# WS Security of XBRL Financial Documents Encoded by SOAP

Il-Sun Park[1] and Seung-Jung Shin[2]

[1]*Dept. of General Education Namseoul University, Chung Nam, Korea*
[2]*Dept. of Information and Technology, Hansei University, Gunpo-City, Korea*

[1]*isparkbobae@hotmail.com,* [2]*expersin@hansei.ac.kr*

***Abstract***

*Extensible Business Reporting Language (XBRL) is developed to provide an efficient and effective means of preparing and exchanging financial information over the Internet to employees, investors, and financial analysts. XBRL financial reporting services are vulnerable in security because there are no regulations of security in XBRL Standard Specification even though the Internet is unsecure in its nature. XBRL financial reporting services need end-to-end, message-level security because XBRL financial documents are transported to an ultimate receiver via multi intermediaries. However, the current security technologies which are transport-level security and point-to-point security such as SSL/TLS, S-HTTP, and VPN, are not sufficient for securing financial information or encrypting only selected portions of an information set. This paper proposes FRWS[2] security model which uses WS-Security. XBRL instance documents are first encoded by SOAP and use UsernameToken, timestamp, and nonce to authenticate. FRWS[2] is sufficient and effective for Authorization, Integrity, Confidentiality, and non-Repudiation of financial reporting services.*

***Keywords:*** *XBRL, SOAP, Message Level, Security*

## 1. Introduction

The web-based XML technology becomes the technology standard of e-Business services, as many companies want to provide corporate financial statements, including balance sheets, statements of profit and loss, and statements of cash flow, more rapidly and accurately in this rapidly developing Internet environment. With eXtensible Business Reporting Language (hereinafter, "XBRL"), corporate financial information needs to be entered into the Internet-based tools just once to enable public announcement of various corporate information that the user requests. It secures transparency in corporate management, while providing rapid and accurate financial information to the corporate information user, such as a bank or an investor.

XBRL specification, however, does not have regulations on security and furthermore, the Internet itself is exposed to security issues, making security strategies in XBRL financial reporting services very vulnerable. XBRL files should be independent of platforms and possible to encrypt only a part of the message, considering the nature of financial information. XBRL financial reporting services should also be neutral in the distributed system among vendors and independent of various applications. XBRL should be an interoperable web service, as the message is routed through several intermediaries.

In this paper, WS-Security technology is applied, instead of the conventional transport-level security, in order to meet requirements of message security, including confidentiality, integrity and certification in XBRL documents encoded by SOAP. However, considering the overhead, one of the technology problems in WS-Security, FRWS[2] security model is designed

first based on security level and then, two symmetric-key algorithms (AES and Triple-DES) are used for encryption based on differential data size for security.

## 2. XBRL Web Service Framework

Most e-commerce businesses introduce SOA (Service-Oriented Architecture) to increase interoperability among computer systems for cost-saving. Figure 1 shows the framework of XBRL financial reporting web services including major components of web services.

XBRL service provider registers XBRL service description (WSDL file) in the discovery registry through the agent. XBRL service request agent makes a XBRL service request through the UDDI interface in the discovery registry, and the discovery registry agent finds the requested service list and sends description and service semantic of the relevant service to the service request agent. With the acquired information, the service request agent sends SOAP-encoded XBRL request messages to the service provider. The service provider sends the requested XBRL financial information to the service requestor.

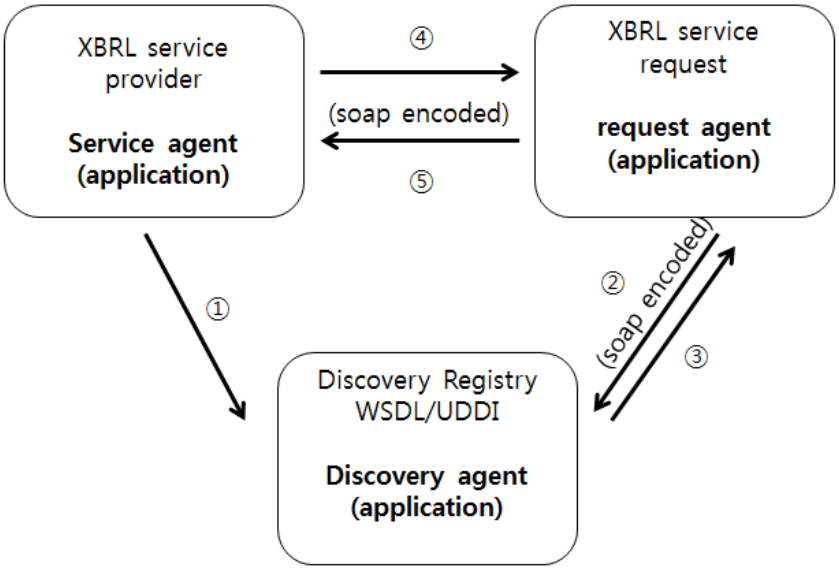


**Figure 1. The Framework of XBRL Financial Reporting Web Services**

## 3. FRWS2 Model Design

### 3.1. Problems of Conventional Security Technology

1) Limitations of Transport-Level Security

Currently used in web-service security, transport-level security, such as SSL (Secure Sockets Layer)/TLS (Transport Layer Security) and VPN, is useful for point-to-point communication. In the case of XBRL web services, however, the message is routed through several points, which requires security throughout the entire communication between the service provider and the end service user. In addition, XBRL web services use other protocol transports, including HTTP, as well as TCP, SMTP, FTP, IBM MQSeries and MSMQ, while SSL security is only limited on HTTP, making it vulnerable to be used for XBRL web service security.

2) Problems of Encoding Process Technology

In marshaling the data, SOAP defines more than one encoding methods and convert them into XML format. Therefore, the system performance may vary depending on SOAP encoding methods selected. Elsevier, in a project in 2003, confirmed that the bigger the message is, the drastically lower the transaction per second (TPS) is, in RPC encoding.

3) Problems of Overhead in SOAP Header

As WS-Security puts various security codes in the SOAP message header and uses the XML encryption method, along with XML digital signature, the size of the SOAP message header increases and faster CPU, bigger memory and wider transmission bandwidth are required. In 2005, Hongbin Liu made a WS-security process test, with 25 types of SOAP messages in various sizes. The results showed that WS-Security signature and encryption method generated about 9 times higher overhead.

### 3.2. FRWS2 Model Design

Considering limitations of the aforementioned transport-level security and problems of SOAP overhead, this paper proposes the phased FRWS2 (Financial Reporting Web Service Security) Model to provide efficient and integrated security mechanism for XBRL financial reporting service [Figure 2].

FRWS2_A Model is a security model that can be only operated within the organization and UsernameToken and time stamp is added in authentication.

In FRWS$^2$_S Model, XBRL financial reporting service is connected to outside organizations and XML digital signature digest is included in addition to authentication to secure integrity.

FRWS$^2$_AE Model considers active integration with outside organizations, openness of web services and multiple agents, while maintaining end-to-end security and both XML signature and XML encryption are applied in addition to authentication.

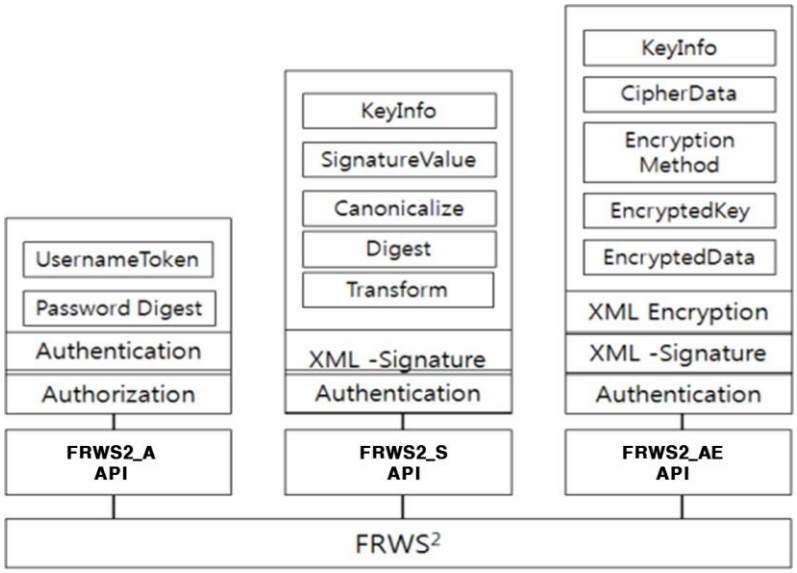


**Figure 2. FRWS2 Security Model**

## 4. Tests and Results Analysis

For the test, soapUI Pro was used to simulate the XBRL SOAP message request and response. Then, a log file was generated to analyze authentication, time stamp, encryption, and response time after request, based on message sizes, algorithms and security models followed by authentication and encryption. The password used for authentication was a digest-type and the nonce value was generated to encode it in base64 binary. mustUnderstand properties in wsse:Security was set to 'True' to ensure that access to the main body is denied, if the header's encryption information is not accurately decrypted. To prevent additional modulation factors, the time stamp period was set to 1 minute. In addition, the load test time limit was set to 120 seconds, thread to 1, and test delay to 1000 milliseconds for basic properties. After requesting a simple login service that includes XBRL financial documents, the total request count within the time limit, the maximum response time, the minimum response time, the average response time and the transaction per second (TPS) was checked.

Table 1 shows the analysis results that indicate differences in performances by message sizes when only UsernameToken and nonce value are added in authentication in $FRWS^2$_A Model. When the message size was 3993 bytes, the request response time was shorter and TPS was high, compared to relatively big message size of.

**Table 1. $FRWS^2$_A Result by Size**

| Authentication by Size | $FRWS^2$_A_user_3993 | $FRWS^2$_A_user_5325 | $FRWS^2$_A_user_8885 |
|---|---|---|---|
| Min. Response Time | 220 | 209 | 220 |
| Max. Response Time | 262 | 1490 | 305 |
| Total Request Count | 123 | 122 | 120 |
| Average Response Time | 238.52 | 242.5 | 255.82 |
| TPS | 1.01 | 1.01 | 0.99 |

When the time stamp is added to $FRWS^2$_A Model, the average response time of 3993 bytes was the shortest (291) and that of 8885 bytes, which has the biggest message size, was the longest (317.87), while TPS of 8885 bytes was the lowest (0.91). [Table 2]

**Table 2. $FRWS^2$_A Result by Size and Time Stamp**

| Authentication by Size/Time Stamp | $FRWS^2$_A__3993 | $FRWS^2$_A_5325 | $FRWS^2$_A__8885 |
|---|---|---|---|
| Min. Response Time | 235 | 254 | 208 |
| Max. Response Time | 415 | 843 | 1348 |
| Total Request Count | 115 | 112 | 111 |
| Average Response Time | 291 | 299.31 | 317.87 |
| TPS | 0.94 | 0.92 | 0.91 |

Next, to compare differences in performances by authentication and encryption, the same message size was applied. Table 3 shows the load test results of FRWS2_A_user_7996, an authentication model that includes time stamp and nonce value, and FRWS2_AE_7996 model that also includes encryption in addition to authentication. With authentication alone, the average response time was 256.27 milliseconds, remarkably faster than the average response time with both authentication and encryption at the same time (327 milliseconds). TPS in the case of simultaneous authentication and encryption was 0.94, showing a great difference from 0.98 with authentication alone.

**Table 3. FRWS$^2$_A Result by Authentication and Encryption**

| Authentication/Authentication and Encryption | FRWS$^2$_A__user_7996 | FRWS$^2$_AE_user_7996 |
|---|---|---|
| Min. Response Time | 212 | 232 |
| Max. Response Time | 569 | 1555 |
| Total Request Count | 118 | 113 |
| Average Response Time | 256.27 | 327 |
| TPS | 0.98 | 0.94 |

For encryption, the difference in performances by message size was analyzed. As shown in Table 4, the same encryption algorithm -AES- was used, with different message sizes. The average response time of the bigger message was longer and its TPS was greatly reduced.

**Table 4. FRWS$^2$_AE Result by size**

| Encryption by Size | FRWS$^2$_AE_10676 | FRWS$^2$_AE_17542 |
|---|---|---|
| Min. Response Time | 239 | 215 |
| Max. Response Time | 2006 | 661 |
| Total Request Count | 114 | 100 |
| Average Response Time | 297.08 | 442.66 |
| TPS | 0.94 | 0.83 |

To analyze the difference in performances by algorithm, AES was used for FRWS$^2$_AE_7996, and Triple-DES was used for FRWS$^2$_AE_7996_t. As shown in Table 5, the average response time in the case of encryption by AES algorithm was 230.08, considerably faster than the case encrypted by Triple-DES algorithm, and TPS also showed great difference between the two.

**Table 5. FRWS$^2$_A Result by algorithm**

| Encryption by Algorithm | FRWS2_AE_7996 | FRWS2_AE_7996_t |
|---|---|---|

| | | |
|---|---|---|
| Min. Response Time | 209 | 232 |
| Max. Response Time | 467 | 2103 |
| Total Request Count | 123 | 107 |
| Average Response Time | 230.08 | 393.68 |
| TPS | 1.01 | 0.88 |

## 5. Conclusion

WS-Security, a security strategy for SOAP documents in the interoperable web services under distributed computing environment, is a more dynamic security mechanism than XML, designed to integrate security methods into SOAP messages.

According to the test results, differences in performances were displayed by message sizes in all security models: the bigger the message was, the longer the request response time and the lower the TPS was. In the case of authentication, the difference in performance by the message size was not considerable. In the case of encryption, however, the difference was remarkable by the message size, and in particular, with Triple-DES algorithm, the request response time was much longer and TPS was much lower, compared to AES algorithm. In the case that encryption was added along with authentication, the average response time was considerably long (327 milliseconds), compared to authentication alone (256.27 milliseconds). The test results confirmed that the stronger the security becomes, the more the overhead is generated and the longer the response time is.

For future research tasks, it is required to make an analysis by asymmetric encryption algorithm, and an analysis with application of various agents and communication protocols.

## References

[1] Y. -h. Namgung, D. -h. Park, S. -h. Heo and D. -g. Pack, “XML Data Encryption System Design and Implementation in consideration with Effectiveness”, KIISC, vol. 29, no. 6, **(2002)**.
[2] B. -h. Park and J. -i. Lee, “Research of SOAP Message Security Technology for Safe Web Service”, KIISC review, vol. 14, no. 4, **(2004)**, pp. 10-18.
[3] S. -j. Shin and H. -s. Kim, “Research of Message Transport Protocol in consideration of Document Class”, Information Technology and Database Journal, **(2002)**.
[4] A. Nadalin, IBM, C. Kaler, Microsoft, R. Monzilo, Sun, P. Hallam-Baker, Verisign, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard Specification, **(2006)** February 1.
[5] B. Dournaee, "XML Security", RSS Press, **(2002)**.
[6] F. Cohen, CEO, "Discover SOAP encoding's impact on Web services performance", **(2003)** March, IBM developerWorks.
[7] IBM and Microsoft, "Reliable Message Delivery in a Web Services World", IBM and Microsoft white paper, **(2003)** March.
[8] J. Rosenberg and D. Remy, "Securing Wdb Services with WS-Security", SAMS, **(2004)**.
[9] J. E. Boritz and H. Bidgoli, (ed.), John Wiley “Business Reporting with XML: XBRL (Extensible Business Reporting Language)”, Publication in encyclopedia of the Internet, vol. 3, **(2004)**, pp. 863-885.
[10] J. E. Boritz and G. Won, “Security in XML-based financial reporting services on the Internet”, Journal of Accounting and Public Policy, vol. 24, **(2005)**, pp. 11-35.
[11] MSDN, 2003a, “Reliable message delivery in a Web Services world: a proposed architecture and roadmap”, **(2004)** February 13.
[12] MSDN, 2003b, “Secure, reliable, transacted Web services: architecture and composition”, **(2004)** February 9.
[13] R. Howard, “SOAP Message Encryption”, Microsoft Corporation, **(2001)**.
[14] S. Thompson, "WS-Security Implementation", IBM, **(2003)**.

[15] S. Guest, Microsoft Corporation, "WS-Security Interoperability using WSE 2.0 and Sun JWSDP 1.4 ", **(2010)**.
[16] S. Hada and H. Maruyama, "SOAP Security Extensions", Yokyo Research Laboratory, IBM Research, **(2000)** November.
[17] "SOAP Version 1.2 Part 1: Messaging Framework" (Second Edition).
[18] W3C Recommendation, **(2007)** April 27.
[19] http://www.w3.org/TR/2001/WD-soap12-part1-20011217.

## Authors

**Il-Sun Park**

She received the Ph.D. degree from Hansei University in 2011, Korea. Currently, she is a part time professor at Department of General Education, Namseoul University. Her current research interests include security and privacy.

**Seung-Jung Shin**

He received the Ph.D. degree from Kookmin University in 2000, Korea. Currently, he is a professor at Department of Information & technology, Hansei University. His current research interests include security and privacy (Corresponding author of this paper).