A Research and Implementation of SSL-based Data Transmission System

MO Jia[1,2]
[1]*College of Information Engineering EastChina Jiaotong University Nanchang,China*
[2]*School of Computer Science and Engineering University of Electronic Science and Technology of China Chengdu,China*
Email: mojia@live.cn

Abstract: As one of the major protocols designed for on-line payment system of electronic commerce or E-commerce, SSL defines a mechanism providing data with security layering between applications and TCP/IP, offering data encryption, server authentication, and information completeness as well as user authentication for TCP/IP, supporting applications with a reliable and transparent end-to-end safety services. In the first place, the basic principles and communication process of SSL protocol are discussed in this paper and then the involved data transmission system is implemented through Java language for the sake of safe data transmission. In this system, the safe transmission of data on web has been successfully achieved by means of data encryption and data encapsulation. Secondly, the workflow and the related implementation of SSL servers and clients are discussed to clarify the mentioned details. And in the end, a conclusion is made to reiterate the applicable values of SSL protocols and its prospect in electronic commerce.

Key words: SSL;TCP/IP;Data transmission; E-business;Java

1. Introduction

With the wide application of E-commerce, it has been a primary issue how to safely complete the business through E-payment when consumers are deciding an on-line transaction. Among a host of E-commerce security transaction protocols, SSL(Secure Sockets Layer) represents a most popular one proposed by Netscape company on the basis of Web-oriented security protocols. The security services of SSL include (1)authorized clients and servers ensuring the appropriate transmission of data to the expected parts; (2) enciphered data and the maintenance of complete data to guarantee the non-interpolation of data when transmitted. This paper aims at the secure transmission of data through JAVA language in the light of SSL principles.

2. Principles of SSL Protocol

SSL is a layered protocol in which for each layer message may include the statement, the length, and the content domain. When transmitting messages, SSL will first divide them into disposable data blocks, compress and encapsulate into MAC packages before ciphering. When received, messages will first be deciphered, verified, decompressed and then sent to higher layers after being reorganized. The involved parts verify the identity of their partners and exchange the conversation secret keys by means of handshake. Handshaking can be presented in **Figure 1** where * represents the optional messages.

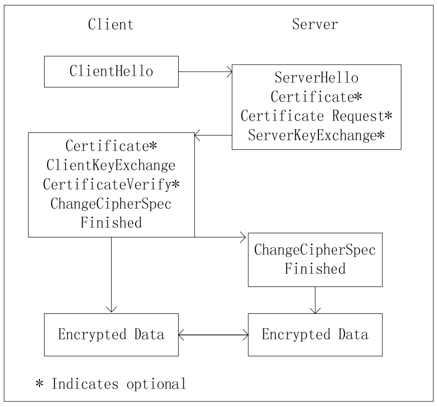


Figure 1. The handshaking of SSL

3.Java-Based Implementioan Of SSL Transmission System

The system consists of the client and server modules in which programs will first read XML configuration files and inquire a ciphered connection to SSL server in accordance with each requirement of the file, and then forward the received data in application client to SSL server. Otherwise, server will set one-way ID certification logos, transmitting data from each verified client to the application servers, which can be demonstrated in **Figure 2** as follows:

For the reason of simplicity, the implementation of the server is hereby introduced. JSSE by Sun Company provides socket-based Java programs SSL support, shielding the built-in implementation details of SSL. By

978-1-935068-09-9 © 2010 SciRes.

adjusting its API, the interactive verification between client and server can be realized and the encystations for all the transmitted data in the socket can be practiced. The following are some key codes for SSL server:

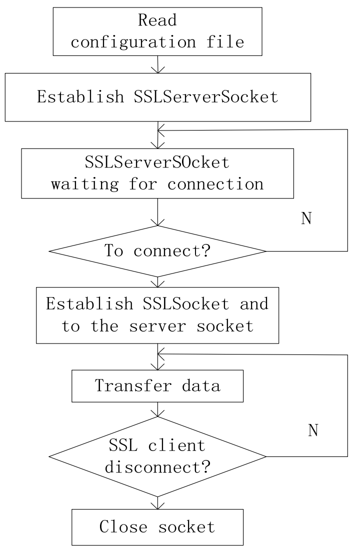
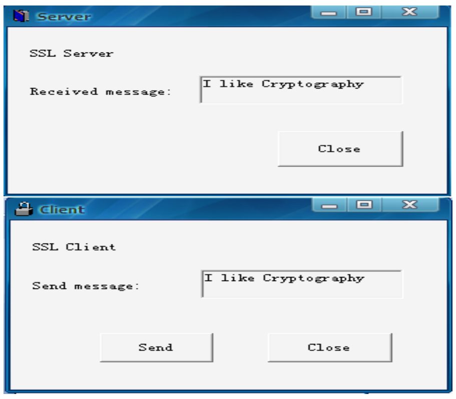

Read configuration file
Establish SSLServerSocket
SSLServerSOcket waiting for connection
N
To connect?
Establish SSLSocket and to the server socket
Transfer data
N
SSL client disconnect?
Close socket

Figure 2. workflow of Server

```
public void start() {
if (serverSocket == null) {
System.out.println("ERROR");
return;
}
while (true) {
try {
Socket s = serverSocket.accept();
InputStream input = s.getInputStream();
OutputStream output = s.getOutputStream();
BufferedInputStream bis = new
BufferedInputStream(input);
BufferedOutputStream bos = new
BufferedOutputStream(output);
byte[] buffer = new byte[8192];
int size = bis.read(buffer);
byte[] data = new byte[size];
System.arraycopy(buffer, 0, data, 0, size);
serverWin.setMessage(new String(data));
String message = new String(data);
if (message != null && !message.equals("")) {
bos.write(message.getBytes());
bos.flush();
}
s.close();
} catch (Exception e) {……    }
class send extends Thread {
BufferedOutputStream bos_ = null;
public send(BufferedOutputStream bos) {
bos_ = bos;
}
public void run() {
while (true) {
String message = serverWin.getMessage();
if (message != null && !message.equals("")) {
try {          bos_.write(message.getBytes());
        bos_.flush();
} catch (IOException e) {…     }
public void init() {
try {
SSLContext
ctx = SSLContext.getInstance("SSL");
KeyManagerFactory
kmf= KeyManagerFactory.getInstance("SunX509");
TrustManagerFactory
tmf= TrustManagerFactory.getInstance("SunX509");
KeyStore ks = KeyStore.getInstance("JKS");
ks.load(
new FileInputStream("data/kserver.keystore"),
        SERVER_KEY_STORE_PASSWORD.toCharArra
y());
tks.load(new FileInputStream("data/tserver.keystore"),

        SERVER_TRUST_KEY_STORE_PASSWORD.to
CharArray());
kmf.init(ks, SERVER_KEY_STORE_PASSWORD
.toCharArray());
tmf.init(tks);
ctx.init(kmf.getKeyManagers(), tmf.getTrustManagers(),
null);
serverSocket          =          (SSLServerSocket)
ctx.getServerSocketFactory()
        .createServerSocket(DEFAULT_PORT);
            } catch (Exception e) {….}
….
```

Its operation result is shown in **Figure 3**.



SSL Server

Received message: I like Cryptography

SSL Client

Send message: I like Cryptography

Figure 3.the operation result of the system

4.Conclusion

The elementary principles about SSL protocol and the related handshaking are first introduced in this paper, on which a reliable SSL-based data transmission system has been designed and implemented. Working above the transmission layer, SSL is said to be transparent for users, which is most favorable to businesses but with no warranty of client message security. For this reason, SET protocol will have been the next focus in our studies.

References

[1] Su Cheng, Yin Zaoling. *An Analysis and Application of SSL Security* [J]. Modern Computers, 2002(29): 29- 32.

[2] Huang Jinchao. *Application of SSL Secure Communications in E-commerce* [J]. Chinese Science and Technology Information, 2007(12).pp.26-28.

[3] Garms J. *Java Security Programming Guide*[M]. Beijing: Electronics Industry Press House, 2002.pp.15-30.

[4] William Stallings.Cryptography and Network Security Principles and Practices[M].Beijing: Electronics Industry Press House, 2006,pp.377-399.

[5] Behrouz A.Fourouzan.Cryptography and Network Security[M].Beijing:Tsinghua University Press House,2009.pp.472-504.