invalid credit card number or claim refund from their banks without cause. Since there is no way merchant can be protected against these set of fraudsters that present themselves as customers, the system provides protection through the initial registration.

According to the design, customers and merchants must register with a *certificate authority* (CA) before they can transact with each other. Figures 2 and 3 show the data flow diagram that checks whether a customer or merchant has registered or not respectively.

The system design consists of customers, merchant and financial institution. The components of the client module are category, good items, purchase details, sales details and merchant. Figure 4 shows entity relationship diagram for merchant shop details.

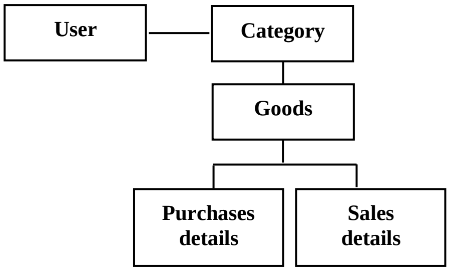


Figure 4: Entity relationship diagram for merchant shop details.

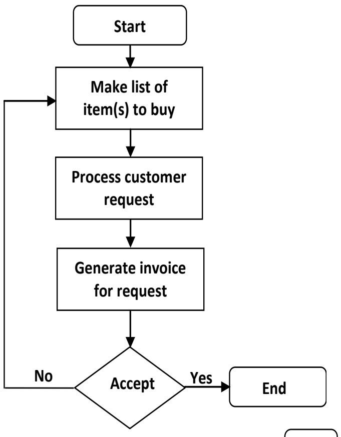


Fig 5: Data Flow Diagram for Viewing Purchase Details at the Customer's side

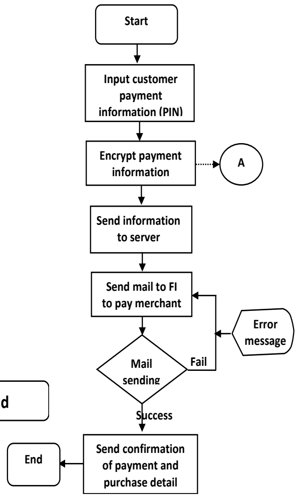

Start
Input customer payment information (PIN)
Encrypt payment information
A
Send information to server
Send mail to FI to pay merchant
Error message
Mail sending
Fail
Success
End
Send confirmation of payment and purchase detail

Fig 6: Data Flow Diagram for Encrypting the Customer's payment information by the Cryptographic Converter detail.

B. Purchase request process: Having registered, the customer will logon to the merchant's site and make a purchase request by sending to merchant his goods ordered information and the payment instructions. The merchant processes the order and starts the payment phase by forwarding the payment instructions to payment transaction host. Note that the proposed system will keep the customer's PIN (Personal Identification Number) secret; merchant cannot simply take this number, as done in telephone credit card transactions, and settle directly with the Issuer[9, 10]. The dataflow diagram for viewing purchase details at the customer's side is as shown in figure 5. In the same manner, figure 6 shows the dataflow for encrypting customer's payment information by cryptographic converter.

C. Client interface for client module: The interface links the clients to the server. For example, it takes all inputs from the clients, and passes them to the server's database system. The clients use the interface to determine what goes into the server. It was created by a click on the client menu.

D Payment Authorisation Process: The merchant will receive the encrypted payment information from the customer and send it to the payment transaction host, which would, in conjunction with the Issuer, check the correctness of the payment information before sending it to the Acquirer, who sends it to the customer for authorisation.

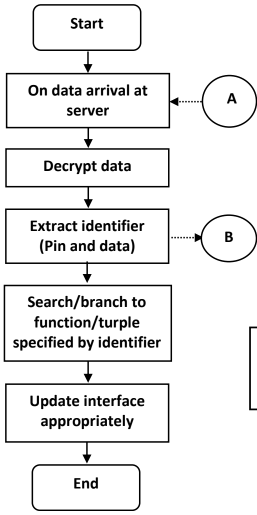

Start
On data arrival at server
A
Decrypt data
Extract identifier (Pin and data)
B
Search/branch to function/turple specified by identifier
Update interface appropriately
End

Fig 7: Data flow diagram for decrypting customer's input information

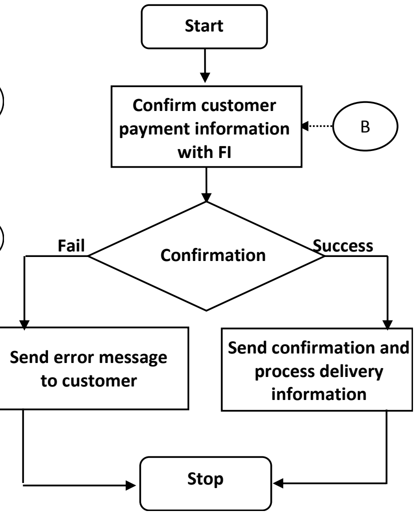

Start
Confirm customer payment information with FI
B
Fail
Confirmation
Success
Send error message to customer
Send confirmation and process delivery information
Stop

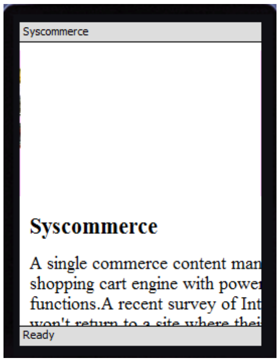
Fig 8: Flow diagram of the customer's information confirmation.

E Payment Capture Process: Here, the merchant sends payment requests to payment transaction host and then payment transaction host checks that everything is correct before responding to merchant's requests. Finally, the actual funds transfer from issuer to acquirer and confirmation of payment with the financial institution will follow immediately (see Figs 7 and 8).

III. IMPLEMENTATION

The layout of the user interface application is designed to be as user friendly as possible. When the user opens the software, the title form and welcome page will appear as shown in Figure 9.

Click on anywhere on the Screen, then a *Register to Begin Page* form will appear. This will allow the registration of both the customer and merchant so that they can *Login* into the system. This is the first step for customer registration. The customer clicks on register new user and supplies his details including the information on the credit card that he wants to use with Certificate Authority (*CA*).


Syscommerce

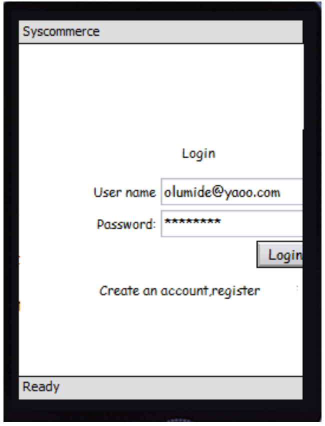
Syscommerce

A single commerce content man
shopping cart engine with power
functions.A recent survey of Int
won't return to a site where thei

Ready

Fi g 9: Welcome Page screen

A. System login process


Syscommerce

Login

User name: olumide@yaoo.com

Password: ********

Ready

Fig 10: System login page screen

The system through *CA* will reply with a registration form, which customer completes and returns, together with the signing key that customer wants to register. The system would process the customer details for registration and send confirmation in form of cipher text. If the registration fails, an error messages would be displayed; and the system would prompt the customer to go through the process again. Then, *CA* checks that the credit card is valid and releases the signature certificate for customer who stores it for future use. All this information (such as credit card details) must be protected, but in case of merchant, he does not supply any credit card details as his form does not request for it. After that, when an OK button is clicked, a *login* page form will appear.

The user will log-in and the system will decrypt the encrypted user name and password from the http request and matches the details of user from database (e.g. validation). If authentication is successful, the system will display a message box with user session that informs the user that he can use the system. Otherwise, the system will display a message box that informs the user to go through the process again.

B. Secure client-side ecommerce payment system

A Click will request the customer to enter his user name and password. If the entry is *successful*, a *login* welcome page is displayed and customer will be allowed access to the system. The detailed security analysis of the login authentication is hereby presented.

Login authentication.

When user (*U*), which can either be the customer or merchant wants to access the *CA*, he carries out the following steps.

a. *U* submits the computed *ID, yId* and *Id* and generates random number a, such that $a \in [1, n - 1]$.

b. Calculates $Q_i = q_iP$ and then $p_i = h(Q_i)$, $X = q_iK_{pub\_ib}$ *and* $g = h(ID||Id||p_i||T_i)$

c. Select random number a, calculates $Q_i = q_iP$ and then $p_i = h(Q_i)$, $X = q_iK_{pub\_ib}$ *and* $g = h(ID||Id||p_i||T_i)$.

d. *U* computes the hashed password $Y = yId$, dynamic identity $dID = p_i\, H(ID_i)$ encrypted and sends message to *CA server*

e. Decrypt $p_i\, H(ID_i)$

f. Verify both certificate & signature

g. *Verification phase*.

h. *Authentication of Server and CA* –After receiving message item number (d) under login phase, and in other to ensure mutual authentication, *CA* runs the following

i. *U* chooses a random number $y \in [1, n - 1]$, calculates $Q_i = q_iP$. *U* computes the shared secret key $SS_1 = x_i^1\, K_{pub\_ib}$.

j. U computes $Auth_i = h\ (Id_i||\ SS_1||\ Q_i||\ T_i)$ and sends $Auth_i$, Id_i, SS_1, Q_i, T_i to *U*.

k. With the aid of the public key parameters mentioned under registration phase, *U* computes the public key, K_{pub} of *CA*.

l. After receiving the $Auth_i$ message in (d), *U* computes $Q_U = q_UP$ and the long term secret key $SS_2 = s_UK_{pub\_i}$. Then, *U* verifies the received $Auth_i$ message. If the verification result is wrong, the protocol will be terminated. Otherwise, the protocol will be continued.

m. *U* computes $p_i = h(s_u^{-1}K_{pubU})$ and extracts $H(ID_i)$ and verifies if $H(ID_i)$ in *U*'s database. If the verification result is wrong, the protocol will be terminated. Otherwise, the protocol will be continued.

n. *U* computes $g_i = h(ID||Id||p_i||T_i)$ and verifies its validity. If the verification result is wrong, the protocol will be terminated. Otherwise, the protocol will be continued.

o. *U* selects a random number $q_U \in [1, n - 1]$, calculates $Q_U = q_UP$. *U* also computes the shared secret key $TB_1 = h(ID||Id||p_i||T_i)$ and $Auth_U = h\ (Id_i||\ SS_s||\ Q_i||\ Q_U||\ T_i)$. Note that TB_1, Q_U and *U* are temporary keys known as secret that is shared with user U_i.

p. *U* sends the output of Q_U, TB_1, and $Auth_U$ to *CA*.

q. *CA* computes the outputs of $Auth_U$ and TB_1 and verifies its validity. If the verification result is wrong, the protocol will be terminated. Otherwise, the protocol will authenticate *U*.

r. *Authentication of Server and User (U)* –After receiving message item number (p) under *Authentication of Server and CA*, then the authentication of server *S* and user *U* would follow.

s. CA_i chooses a random number $y \in [1, n - 1]$, calculates $R_i = \mathrm{h}\ (TB_i\ ||\ Q_i)$ and sends R_i, and Q_i, to U_i.

t. U_i computes $TB_1^* = h\ (ID\ ||Id||p_i\ ||T_i)$ and also $TB_2^* = h\ (TB_i||\ Q_i)$

u. U_i compares TB_1^* with TB_2^*, and verifies the validity in a way of equality. If TB_1^* equal to TB_2^*, U_i sends the TB_1^* and TB_2^* to *CA* and continue. Otherwise, the protocol will terminate.

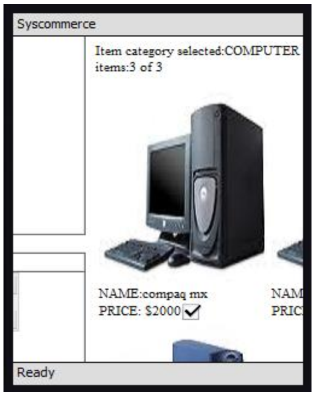
v. *CA$_i$* then computes TB_3^* = (*Ri*||*Q$_j$*) and compares it with either TB_1^* or TB_2^*. If $TB_3^* = TB_2^* = TB_1^*$, then *U$_i$* is assured and the session keys for *U$_i$* and *CA* will be stored in the database. Otherwise, the protocol will terminate.

C. User interface

The user has two interfaces: the merchant interface and customer interface. The merchant interface is used to upload and display available items for shopping. When the customer chooses the desired merchant, the items that are available in his store will be displayed. Then, the customer can select the needed items from the item list and add them to the cart. For example, when an item is selected from item list, the system displays item ID, item price, and the required quantity is filled by customer. Other items can be selected. When icon "Add Item" is clicked, item ID, item name, price, quantity (Qty) and cost can be seen on the table. The purchase total amount is shown under the table. After pressing "OK" button, the purchasing goods quantity will be added to the remaining goods or items (see figure 11). Payment will be initiated as follows:

```
Payment Initialization
Verify $PIN$
        is correct THEN
{$mp$  A: [[PI, signed], C $ID$]}
ELSE $terminate$
      $C \rightarrow M: \{\{Ordered\ Items, Tr.ID\ \ , M_{ID}\}K^{-1}\}K^{secret}$
      $M \rightarrow C: \{Item, M_{ID}, Tr.ID\ \ , ID_{CA}\}K^{secret}$
      $C \rightarrow M: \{\{PaymentOrder, Tr.ID\ \ , M_{ID}\}K^{-1}\}K^{secret}$
```

M processes the order and starts the payment phase by forwarding the payment instructions to *TH (payment Transaction Host)*. Note that the *TH* will obtain transaction data via the network and processes the payment transaction on behalf of a financial institution *FI* that holds the account of the customer *C* for the payment method selected.


Syscommerce

Item category selected:COMPUTER
items:3 of 3

NAME:compaq mx
PRICE: $2000 ☑

NAM
PRIC

Ready

Fig 11: Selected Items and the Prices Displayed

V CONCLUSION

Despite the numbers of client-side ecommerce payment system that have been proposed in the past, not many of them are practicable or implementable. As a result, this paper has suggested a secure client-side ecommerce payment system in mobile environment that takes clients as principal players. The security applied is based on symmetric cryptographic technique and the system will not only reduce the computation cost of all engaging participants but also satisfies the five transaction security properties.

REFERENCES

[1] D. Khusial and R. McKegney, e-Commerce security: Attacks and preventive strategies. Published in 2005. Retrieved in from http://www.ibm.com/developerworks/websphere/library/techarticles/0504_mckegney/0504_mckegney.html

[2] D. N. Chorafas, Management risk. Published in 2004 http://site.ebrary.com/lib/strathmore/Doc?id=10076986

[3] R. Kevin, Internet Security Threats And Protection Methods. Published in 2010. www.krio.me/internet-security-threats-and-protection-methods

[4] K. Allen, and R. Krio, Enterprise Security Risk and Evaluation. Published in 2008 http://www.krio.me/enterprise-security-risk-and-evaluation/

[5] http://www.krio.me/internet-security-threats-and-protection-methods/ (accessed 22/8/2013)

[6] Z. B Omariba, N. B. Masese and G. Wanyembi, Security and Privacy of Electronic Banking. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012 ISSN (Online): 1694-0814 www.IJCSI.org

[7] C. J. Date, An introduction to Database System, ISBN 0-201-824582,Seventh Edition, The System Programming Series, Addison-Wesley Publishing Company,1994

[8] H. P. Pyae, Design and Implementation of Secure Electronic Payment System (Client). Design and Implementation of Secure Electronic Payment System (Client). World Academy of Science, Engineering and Technology, 2008.

[9] D. O. Mahony, M. Peirce, and H. Tewari, Electronic payment systems. The Artech House computer science library. Artech House. 1997 http://www.springerlink.com/content/b2w26822h2252757/

[10] G. Bella, F. Massacci, L. Paulson and P. Tramontano, Formal Verification of Cardholder Registration in SET. Lecture Notes in Computer Science, 2000, Volume 1895/2000, 159-174, DOI: 10.1007/10722599_10