EFFECTIVE METHOD FOR SECURED SOAP BASED WEBSERVICE INTERACTION USING JASYPT

A.Sudha M.Tech, AP/CSE
Department of computer science and Engineering,
Al-Ameen Engineering College,
Erode.

K.Vanitha M.E., (PhD), AP/CSE
Department of computer science and Engineering,
Al-Ameen Engineering College,
Erode.

A.NooralShaba M.E, AP/CSE
Department of computer science and Engineering,
Al-Ameen Engineering College,
Erode.

***Abstract*— several applications are developed with SOA techniques, so security for the web services in heterogeneous platform is very important. In order to secure the web services wide set of security requirements need to be considered for providing authentic secure transfer of message over the internet. In order to solve the security issue of heterogeneous platforms, a security processing model named JASYPT (Java simplified encryption Technique) based on SOAP and authentication is proposed in this paper. By this model it ensures the safety of SOAP message transmission and enhances the security of Web service in heterogeneous platforms.**

Index Terms— SOAP, Heterogeneous, Web Service, Security Interaction.

I INTRODUCTION

Service-Oriented Architecture is a technique where loose coupling is mainly considered which enables the communication with various services and dynamic discovery is done so it can work on any platform. So many services use XML language for message transfer, because XML is platform independent. If the services have some limitations then it reduces the usefulness. Security is a major consideration in these environments, since they are dealing with resources that are provided and managed by separate organizational entities across domain boundaries as discussed in the paper [1] in which web service layer uses SOAP over HTTP in which it should rely on a web portal for providing security so the browser and a Web Service was secured using a plug-in implementing the methods described in the WS-Security for protecting SOAP messages. While transferring the message over internet two type of security can be provided one is transport level security and the other is message level security.

To provide the security in transport level is very difficult because the message passes through several intermediate nodes, so message level security can be provided by several encryption techniques [3]. So it improves the flexibility of the system.
Web services are open standard (XML, SOAP, and HTTP etc.) based Web applications that interact with other web applications for the purpose of exchanging data. Web Services can convert existing applications into Web-applications. Web Services uses SOAP over HTTP protocol for the communication, so you can use your existing low cost internet for implementing Web Services. This solution is much less costly compared to proprietary solutions like EDI/B2B [9]. Beside SOAP over HTTP, Web Services can also be implemented on other reliable transport mechanisms like FTP etc.

II. SOAP (SIMPLE OBJECT ACCESS PROTOCOL)

The Web Services uses XML-based SOAP messages [2] to invoke functions and transport the data. The SOAP messages are transferred using HTTP over the Internet. When transferring over the internet sensitive and confidential data will be facing the security attacks. To ensure the message-level security in SOAP messages the Organization for the Advancement of Structured Information Standards (OASIS) defined the WS-Security standard [3]. WS-Security uses two underlying standards: XML Encryption [4] and XML Signature. These standards offer a flexible usage of security mechanisms in SOAP messages and therewith improve the Web Services technology to provide integrity, confidentiality, and authenticity. However, the use of security mechanisms in SOAP messages supports attacks on a Web Services. Providing security in the soap messages is not mandatory so if the client is transferring secure information some authentication features can be included in the header. This is done in order to avoid much time consumption while transferring huge data. In WS-Security data integrity and confidentiality is ensured but it is done for the entire soap message so it becomes too complex.

A. Security Features

The main concern for the developers is to provide security when transmitting the sensitive information across the internet. So it depends upon the sensitivity of the information that is

transmitted. Many protocols like HTTPS (hyper text transfer protocol) or secure socket layer protects the message when it is transmitted but it will not provide security after reaching destination, so the data can be hacked from the destination. So encryption is mainly done in order to maintain the integrity, confidentiality, and authentication. Many cryptographic security methods are also present for secure communication by using the private key, public key, digital signatures. But for a particular type of encryption, decryption is present in order to fetch the data so if the hackers know the type of encryption they can easily hack the data by fishing.

III. LITERATURE SURVEY

A. Encryption based on Rampart

The security policy of the rampart model realizes the encryption and signatures of the soap messages. In which it has considered some special requirements in which the entire soap message is not encrypted only the password or very important values are alone encrypted so user defined security policy is considered and validation is done. On the server side decryption and verification takes place. Users can know about the encryption of the message by the client. After applying the security policy on the server side the encrypted message has two parts in which <X.509IssuerName> specifies the certificate information so it specifies that the message is more secure and non-encrypted part of soap message is not handled safely so it does not have any description on the header and the content in the <Cipher Value> is the encrypted information the soap header as discussed in the paper [5].

B. Rijndael

The implementation of flexibility for Rijndal in the paper [6] mainly focuses on internet information services in which the Rijndal encryption method is used , in which at the cost of the extra transformation in key schedule decryption can be done with same structure but different components . To get the maximum throughput a word length of 32 bits or different combinations are also considered.

IV PROPOSED SYSTEM

A. JASYPT

In this techno savvy era where we are using rapid technology changes at the same time market leaders are innovating new apps or services in order to cut the cost and reduce the time to market to have a cutting edge than with their competitors. In this paper we proposed architecture for providing a interoperable platform independent way of providing web services, to demonstrate our concept we have adapted the axis based framework for developing our web services and we have developed a security framework based on Jasypt were we take the advantage of java language (i.e. write once run anywhere).In order to prevent security for sensitive information, we are using the JASYPT as the metric to send the secure information over the internet. Normally when sending the messages over internet or when doing any business transaction ,the hacking, fishing are done to get the sensitive information like password , pass code and some important numbers . There are many cryptographic techniques that can be adopted but for every encryption there is a related decryption. Due to this hacking is very easily done. So the Jasypt security framework proposed in this architecture (fig: 4.1) is to address this issue where we are encrypting the clients sensitive data which are entered on the fly through the different forms in the browsers data provided by the touch screen in the atm machine though they are provisions to encrypt data, so every time a different encrypted key is obtained and it is checked only by the tool in the server side.

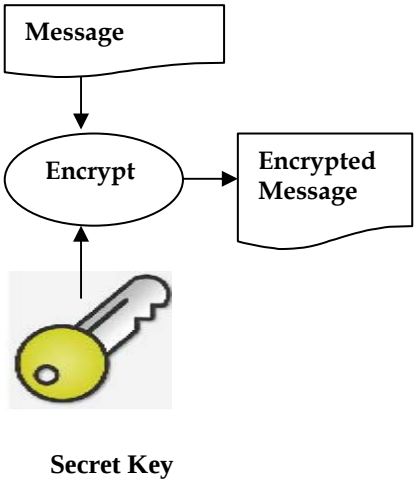

Fig 4.1 security techniques using encryption.

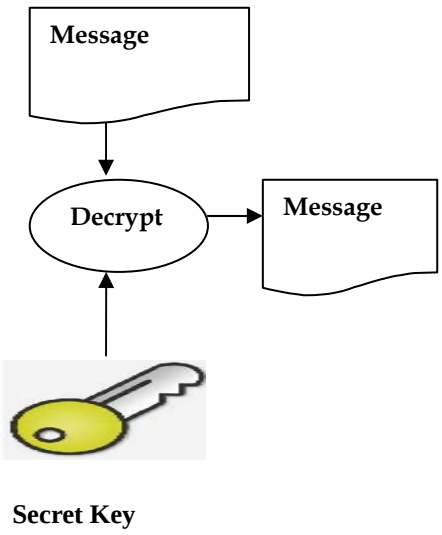

Fig 4.2 Security techniques using decryption

The hackers may don't have an option to encrypt password and hack information. Decryption of sensitive information methods becomes impossible. This can be made by adopting the technique of Jasypt in providing security to the password or any sensitive information over the network shown in fig (4.2).

B. FEATURES OF JASYPT

1. Jasypt follows the RSA standards for password-based cryptography, and provides you with both unidirectional and bidirectional encryption techniques.
2. Open API for use with any JCE providers
3. Higher security for your users' passwords.
4. Binary encryption support. Jasypt allows the digest and encryption of binaries (byte arrays). Encrypt your objects or files when needed (for being sent over the net, for example).

In this technique when the user types in the password the atm machine will encrypt immediately. And that password will be checked in the server end by using jasypt method of decryption. if it decrypts it enters to the account else invalid . The main advantage here is every time it generates a different encrypted text so the hackers will not be able to decrypt. Here only the sensitive information like the password is encrypted because when the full text message is encrypted it takes time and the performance reduces. Https protocol which again can easily decrypt by the hackers with the help of plethora of tools available in market. The framework we proposed to addresses the above pitfall by taking the advantage of Jasypt features (we can develop our own cryptography based algorithm to encrypt and decrypt sensitive data) as shown in (fig: 4.3).

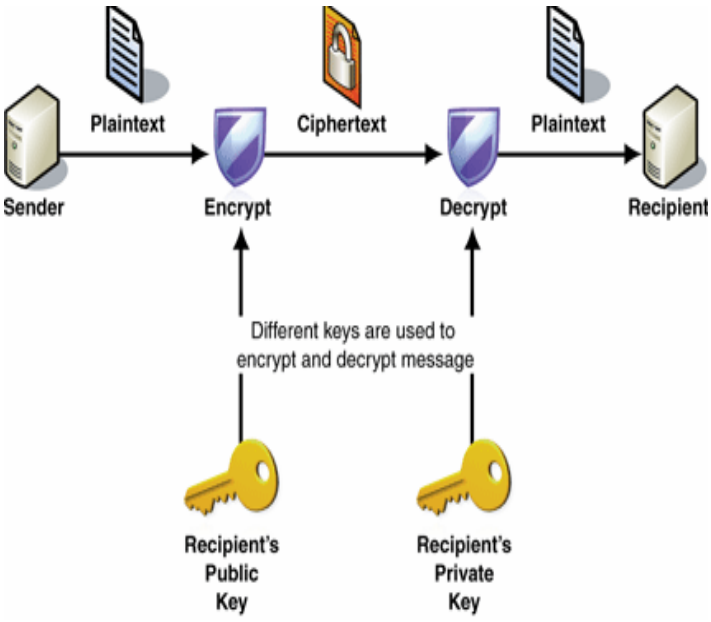

Fig 4.3 encryption and decryption using Jasypt

C. Need for Encryption and Decryption

Sending sensitive messages and files over the Internet is very dangerous as all emails are transmitted in a form. If you need to send sensitive information over the Internet you should encrypt it first. With Encryption and Decryption you can safely send sensitive messages and files. Encryption and Decryption works with both - text information and files. Just select what you want to encrypt, and Encryption and Decryption helps you keep documents, private information and files in a confidential way. Encrypt sensitive information to protect your privacy. Nowadays when more and more sensitive information is stored on computers and transmitted over the Internet, we need to ensure information security and safety.

V. IMPLEMENTATION OF PROPOSED SYSTEM

A. Steps for encryption and decryption using jasypt

1. Initially the password given to the users that can be decrypted for entering into the account transaction, so care should be taken for not falling in wrong hands.

2. Enter the username and password to login the account. Shown in fig 4.1

3. If the entered username and password is correct it enters into the banking services as given in the Fig 4.1(login details)

4. The pin number entered by the user is encrypted by using jasypt technique as in Fig: 4.2(pin number encryption)

5. The encrypted pin number is stored in the hash table, each and every time a different encrypted key is generated for more security, so the hackers cannot apply a particular decryption to attack sensitive information. When the correct pin number is given it enters into the account and soap request and response is for that particular information will be generated.

6. When the server decrypts the pin number it should be compared with the stored password hash table. If the user enters correct pin number the transaction is successful.

7. When the user enters incorrect pin number the transaction is unsuccessful and generates wrong response code with Wrong response code.

8. For the correct pin number entered by the user the encrypted key is generated by Fig 4.3(encryption).

B. Soap Request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://app.ser.com"
xmlns:q1="http://model.ser.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance">
 <soapenv:Body>
 <q0:enquireAccount>
  <q0:inputObj>
    <q1:accountNo>124578234</q1:accountNo>
    <q1:serviceHeaderObj>
      <q1:clientName>SBI_VIT</q1:clientName>

<q1:clientPassword>4D/0xCtLPzb+fcrOlmg65w234</q1:clie
ntPassword>
    </q1:serviceHeaderObj>

<q1:userSecurityKey>tQ3NTYsQ1XVYSs1vbEVwxg==</q1:
userSecurityKey>
  </q0:inputObj>
</q0:enquireAccount>
  </soapenv:Body>
  </soapenv:Envelope>
```

C. Soap Response

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
" xmlns:q0="http://app.ser.com"
xmlns:q1="http://model.ser.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
>
 <soapenv:Body>
 <q0:enquireAccount>
  <q0:inputObj>
    <q1:accountNo>124578234</q1:accountNo>
    <q1:serviceHeaderObj>
      <q1:clientName>SBI_VIT</q1:clientName>

<q1:clientPassword>4D/0xCtLPzb+fcrOlmg65w234</q1:c
lientPassword>
    </q1:serviceHeaderObj>
<q1:userSecurityKey>tQ3NTYsQ1XVYSs1vbEVwxg==</
q1:userSecurityKey>
  </q0:inputObj>
</q0:enquireAccount>
  </soapenv:Body>
  </soapenv:Envelope>
```

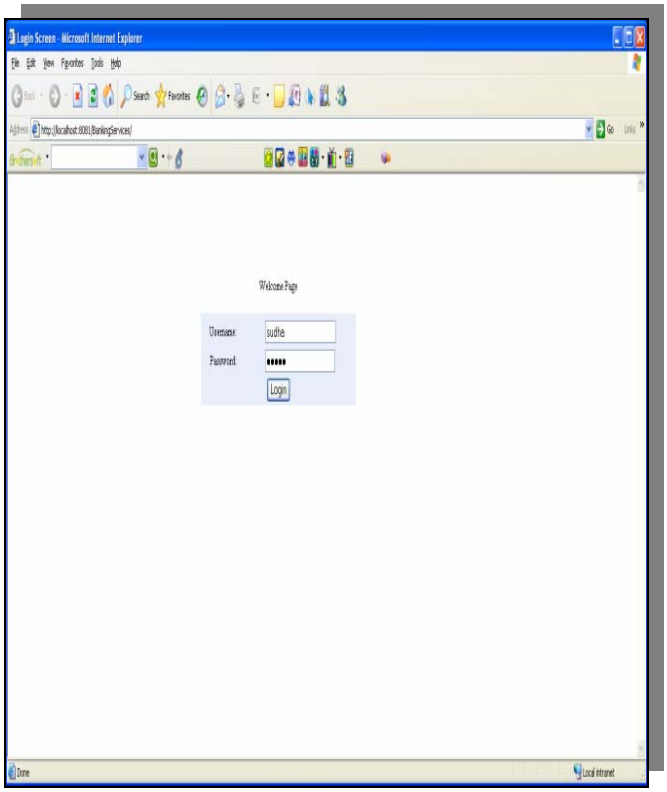

Fig 4.1 pin number encryption with login account

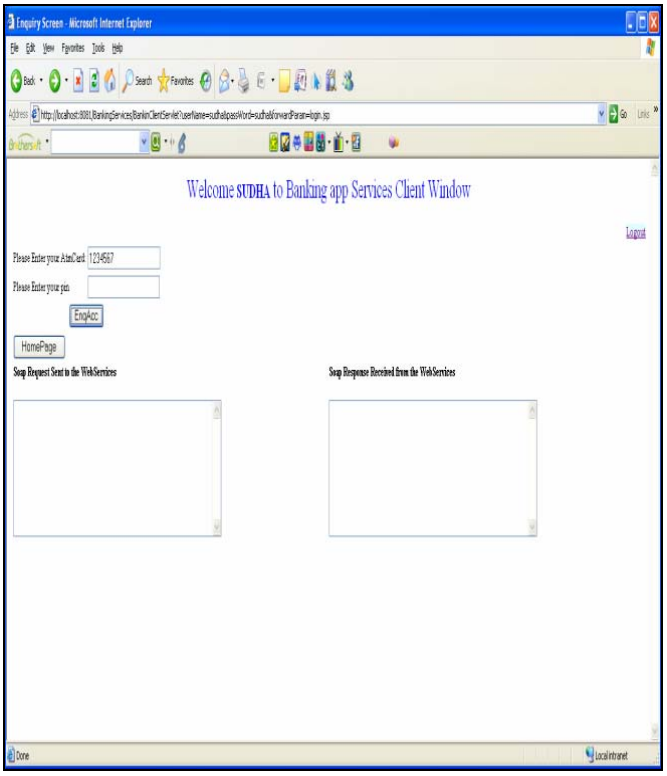

Fig: 4.2 Transaction details after successful pin number entered

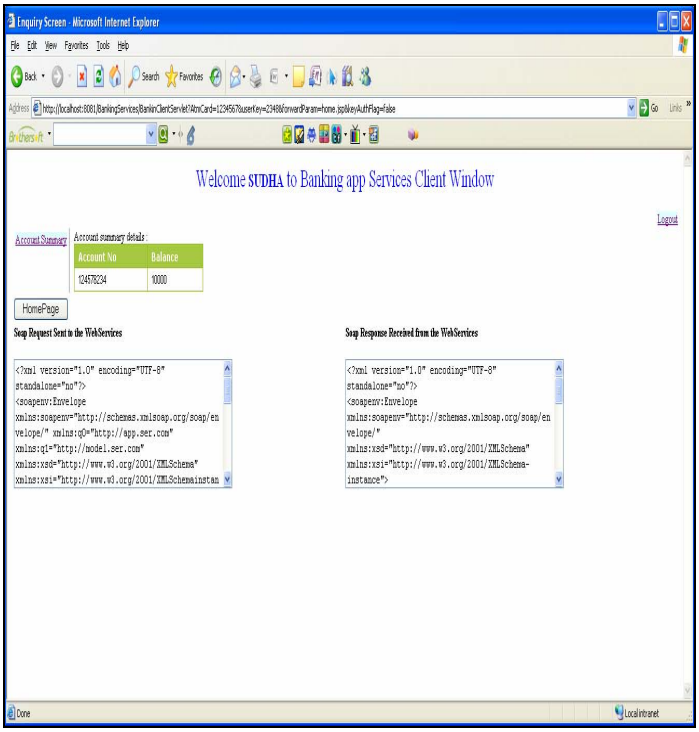

Fig: 4.3 encryption

D. Wrong Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<enquireAccountResponse xmlns="http://app.ser.com">
<enquireAccountReturn>
<accountName xsi:nil="true"/>
<accountNo>124578234</accountNo>
<accountShortName xsi:nil="true"/>
<balance>0</balance>
<serviceResponseObj>
<respnonseCode>100</respnonseCode>
<respnonseDesc>UserKey is Wrong</respnonseDesc>
</serviceResponseObj>
<userSecurityKey xsi:nil="true"/>
</enquireAccountReturn>
</enquireAccountResponse>
</soapenv:Body>
</soapenv:Envelope>
```

VI CONCLUSION

In this paper we have discussed the security requirements and challenges of securing Web services .We have also reviewed the most important efforts that addressed this problem statement by exposing their strengths and limits. As a contribution to this important concern, we have presented a framework for managing the service security, the service composition, and the service semantics. They are critical to the successful deployment of Web services. This model realizes security requirements during the process of calling Web services in heterogeneous platform. By making client authentication, signing and encrypting SOAP message in the process of Web service interaction in heterogeneous platform.

There are several areas focused for future work to further improve the presented work. Future work will focus on the policies refinement and the generalization of the reasoning technique to handle other security properties.

REFERENCES

[1] Luigi Lo Iacono, Hariharan Rajasekaran Secure Browser-based Access to Web Services, Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, IEEE Computer Society 2009, pp. 104-111.

[2] H. F. Nielsen, M. Hadley, A. Karmarkar, N. Mendelsohn, Y. Lafon, M. Gudgin, and J.-J. Moreau, “SOAP version 1.2 part 1: Messaging framework (second edition),”W3C, W3C Recommendation, Apr. 2007.

[3] K. Lawrence and C. Kaler, “Web Services Security v1.1,” OASIS Open, Tech. Rep., Feb. 2006.

[4] D. Eastlake and J. Reagle, “XML encryption syntax and Processing,” W3C, W3C Recommendation, Dec. 2002.

[5] Tao Xu, Chunxiao Yi, Web Signature and Encryption on Parts of SOAP Message Based on Rampart, W3C Recommendation, Apr. 2010.

[6] Ghossoon M. Waleed , R. Badlishah Ahmad, Security Protection using Simple Object Access Protocol (SOAP) Messages Techniques, 2011 International Conference on Electronic Design.

[7]Simple Object Access Protocol, http://www.w3.org/TR/soap.

[8] Epstein, J., Matsumoto, S., McGraw, G.: Software
Security and SOA: danger, Will Robinson! IEEE Security & Privacy, Volume 4, Issue 1, p80-83.

[9] G. H. Hwang, Y. H. Chang and T. K. Chang, "An Operational Model and Language Support for Securing Web Services", IEEE International Conference on Web Services (ICWS), 2007.

[10]Wei She, I-Ling Yen and Bhavani Thuraisingham, "Enhancing Security Modeling for Web Services using Delegation and Pass-on",IEEE International Conference on Web Services (ICWS), 2008.