Introducing Secure Data Transmission Scheme in a Heterogeneous Environment

Kun Liu
Shandong Provincial Key Laboratory of Network Based Intelligent Computing
University of Jinan
Jinan, China
liukun@ujn.edu.cn

Kun Ma
Shandong Provincial Key Laboratory of Network Based Intelligent Computing
University of Jinan
Jinan, China
ise_mak@ujn.edu.cn

Ajith Abraham[1,2]
[1]Machine Intelligence Research Labs (MIR Labs), WA, USA
[2]IT4Innovations, VSB-Technical University of Ostrava,
Czech Republic
ajith.abraham@ieee.org

***Abstract*—Various information systems in the heterogeneous environment have resulted in a lot of discrete business data, and this type of storing method decreased the sharing of the data. In this paper, we designed and implement a secure data transmission scheme in the context of High Education Examination Management Information System of Shandong Province (HEEMISSP), in which all kinds of factors in the heterogeneous environment is taken into consideration. We integrate the security certificate technology and the encryption method with the remote method invocation (RMI). Measured by the security, completeness and non-repudiation of data transmission, the design fully satisfied the efficiency of data transmission which is requested by customer.**

Keywords-Data Transmission Plan; Heterogeneous Environment; HPROSE; Identity Authentication

I. INTRODUCTION

With the fast development of information technology, the management information system (MIS) in the heterogeneous environment increased rapidly, which produces a large number of discrete business data. This type of discrete storing method decreased the sharing of data. Distributed system bridged these discrete resources, which enables these resources to be shared. However, since the openness and unstability of the Internet seriously affects the data transmission, the strategy of data transmission is quite essential in the distributed environments. The security, completeness and non-repudiation of data should be considered during transmission. In addition, the transmission speed still needs to be considered according to the urgent priority. The data transmission in the distributed environments also applies to this rule. This paper designed and realized a solution of secure data transmission of distributed system, at the basis of HEEMISSP.

II. STUDY BACKGROUND

With the further progress of informatizaiton, a lot of management information platforms emerge accordingly, in which process produces a lot of business data. Although these data belong to the same system logically, they all stored in the heterogeneous databases and the business systems in different format [1]. A distributed system is a collection of individual computers, and its application that could connect to these scattered resources, and enable the users to utilize them as one system [2]. Therefore, distribution system acts as a bridge for these discrete data.

The application of distributed system communicates with these discrete business data, and realizes the information exchange through the internet. But as it is well known that, the data transmission process through the Internet is not absolutely secure. There exist many risks. To be summarized, they could be represented as following basic risks [3]: 1) information leakage, i.e. disclosed or revealed to an unauthorized person or entity; 2) Integrity damaged, i.e the data is not the original one; 3) business rejection, i.e legal access is blocked, and it is blocked out unconditionally; and 4) Illegal use, i.e. the resources are illegally used. Discrete data need to be transferred through the Internet, however, its secure bug is worried.

Therefore, security and speed are the essential issues to be considered in the context of data transmission. How to bring security and speed into balance is the challenging, and that is also the focus of this paper. This paper designed and realized a solution of data transmission with the background of distributed environment, and balanced the issues between security and speed of transmission.

The rest of the paper is organized as follows. Section 2 discusses the related work, and Section 3 introduces the architecture of the structure design of the secure distributed data transmission. Section 4 discusses the further design of our system. Conclusions are outlined in the last section.

III. RELATED WORK

Protocol is the standard for data transmission. With the protocol with high quality, we can transmit the data in a more efficient and simpler way [4].

So far, there are a lot of technologies and protocols for the data transmission. For example, DCOM, CORBA, JSON-RPC [5], Web Service [6], SOAP, XML-RPC, publish/subscribe message middleware [13] [14], etc. The traditional way is the remote mutual invocation between the distributed objects. However, with the background of Internet, it is not easy to realize Web system through traditional distributed objects. As for the networking, using remote invocation also has some security issues. This type of

data transmission may be blocked out by the proxy server sor firewalls, while HTTP protocol could easily penetrate the firewalls. Most of firewalls do not block out HTTP, and HTTP protocols are supported by all the browsers and servers. So the data transmission based on HTTP is widely used [2].

High Performance Remote Object Service Engine (HPROSE) is high performance Remote Procedure Call (RPC) Protocol which is based on HTTP protocol, and it is a lightweight, secure, cross-net, cross-linguistic, cross-platform, cross-environment, cross-domain protocol which supports complex transmission among objects and invocation parameters, output re-orientations, classified errors, dialogues and service orientations. HPROSE transmits data through an efficient serialization, and does not need a second coding. Its streaming reads and writes directly. The data in the remote method invocation is directly restored in the target type with the high efficiency. At present, HPRSOSE supports C++, NET, Java, Delphi, JavaScript, ASP, PHP, Python, Perl, etc. It could keep all these kinds of languages consistent, and it could be used to create powerful cross-platform, cross-language and distributed applications. HPROSE supports six major platforms: desktop applications, browsers, enterprise servers, cloud platforms, mobile devices and built-in system micro terminals. It could be applied in any distributed system which is related to network, such as the education and the scientific research, the government organizations, the medical treatment and the public health, the enterprises and the daily life etc.

Compared with JSON-RPC and XML-RPC, HPROSE is better in terms of serialization way, transferring of objects and usability etc [7-9].

IV. STRUCTRUING DESIGN OF AN SECURE DISTRIBUTED DATA TRANSMISION

The secure data transmission is decisive in the rapid informatization of modern life. If the data is stolen or tampered during the transmission process, the individual interest is not the only affected factor. This paper analyzed current security technologies and theories which are based on the network transmission to integrate it with our solution, and implemented the solution of secure data transmission in the heterogeneous environment for the purpose of secure data transmission among servers in distributed environment.

A. Overall structure of distributed system's data transmission

There are two ends in the data transmission: sending end and receiving end. Sending end is the producer of data, while the receiving end is the user of the data. Sending end sends data, while receiving end receives the data sent by sending end through network. Figure 1 shows the detailed architecture.

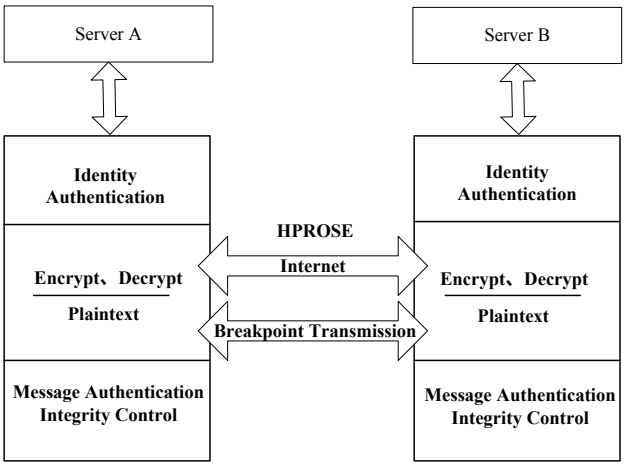

Figure 1. Data transmission architecture between distributed system servers

Server A and Server B are two random servers in the distributed environment respectively. The data produced between these two servers are transmitted through network. They all support the identity of authentication, the transmission of the plaintext and encryption. They all could verify the completeness of the data.

B. Overall design of distributed system data transmission scheme

The data transmission in the distributed scheme includes the identity of authentication, the data sending and receiving, and the verification of data completeness. Figure 2 shows the overall design of the transmission scheme:

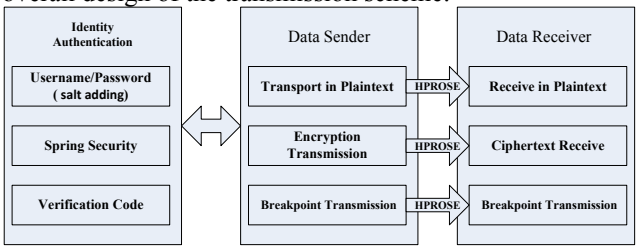

Figure 2. Overall design of distributed system data transmission plan

(1) Data transmission authority

Only those who have the authority are allowed to transmit data through the HPROSE. The identity of authentication part will verify the data transmitter's identity, to avoid any illegal user to transmit data through the HPROSE.

(2) Security requirements of data

Different data have different security requirements: the encryption or plaintext transmission. For those data which have higher sensitivity coefficient or have less requirements for transmission speed, they can adopt the encryption transmission; for those data which have lower sensitivity coefficient while have higher requirements for transmission speed, they can adopt the plaintext transmission.

(3) Data completeness

Data sending end sends out the processed data, and the receiving end receives the data accordingly, and restores the

data in their original formats for backup. Receiving end compares the received digital digest with the generated one to verify whether the received data is complete or not. If they are the same, it means that the data is complete; otherwise, the data is tampered.

(4) Speed requirements of data transmission

The unstability of the network connectivity, or too much transmitting data will lead to the interruption of the transmission. Re-transmitting will certainly cost a lot of additional work, and it will decrease the transmission speed and waste a lot of time. Therefore, in order to increase the transmission speed, the breakpoint resume is developed.

Breakpoint resume could resume the transmission which is interrupted by the network disconnection or the delay to increase the transmission speed. In addition, for massive data, they could be compressed first, and then be transmitted.

C. *Technical architecture of distributed system data transmission plan*

The system is divided into five layers: presentation layer, business logical layer, service supporting layer, objects persistent layer and data layer. They are showed in figure 3.

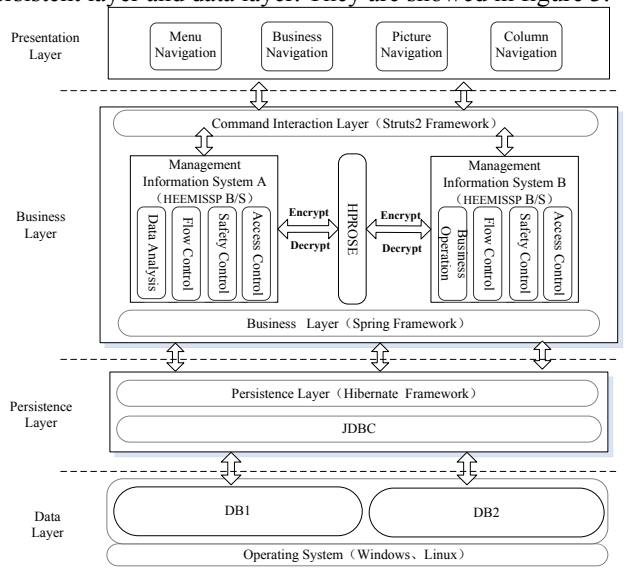

Figure 3. Technical architecture of distributed system data transmission plan

The presentation layer is responsible for the interacting with the users and presenting the data, and it presents the view objects transferred by business logic layers through struts tags. Command interactive layer is responsible for forwarding service requests, error handling, abnormal handling, page orientation, and mutual management between users and system, and providing presentation logic in the user layer and interface access in the service supporting layer. Business layer is the main interface of business logics. It realizes all the business process and the secure distributed data transmission through the HPROSE, only invoke the interface of persistent layer, and provides the access interface for the presentation layer. Persistent layer realizes the uniform access for various resources in the system and external resources, and provides interface of accessing database and data for business logic layer.

V. FURTHER DESIGN OF AN SECURE DISTRIBUTED DATA TRANSMISION

A. *Identity authentication module*

In the distributed environment, when users access various systems, the identity of authentication is a very serious issue. Normally, it uses the username and password to be authenticated in the server, and then verifies the resource of the data. The system in this paper uses the most common user name and password. It integrates MD5 and salt adding technology. The username is the salt, after the encryption of password, the security is reinforced. While Spring Security Framework perfectly packages the identity authentication, it avoids the writing in the development process and optimizes the structure of procedure code at the basis of acquiring the same effect.

In addition, the computer could not identify the distorted words accurately and rapidly. Therefore, applying the security code is beneficial for preventing brute force attack, and it could protect the security of the system to some extent.

B. *Design of data encryption level*

According to the security requirements of data itself, and the requirements of transmission speed, we applied levelled data transmisison method. We set two transmission levels: level 0 and level 1. Level 0 is the default level. Level 0 represents no encryption for the data, while level 1 represents encryption for the data.

The transmitted data through plaintext has no high demands for the confidentiality. Even they are tampered during the transmission, it will not result in big damages. It applies to the data which have low requirements for the confidentiality, and it also applies to those data which not only has high requirements for the confidentiality, but also the transmission speed. But all the transmitted data should ensure its completeness and non-repudiation as much as possible, and also ensure the leglity of the data received by receiving end. Therefore, for those data which have high requirements of transmission speed, they could choose RSA-based signature authentication transmission method.

Encrypted transmission still have the function of validating the data completeness and non-repudiation. It uses AES algorithm to encrypt the data transmitted, and then uses RSA algorithm to encrypt AES encryption key, and finally sends the data out to receiving end after it signs.

Figure 4 shows the encryption process of the sending end, and the decryption process of the receiving end.

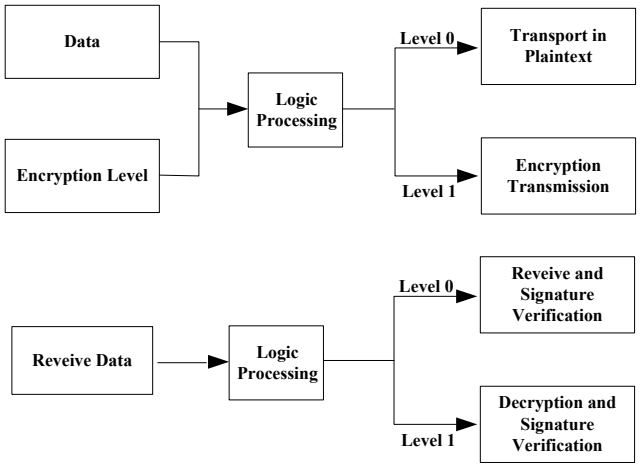

Figure 4. The flow chart of the encryption process of the sender and the decryption process of the receiver

C. *Design of the encryption and decryption of the data transmission*

This paper uses AES and RSA algorithm [10-12] to encrypt the data.

Before the first data transmission, it needs to generate an encryption key required by encrypted transmission. Meanwhile, in order to reinforce the security of encryption key, we designed the system to recreate an encryption key before every encrypted transmission.

(1) The generation of encryption key

The generation of encryption key includes the creation and management of the encryption key. The creation of encryption key is divided into the creation of AES and RSA encryption key. The management of encryption key involves the storing of AES encryption key and preservation of RSA encryption key.

Sending end needs to generate a pair of AES and RSA encryption key respectively, store the AES encryption key and RSA private key in the hard disk, and distribute out the RSA public key. AES encryption key is used to encrypt the data transmitted, while the RSA private key is responsible for the signature during the transmission process. RSA public key is used to decipher the signature after it's downloaded by receiving end.

Receiving end needs to generate a pair of RSA private keys, and then store them in the hard disk, and finally distribute the public key out. Public key is used to encrypt the AES encryption key in the sending end, while the private key is used to decipher the AES encryption key to get the original data.

(2) Data encryption

Before sending end transmits encrypted data, it needs to have the encryption key firstly. The encryption process is as figure 5 shows. F represents the plaintext transmitted.

Sending end uses its own AES encryptionk ey to encrypt the data transmitted, and produce the ciphertext;

Sending end uses the received RSA public key from receiving end to encrypt the AES private key which is randomly generated in step 1);

Sending end will attach the AES encryption key which is encrypted by RSA public key to the ciphe rtext which is encrypted by AES encryption key, to form a new data F1;

Sending end gets the digital digest H of data Fa by MD5 calculation;

Sending end uses its own RSA private key to encrypt the digital digest H, then get the signature S;

Sending end attaches the signature S to data F1 to form data F2.

After the encryption, data F2 then have the authenticity, completeness, and non-reputation, and they could be sent to the receiving end.

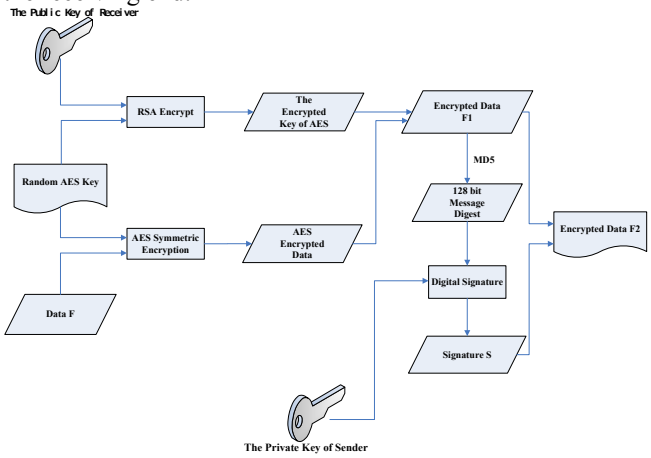

Figure 5. The encryption process of data transmission

(3) Data decryption

The data decryption occurs in the receiving end, it includes the data decryption and the verification of data completeness. Figure 6 shows the process, and the signs in the figure are the same as encryption process.

1) Receiving end receives the data F2 which is sent by sending end. Firstly, it separates the ciphertext data F1 from the digital signature S according to the integration rules set by sending end.

2) Secondly, receiving end uses the MD5 to calculate the digital digest of data F1, meanwhile, it encrypts the signature S by the RSA public key received from sending end to obtain the digital digest sent by sending end, and compares the calculated digital digest with the new generated ones byte by byte. If they are the same, then continue the next step. Otherwise re-get the ciphertext data F2, and re-separate and re-judge the data;

3) Receiving end separates the encrypted AES encryption key and the ciphered data which is encrypted by AES encryption key from F1;

4) Then, receiving end uses its own RAS private key to decrypt the AES encryption key;

5) Lastly, receiving end uses decrypted AES encryption key to decrypt the cipher text which is encrypted by AES, then get and store the original plaintext data F which is sent by the sending end.

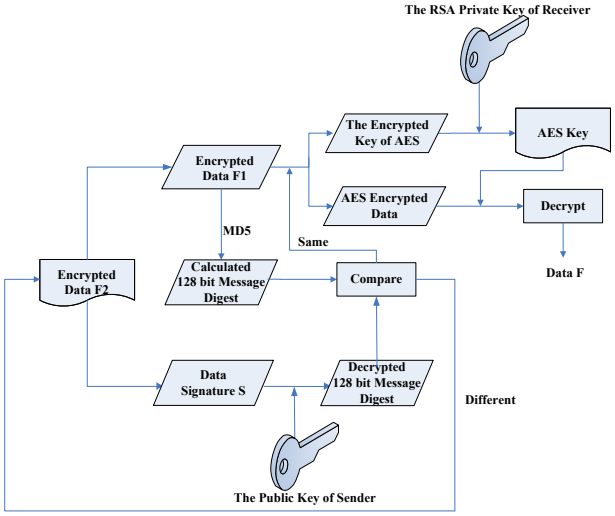

Figure 6. The decryption process of data transmittion

D. *Data transmission degign based on RSA signature*

It's obvious that the data transmission by plaintext in the network will face risks. The existence of such risks forces us to seriously transmit all data, even under the circumstances that they do not have very high requirements for security. Balancing the speed and security of data transmission, and at the precondition that we do not need to encrypt the data itself, we could ensure the data completeness and non-repudiation through RSA signature.

(1) Sending end

Before sending the data, sending end needs to consider the completeness and non-repudiation of the data transmitted. Therefore, under the situation of plaintext transmission, we designed the sending end as follows:

Data F is the data to be transmitted. Sending end uses MD5 to calculate the digital digest of F. Then it encrypts this digital digest with the received RSA public key from the receiving end, and encrypts the ciphertext data F1 with its own RSA private key(see figure 7) to get signature S. Finally, it attaches the signature S to data F, and sends them together to receiving end.

By such, the sent data F2 have the signature of sending end, thus the non-repudiation. While the authentication of signature could ensure the data completeness, and enable the F2 have the completeness and non-repudiation.

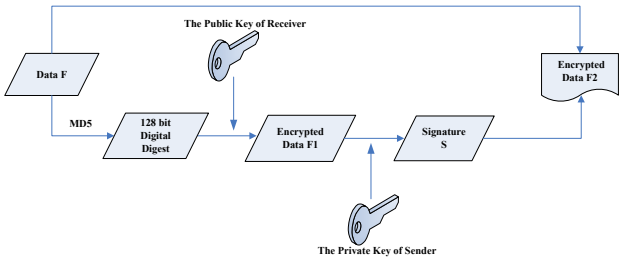

Figure 7. The deisign of sending end of plaintext data transmission

(2) Receiving end

According to the design of sending end, figure 8 below shows the design of receiving end.

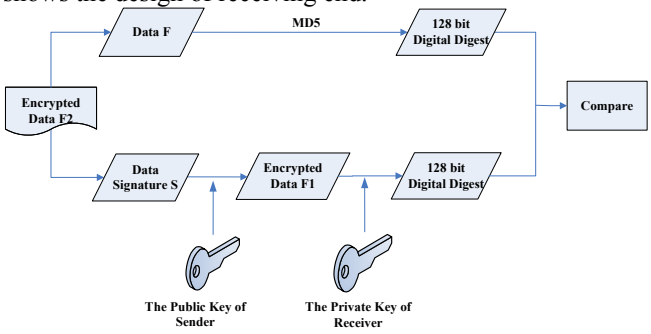

Figure 8. The deisign of receiving end of plaintext data transmission

Receiving end receives the data F2 sent from sending end. It firstly separates the data F from signature S. Then it deciphers the signature S with the previously received RSA public key from sending end, and decrypts ciphertext data F1 with its own RSA private key(see figure 8) to get a 128-bit digital digest. Meanwhile, it uses MD5 to calculate the received digital digest of data F, and compares these two digests. If they are same, then it shows that the data have the completeness, and the signature from sending end could demonstrate the non-repudiation of the data. Otherwise, they need to be downloaded, and re-authenticate the data.

VI. CONCLUSION AND FUTURE WORK

The accompanying problem of a distributed system is the data transmission. It has become a very important problem in the areas of data transmission. That is to say that the design of secure data transmission strategy is challenging. Currently, most widely used transmission solutions usually have very good applicability, and they are used in the data transmission in the distributed system. Obviously, it has functional redundancy, thus influencing the overall operational efficiency.

Based on the above analysis, we designed and realized a data transmission solution in the heterogeneous environment. This solution supports the leveled encryption of data transmission. It is able to transmit not only through encryption, but also through plaintext. Users may choose the most appropriate transmission method according to the requirements of the data security and speed tolerance, to satisfy the requirements of data security and speed by users. In the encryption part of our solution, it uses AES symmetrical encryption algorithm to encrypt the data to be transmitted, and uses RSA non-symmetrical encryption algorithm to manage AES encryption key, to solve the inconvenience in the transmission of symmetric algorithm. The data is finally transmitted in digital signatures, thus it ensures the data security, completeness and non-repudiation.

For those data with high sensibility, their own value is relatively higher than the normal data, so it is more concerned than the normal ones. When they are transmitted through the network, they are more likely to be intercepted by illegal users. But the data with high sensibility could adopt the encrypted transmission. Although illegal users also

may intercept them, the intercepted data are messy ones. The illegal users could not know, or at least could not know right away about the plaintext. Thus it protects the security of the data. But the illegal users may discard the intercepted data, forge a signature, and send this forged signature to receiving end. Though the forged signature is almost impossible to pass the authentication of the receiving end, it is still a hidden risk for the transmission speed. Therefore, if it is possible to add a digital certificate from a credible organization to complete a strict identity authentication, then this solution would be more perfect. But the transmission speed will decrease with the increasing security factors. Therefore, how to improve this transmission solution without influencing or less influencing the transmission speed, is further to be researched and discussed in the future work.

ACKNOWLEDGMENT

This work was supported by the Doctoral Foundation of University of Jinan (Grant No. XBS1237) and A Project of Shandong Province Higher Educational Science and Technology Program (Grant NO. J12LN44). Ajith Abraham acknowledges the support from IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 funded by Structural Funds of the European Union and state budget of the Czech Republic

REFERENCES

[1] Wei X. Heterogeneous Database Integration Middleware Based on Web Services. Physics Procedia, vol.34,part B,2012,pp.877-882.

[2] Kun Ma, Chengping Fang. A Security Extension Framework Based on SOAP Header. Journal of Information & Computational Science, vol.9, no. 17 2012, pp. 5249-5256.

[3] Marc Brittan, Janusz Kowalik . Autonomous performance and risk management in large distributed systems and grids. Advances in Parallel Computing, vol.14, 2005, pp.225-253.

[4] Huang Lin. A Transmission Security Plan based on Distributed System. Beijing: Beijing Jiaotong University,2006.

[5] JSON-RPC. http://en.wikipedia.org/wiki/JSON-RPC.

[6] Christopher F, Joel F. What are Web Services. IBM Communications of the ACM ,vol.46,issues 6,2003 ,pp. 31-33.

[7] PHPRPC. http://www.phprpc.org/zh_CN.

[8] Compare Study on PHPRPC、JSON-RPC and XML-RPC. http://www.coolcode.org/?action=show-&id=185.

[9] HPROSE. http://www.hprose.com.

[10] Hamdan O A,Zaidan B B,Zaidan A A, et al. New Comparative Study Between DES, 3DES and AES within Nine Factors. JOURNAL OF COMPUTING,vol.2,2010 ,pp.152-156.

[11] Majithia Sachin and Dinesh Kumar. Implementation and Analysis of AES, DES and Triple DES on GSM Network. IJCSNS International Journal of Computer Science and Network Security, vol. 10, 2010 ,pp.298-303.

[12] HeeSeok Kim, Seokhie Hong, Jongin Lim. A Fast and Provably Secure Higher-Order Masking of AES S-Box. CHES,2011,pp.95-107.

[13] Guangwei Chen, Bo Yang, Kun Ma, Zhenxiang Chen, Universal data interactive interface model based on publish/subscribe. In Journal of Huazhong University of Science and Technology (Natural Science Edition). vol. 40. no. S1, 2012, 141-145

[14] Guangwei Chen, Bo Yang, Kun Ma, Zhenxiang Chen. A Universal Data Interactive Interface Model and Its Application. Journal of University of Jinan (Natural Science). vol. 27, no. 1, 2013, 1-5.