Research and Application of Data Security for Mobile Devices

Xiandi ZHANG[1], Feng YANG[1], Zhongqiang LIU[1], Zhenzhi WANG[1], Kaiyi WANG [1]*

[1]National Engineering Research Center for Information Technology in Agriculture
Beijing, P. R. China
zhangxd@nercita.org.cn, wangky@nercita.org.cn(*Corresponding author)

Abstract. Mobile devices have been increasingly become important tools for the information system application in agriculture, but the safe problem that follow also results in risks of economy lose. Considering above problems and combining the application of an agriculture chain logistics management system, a series of jobs have been done. This paper presents a data security solution which encrypts the configuration files to protect sensitive information and uses SSL protocol to protect the network transmission security. At present, this solution has been modular integrated with the agriculture chain logistics management system. This paper also provides reference for data security problems of other mobile devices.

Key words: Mobile devices, data transmission, security, XML encryption, SSL protocol

1. Introduction

Due to the flexible and customizable feature, wireless mobile devices agree with the special requirements of the agriculture application, such as wide region and poor infrastructure condition. Therefore, related application based on wireless mobile devices will become the development direction of agricultural information system. However, the smallness and portable characteristics also cause the security problems of the sensitive data seem more outstanding. Moreover, the wireless mobile devices currently send data by GPRS while GPRS at this stage need to transmit information via public network. So, the data security in the process of passing data through network faces crucial threat. Data security during transmission of mobile devices application has become an urgent problem to be solved. Aiming at these problems and according to the practical requirement, this paper proposed a set of data security solution. This solution has been applied to an agriculture chain logistics management system. The effectiveness has been demonstrated.

2. Introduction of the agriculture chain logistics management system

The system is composed of client-side and server-side, which communicate data via GPRS. The client-side system running on mobile devices integrates agricultural material reservation, acceptance of the goods, returned goods, sales record and price acquisition. These five modules implement basic business requirement of agricultural material chain stores. The server-side system also integrates these five modules to ensure that the business functions can be completed in the case of client-side system failure. In addition, the server-side system provides inquiry and statistics to each module. Basic data management module of the server-side system makes a uniform management on basic information of agricultural material. These basic information can be transmitted to client-side to support the client-side system operation. And server's IP address, data transmission URL, system username and password are stored in the system configuration module of client-side system.

In the system, there are two significant data security risks as below:

a. Loss of mobile devices may compromise business data and sensitive information which is stored in the devices. People who intercepted above information would spread trade secrets or attack servers.

b. Data interception during transmission will pose a threat to confidentiality,interity and availability of data.If the hackesrs tampered,disguised, replayed the data or made the server deny the data from client-sides,the agricultural material sellers would sustain a great loss.

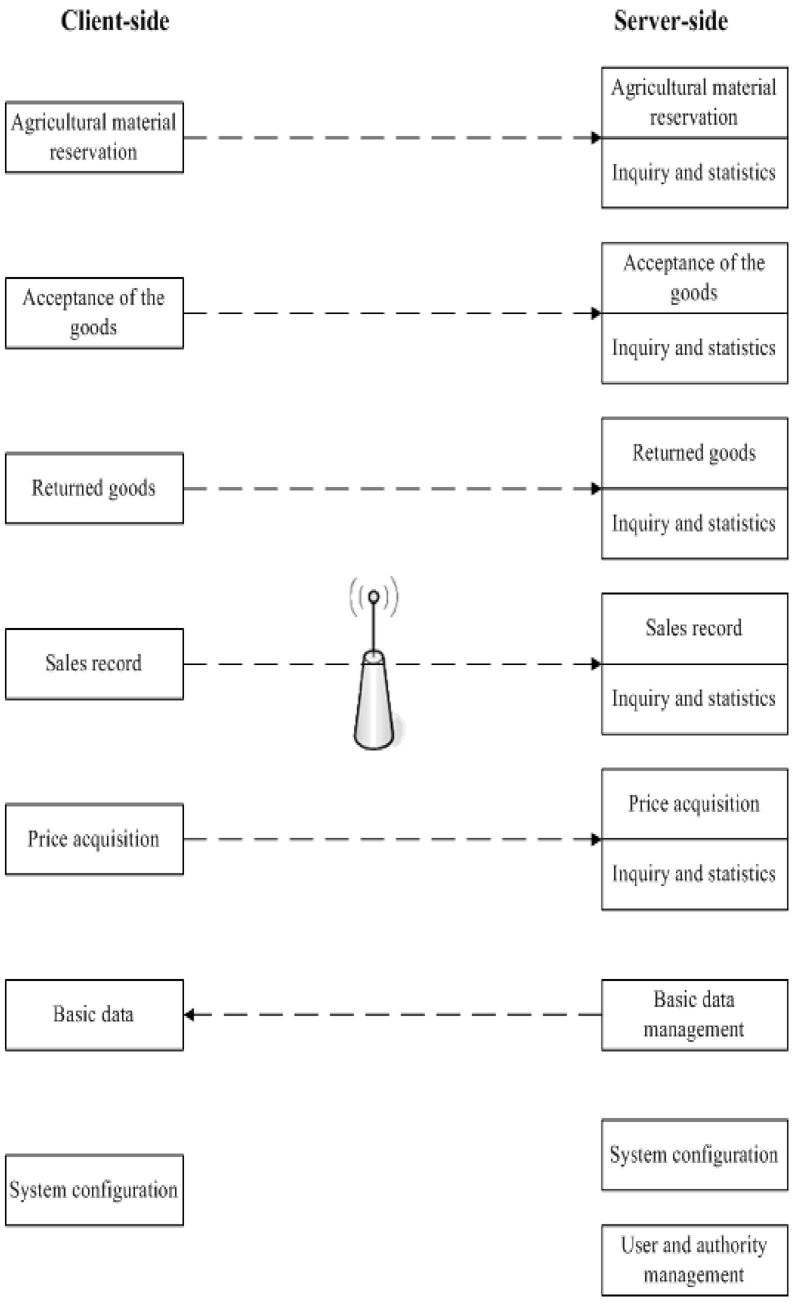

Fig. 1. Agriculture Chain Logistics Management System Structure

3. Data security solution design

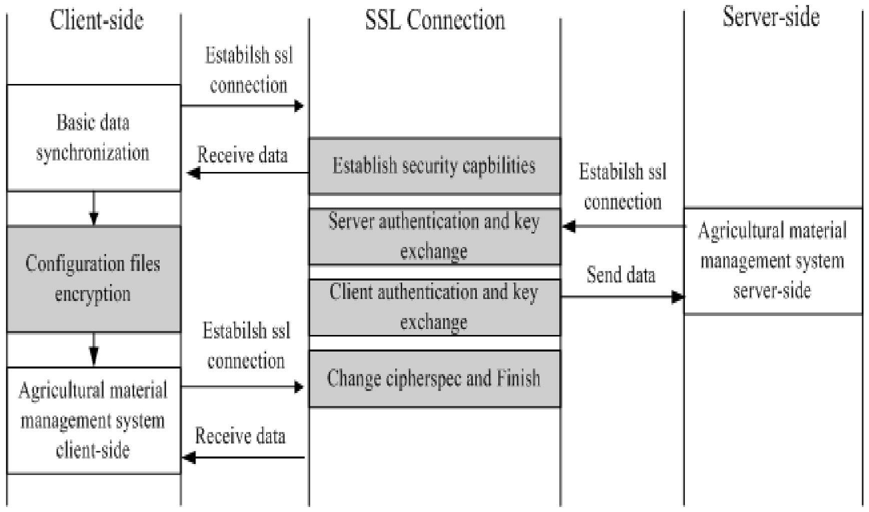
Client-side
SSL Connection
Server-side
Estabilsh ssl connection
Basic data synchronization
Receive data
Establish security capbilities
Estabilsh ssl connection
Server authentication and key exchange
Agricultural material management system server-side
Configuration files encryption
Send data
Client authentication and key exchange
Estabilsh ssl connection
Change cipherspec and Finish
Agricultural material management system client-side
Receive data

Fig. 2. Data security solution

Aiming at the significant safety hazards in the application process of the system and following the simple, practical, low-power principle, this paper designs a security solution as below:

a. Block encrypting configuration files of client-side system. Because part of the profile information needs to be exposed to prepare for flexibly changing, only the server address, database address and other important information require for encrypted storage.

b. SSL protocol to protect data security during the process of network connection. SSL handshake protocol will automatically perform exchanging keys and symmetrical encrypt to prevent the safety hazards after the transmission attacking.

Above both security modules have been established in the agriculture chain logistics management system. To meet the needs of flexible application on other systems, lots of jobs have also been done to reduce the coupling degree. This solution, though, has certain pertinence; it covers most of the basic security requirement of mobile devices. Superadded the modular development, this solution has a general significance for other applications.

4. Configuration files encryption

The system configuration files are XML format, so we mainly study XML encryption. The biggest difference between XML encryption and conventional encryption is the introduction of the concept of encryption granularity; you can encrypt the whole document, the document element or document element content. After encryption the whole element is replaced with an element named <EncryptedData>. The element contains information with encryption or decryption information, encrypted data and encrypted data references, etc.. The <EncryptedData> element is the core element in the syntax. Not only does its cipherdata child contain the encrypted data, but it's also the element that replaces the encrypted element, or serves as the new document root. When encrypting an XML element or element content the <EncryptedData> element replaces the element or content (respectively) in the encrypted version of the XML document [1].

Encrypt the xml using a combination of asymmetric and symmetric encryption requires a symmetric session key to encrypt the data and an asymmetric key to protect the session key. Both the encrypted session key and the encrypted data are stored together in the xml document. The public asymmetric key is used to encrypt the session key while the private asymmetric key is used to decrypt the key[2].

5. SSL

Short for Secure Sockets Layer, a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with https: instead of http:.

The primary goal of the SSL Protocol is to provide privacy and reliability between two communicating applications. The protocol is composed of two layers. At the lowest level, layered on top of some reliable transport protocol (e.g. [TCP]), is the SSL Record Protocol. The SSL Record Protocol is used for 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 and cryptographic keys before the application protocol transmits or receives its first byte of data. One advantage of SSL is that it is application protocol independent. A higher level protocol can layer on top of the SSL Protocol transparently.

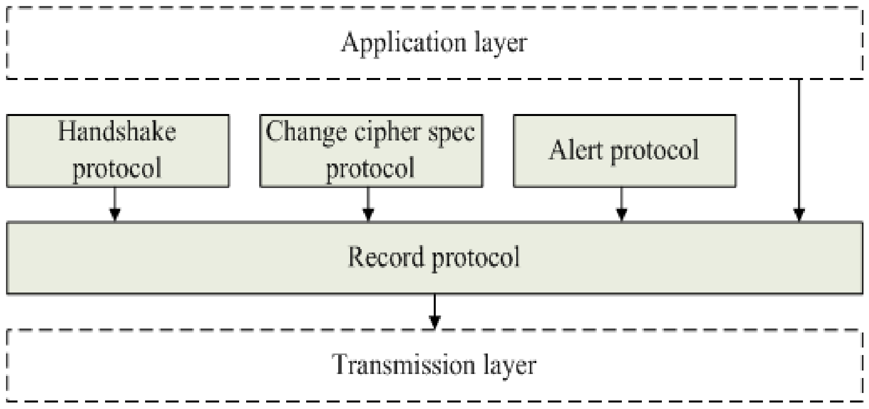

Fig. 3. SSL Protocol

The SSL Handshake Protocol is one of the defined higher level clients of the SSL Record Protocol. This protocol is used to negotiate the secure attributes of a session. Handshake messages are supplied to the SSL Record Layer, where they are encapsulated within one or more SSLPlaintext structures, which are processed and transmitted as specified by the current active session state [12-14].

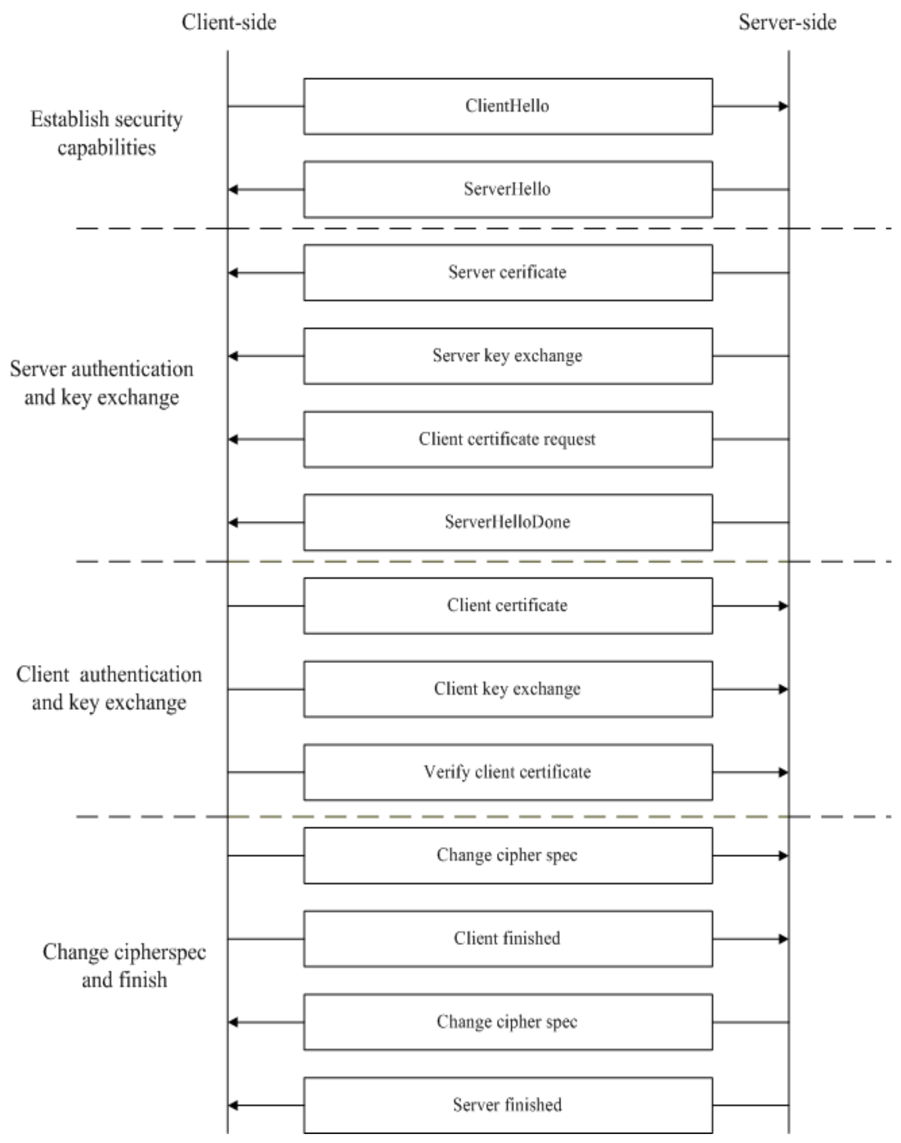

Fig. 4. Handshake protocol

a. The client sends the server the client's SSL version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL.
b. The server sends the client the server's SSL version number, cipher settings, randomly generated data, and other information the client needs to communicate

with the server over SSL. The server also sends its own certificate and, if the client is requesting a server resource that requires client authentication, requests the client's certificate.

c. The client uses some of the information sent by the server to authenticate the server (see Server Authentication for details). If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established.

d. Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server's public key, and sends the encrypted premaster secret to the server.

e. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case the client sends both the signed data and the client's own certificate to the server along with the encrypted premaster secret.

f. If the server has requested client authentication, the server attempts to authenticate the client (see Client Authentication for details). If the client cannot be authenticated, the session is terminated. If the client can be successfully authenticated, the server uses its private key to decrypt the premaster secret, then performs a series of steps (which the client also performs, starting from the same premaster secret) to generate the master secret.

g. Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity--that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL connection.

h. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.

i. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.

j. The SSL handshake is now complete, and the SSL session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.[12-14]

6. Implementation of secure data transfer solution

6.1. Encrypting configure file

We encrypted the key of symmetric encryption algorithm using asymmetric algorithm on configure file, to ensure the security of key exchange, then we make encryption and decryption on file data using symmetric encryption algorithm, which is the recommended method of XML encryption specification. In the process of configure file encrypting，firstly public key and private key pair was generated and saved into security key container，secondly one single session key was generated by AES algorithm, by which XML document element was Encrypted，thirdly AES session key was Encrypted using RSA public key, finally the encrypted AES session key and encrypted XML data was saved to new element of <EncryptedData> in XML document. When calling configure file, decrypt opposite direction[10]. Below is one example of configure file.

```
<?xml version="1.0" encoding="utf-8" ?>

<BaseData>

  <SERECTDATA>
<SENDURL>http://192.168.2.215:8080/SMS/ws/WSPort?wsdl</
SENDURL>
<BASEURL>http://192.168.2.215:8080/SMS/ws/BaseWSPort?ws
dl</BASEURL>

    <CONNECT>Data Source =
{0}\DB\SMSDB.sdf;Password=******</CONNECT>

  </SECRETDATA>

  <ORDERTIME>14</ORDERTIME>

</BaseData>
```

Among of them, <SENDURL> is the URL where sends agricultural material reservation order；<BASEURL> is the URL where Synchronize basic information of agricultural material ；<CONNECT> is the connecting information to database system. Because the leaking of this information will bring threat to the server, it is necessary to encrypt these elements. Whereas <ORDERTIME> is time limiting information of sending agricultural material reservation order, which can be exposure and edited, so we don’t encrypt this element, then the encrypted configure file is as below:

```
<?xml version="1.0" encoding="utf-8"?>

<BaseData>

  <EncryptedData Id="EncryptedElement1"
Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#"><EncryptionMe
thod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"
/><CipherData><CipherValue>OQ+TkM4IIcQqDbUzoiBo8R1ZG6mB
YdZgJKqHR6xNy0FE0OeVx2pc11NbEcKyW45vHFtU7Ihe6X0f8+z9kX0
K5rqGarPn9ARm/rDUUlrhXPF54981/7uoIw3gR7q1u24uKwpWwdERtn
gjo73GfwXgpGTrF/hjtv91X1RQO1BShMNuJTEducitlNUEy82JBHWgo
+2yvLzhFHWXScJP+w89oVFIhy3XTS880tehjBYI8Sz4xbo9/AO5ptDc
w4DZEVOVKZ9uqfh4497UgDy4hh6WAg7OhgsXTecFkzpNFG3LkmY3/hS
5mxDCLE7HAdk8k/uhnu5EDI5osc1r+xSCkkip3uqOFC85GRyiRNDu13
OMwj0=</CipherValue></CipherData></EncryptedData>

  <ORDERTIME>14</ORDERTIME>

</BaseData>
```

6.2. Implementation of SSL

There are several kinds of SSL implementation tool box in the market, such as OpenSSL, Mod_ssl etc. Although these tool boxes are powerful, they are too large for embedded device. MatrixSSL and axTLS are specially developed for SSL implementation in embedded device. MatrixSSL’s library is smaller than 50K, but tedious when handling I/O. AxTLS has least function, just providing several encryption algorithm, but considering the specificity of embedded device, it is a good choice. In the Agriculture Chain Logistics Management System, We implemented SSL protocol using axTLS. The client software of the system was developed in C# language, whereas the server software was developed in Java language. The client was hosted in PDA moving equipment. Server software was hosted in the Server machine, handling accesses from external network. One example of SSL connecting process is below:

- Firstly initialize client function as below:

```
axTLS::SSLClient::SSLClient  (  uint  options,
   int  num_sessions
  )  [inline]
```

- Apply for establishing a new SSL connection from server

```
SSL axTLS::SSLClient::Connect  (  Socket  s,
   byte[]  session_id
  )  [inline]
```

The sever side will return handshake status to judge if success of Handshake

- Data Transfer

After the success of handshake，we can make secure data transfer. In axTLS, we use the function of read() and write() to accomplish read and white operations of socket

```
int axTLSj::SSLCTX::read  (  SSL  ssl,
   SSLReadHolder  rh
  )  [inline]
int axTLSj::SSLCTX::write  (  SSL  ssl,
   byte[]  out_data
  )  [inline]
```

- Close Communication

When the communication finishes between client and server, Dispose() function should be called to release SSL resources.

```
void axTLSj::SSLCTX::dispose  ()  [inline]
```

7. Conclusions

In this article,we introduced a secure data transfer solution of moving equipment using XML encryption and SSL protocol, and implemented it in Agriculture Chain Logistics Management System. In the practice, the solution satisfied the security requirement of data transfer in system, in which the server side is in Myeclipse framework and client side is in .NET3.5 framework. We believe as the moving equipment in agriculture becomes more and more Diversity，the data transfer security problem will drawn wide attention.

Acknowledgements. This research is funded by the Support of Science and Technology Project of the State, Programmed award No. 2008BADA0B02.

References

1. Geng Jiangyong,Lu shiwen.Safe Data Exchange Based on XML Encryption[J].Computer Applications and Software.2005,22(2):99-101.
2. Che Kui,Niu Xiaotai,Xing Shutao.Research and implementation of encryption method based on XML[J].Computer Engineering and Design.2008,29(20):5180-5183.
3. Behrouz A.Forouzan.Cryptography and Network Security[M].Beijing:Tsinghua University Press,2009.471-506.
4. Liuqiang.The Research and Design of Network security system based on SSL protocol[D].Shandong University,2009.
5. Canxianjv.SSL protocol and its application in the realization of mobile terminals[D].University of ELectronic Science and Technology of China,2008.
6. Bai Zhizhong,Luo Yunqian,Xia Jingbo.Security realization of embedded Web Server based on SSL[J].Electronics Opitics & Control.2006,13(3):61-63.
7. Ruan Xiaowei,Du Jiang.Study & Implementation of SSL Protocol's Security[J].Computer Science.2009,36(4):182-184.
8. Luo Yunqian,Xia Jingbo,Zhao Xizhen.The Realization of SSL in the Embedded System Based on MatrixSSL[J].Microcomputer Information.2005,21(11-2):33-35.
9. Zhang Xiaoming,Wang Zelin,Lu Jiande.Design and Implementation of Embedded Network Secure Communication Based on OpenSSL[J].Computer Technology and Development.2006,16(1):217-220.

10. http://msdn.microsoft.com/zh-cn/library/ms229746(VS.90).aspx,2007-7.
11. http://axtls.sourceforge.net/dox/index.html.aspx,2007.
12. http://publib.boulder.ibm.com/infocenter/wmqv6/v6r0/index.jsp?topic=/com.ibm.mq.csqzas.doc/sy10660_.htm.
13. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt.
14. http://www.cryptoheaven.com/Security/Presentation/SSL-protocol.htm.