International Journal of Control and Automation
Vol.7, No.1 (2014), pp.391-400
http://dx.doi.org/10.14257/ijca.2014.7.1.35

# A Study on Detection Techniques of XML Rewriting Attacks in Web Services

Aziz Nasridinov[1], Jeong-Yong Byun[2] and Young-Ho Park[1]

[1]*Department of Multimedia Science, Sookmyung Women's University*
[2]*School of Computer Engineering, Dongguk University at Gyeongju*

*{aziz, yhpark}@sm.ac.kr, byunjy@dongguk.ac.kr*

***Abstract***

*Making Web Services secure means making SOAP messages secure and keeping them secure wherever they go. Several security standards of Web Service Security (WS-Security), such as XML Digital Signature, are used to secure SOAP messages exchange in Web Service environment. However, the content of a SOAP message, protected with XML Digital Signature, can be changed without invalidating the signature. In this paper, we present a study on detection techniques of XML Rewriting attacks in Web Services. We first explore the XML Rewriting Attack that can take place in Web Service communication. We further investigate detection techniques and describe their limitations. Finally, we discuss general countermeasures for prevention and mitigation of XML Rewriting attacks.*

***Keywords:*** *Web Services, SOAP message, XML Digital Signature, XML Rewriting attacks*

## 1. Introduction

Web Services are platform and language independent interfaces that enable communication with other applications using standard network technologies, such as HTTP and XML [1]. Due to Web Services' platform and language independent nature, many business corporations, such as Amazon and Yahoo, have used them for the integration of various applications. However, despite the growing use of Web Services by many corporations, security is still a major concern that is slowing their deployment, according to a new survey of senior IT executives sponsored by Computer Associates [2]. One key finding in this survey was that 43% of the respondents felt that security was the most significant issue related to the deployment of Web Service applications.

The communication between Web Services occurs via the Simple Object Access Protocol (SOAP) messages. Thus, making Web Services secure means making SOAP messages secure and keeping them secure wherever they go [3]. Several security standards of Web Service Security (WS-Security), such as XML Digital Signature, are used to secure SOAP messages exchange in Web Service environment. However, McIntosh and Austel [4] demonstrated that the content of a SOAP messages, secured with XML Digital Signature, can be changed without invalidating the digital signature. This is so called *XML Rewriting attacks* or *XML Signature Wrapping attacks* can occur because XML Digital Signature does not protect the location of the signed element within the XML document. This allows attacker to re-locate the signed element of SOAP messages while keeping the signature valid.

In this paper, we present a study on detection techniques of XML Rewriting attacks in Web Services. We first investigate the XML Rewriting Attack that can take place in Web Service communication. Different detection techniques are proposed to solve this problem. All of

ISSN: 2005-4297 IJCA

these techniques can detect XML Rewriting Attacks in some cases, however, they fail to handle some other cases. We then explore detection techniques by categorizing them into three groups, such as the policy-based approaches, the inline approaches and the string-based approaches, and describe their limitations. Finally, we discuss general countermeasures for prevention and mitigation of XML Rewriting attacks.

The rest of the paper is proceeds as follows. Section 2 presents preliminaries for our research. Section 3 explains XML Rewriting Attack. Section 4 describes the detection approaches of XML Rewriting attacks. Section 5 discusses the limitations of these approaches and presents countermeasures for prevention and mitigation of XML Rewriting attacks. Section 6 highlights conclusions.

## 2. Preliminaries

In this section, we present preliminaries for our research. We first give an introduction to the Web Services and SOAP messages. Then, we explain XML Digital Signature.

### 2.1. Web Services

The World Wide Web consortium (W3C) defines Web Services as following [5]: a Web Service is a software system identified by a Uniform Resource Identifier (URI), whose public interfaces and bindings are defined and described using Extensible Markup Language (XML). These systems may then interact with the Web Service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols. In particular, a service provider uses Web Service Description Language (WSDL) to describe the functionality which a service offers and publish the description in Universal Description Discovery and Integration (UDDI). On the other hand, a service requester discovers the service in UDDI and consumes it by sending Simple Object Access Protocol (SOAP) messages. Above mentioned definition can be drawn as Figure 1.

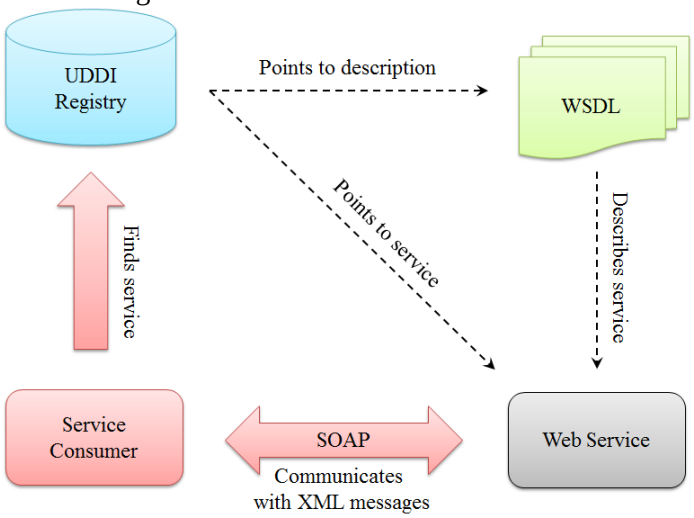


**Figure 1. Web Service architecture**

### 2.2. SOAP Messages

The communication between different Web Services occurs via the SOAP messages. SOAP 1.2 specification defines SOAP messages as following [6]: SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed

 

environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

Figure 2 illustrates a sample XML tree representation for the SOAP message. A SOAP message consists of an Envelope element that identifies the XML document as a SOAP message, an optional Header element containing routing or security information, a mandatory Body element containing a request and response and a Fault element containing errors and status information.

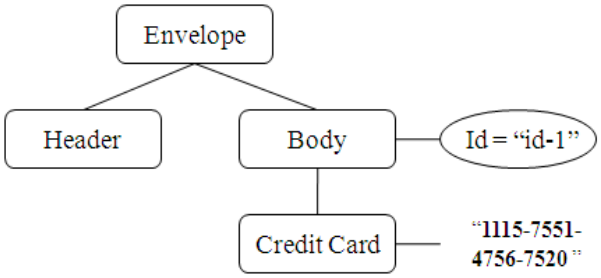


**Figure 2. XML tree representation for the SOAP message**

### 2.3. XML Digital Signature

Traditionally, transport level security mechanisms like Secure Socket Layer (SSL) and Transport Layer Security (TLS) can be used to secure SOAP message in Web Service environment. These mechanisms create a security tunnel for communication between the two end points, and preserves message integrity and privacy while the message is in transit. However, these security mechanisms are not suitable with SOAP's message-based architecture, where intermediaries cannot see the contents of the tunnel, and so cannot route or filter messages [6].

In order to overcome the above limitation of transport level security mechanisms, OASIS (Advancing open standards for the information security) released a security standard that can be used to achieve end-to-end security in Web Services called Web Services Security (WS-Security). WS-Security is the security mechanism for Web Service working in message level. It adopted XML Digital Signature and XML Encryption mechanisms to ensure that messages are secured during in transmit.

XML Digital Signature provides a method of digitally signing the SOAP messages to ensure their integrity. XML Digital Signature is suitable to SOAP messaging framework because its ability to sign only specific parts of the XML tree rather than the whole document. This feature is useful when a SOAP message travels through different intermediaries, each signing only those elements relevant to themselves. This flexibility will also be important in situations where it is important to ensure the integrity of certain parts of an XML tree, while leaving open the possibility for other parts of the document to change.

An example of a SOAP message secured with a XML Digital Signature is depicted in Figure 3. In Figure 3, XML Digital Signature (displayed as a red dashed region) 2-tuple signature where the signed soap:Body element (displayed as red solid region) is referenced by its URI under the ds:Reference element.

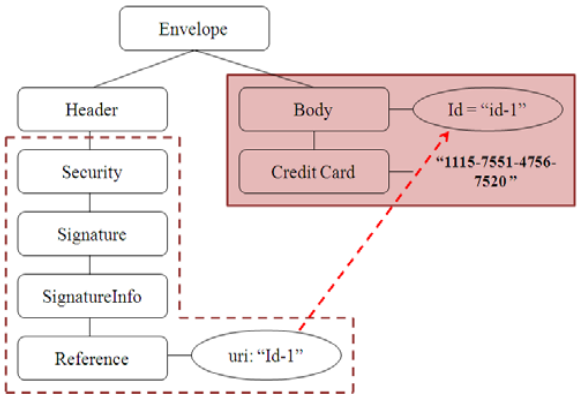


**Figure 3. SOAP message secured with XML Digital Signature**

## 3. XML Rewriting Attack

Recall from Section 1 that the content of a SOAP message, secured with XML Digital Signature, can be changed without invalidating the signature. This is called XML Rewriting attack. In this section, we will demonstrate XML Rewriting Attacks that can take place in Web Services communication.

An example of a XML Rewriting attack on SOAP message is depicted in Figure 4. We show the XML Rewriting attack on SOAP message secured with XML Digital Signature in Figure 3. In Figure 4, after the attack, Body element is moved under a new atk:BOGUS header element, making this soap:Body element meaningless for the receiver. Attacker creates his own soap:Body element with a new id and content information. In discussed figure, the result of this modification is depicted as a black solid region. According to SOAP specification the new soap:Body element is not required to have an identifier. This signature is still valid because the referencing element "body" is still present and not changed. Thus, the attacker bypasses the detection checks.

In [7, 8], it is demonstrated the application of the different types of the XML Rewriting attacks on SOAP message, using the same method demonstrated in Figure 4. Also, the practical impact of the XML Rewriting attack is presented in [9, 10], wherein the attacks on the Amazon EC2 SOAP and the Eucalyptus Cloud Web Services are demonstrated.

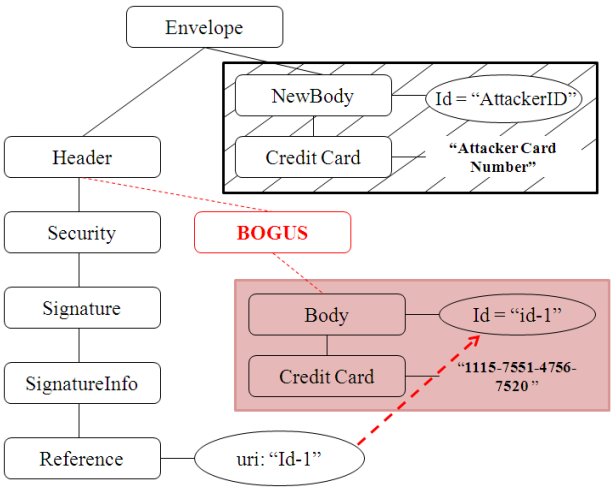


**Figure 4. XML Rewriting attack on Body element of SOAP message**

## 4. XML Rewriting Attack Detection in Web Services

Different detection techniques are proposed to solve XML Rewriting attacks. All of these techniques can detect XML Rewriting Attacks in some cases, however they fail to handle some other cases. In this section, we investigate detection techniques by categorizing them into three groups, such as the policy-based approaches, the inline approaches and the string-based approaches.

### 4.1. Policy-Based Approaches

Despite all of the Web Services security techniques, SOAP messages may still be vulnerable to a different class of attacks. In [4], the authors demonstrated that the content of a SOAP message, protected with XML Digital Signature, can be changed without invalidating the signature. In their paper, the authors present several examples of XML Rewriting attacks, and explain how to use security policies correctly in order to prevent attacks. Security policies are configured by showing that more sophisticated XML Rewriting attacks may avoid them, and by then improving the policy.

Used correctly, Web Service policy (WS-Policy) can prevent XML Rewriting attacks by enforcing the position of the signed element in policy files. Thus, advisor tools for Web Services policies are proposed in [11, 12]. A policy-based advisor tool, presented in [11], diagnoses SOAP messages according to a policy file, explain the risks and suggest appropriate recovery actions. It generates a security report by running queries that check for over thirty syntactic conditions. For instance, if the policy file specifies that <MessageID> element is optional or it is not necessary for the <MessageID> element to be signed, the policy advisor will throw a warning stating the possible existence of replay attack and will suggest to make <MessageID> element mandatory and signed for a request message.

A policy-based method, presented in [12], is able to detect and fix typical faults in SOAP messages. In this paper, a technique called Bit-Stream is developed. It works based on the importance of SOAP elements in order to automatically detect the vulnerabilities and risks, while offering advice for higher security. The system adapts simulation-based approach which allows self-optimization of its performance in different conditions.

### 4.2. Inline-Based Approaches

Rahaman *et al.*, [13, 14, 15] proposed an inline approach that takes into account information about the structure of the SOAP message by adding new header element called SOAP account. The SOAP Account header contains the number of child element of envelope; the number of header elements; the number of references for signing element; predecessor, successor, and sibling relationship of the signed object. This SOAP Account element must be signed by the creator using its X.509 certificate. Each successive SOAP node must sign its own SOAP account concatenated with the signature of the previous node. While this approach can prevent certain XML Rewriting Attack, in [7] the authors demonstrated that SOAP Account cannot address all type of XML Rewriting Attack.

In [7] the authors demonstrate that SOAP account is not able to detect all types of XML Rewriting Attacks and present some ideas to fix the issue but do not present a complete working solution. A similar works, called RewritingHealer, are done in [6, 16], where authors extend the inline approach by proposing to take into account new characteristics of SOAP message such as the depth of information and parent elements of the signed node as well as a way uniquely identify the parents elements. While this extension provides a significant improvement, it is still vulnerable. This is due to the fact that one can move the signed

element together with its parent to somewhere so that the depth of signed element is not changed.

In [17], the authors proposed to use a new header in SOAP message containing the signed elements positions in the message. This header is added to the SOAP message after the detection of signed elements positions located in the Document Object Model (DOM) tree. Elements position is obtained using a post-order traversal by visiting left node first, then right node, and finally root node. If the attacker modifies the SOAP message, it is detected by observing the change of the location of the signed elements. Somorovsky *et al.*, [10] discusses the verification steps required to effectively validate an incoming SOAP request. It reviews the available work in the light of the discovered Amazon Elastic Compute Cloud vulnerability and provides a practical guideline for achieving a robust and effective SOAP message security validation mechanism.

Recently, Nasridinov *et al.*, [18] proposed an approach called XaT-SOAP (XML-Based Attacks Tolerant SOAP). In this approach, the authors first construct SOAP message structure as ontology, and then attach it in SOAP message's header. The attached ontology is validated in the receiving end, which allows to detect XML Rewriting attacks. Also, in this approach, all modifications on SOAP messages are written to a log. Thus, if security failures have occurred, we can check this log and recover from effect of recent successful execution. In [19], the authors proposed a self-adaptive approach to ensure the integrity of SOAP messages. As approach in [18], the proposed approach first builds SOAP message structure as ontology, and then attach it in SOAP message's header. Here, if any changes in the pattern of attack are occurred, the proposed method learn them and save in a reliable storage. This can be used later in order to detect new attacks.

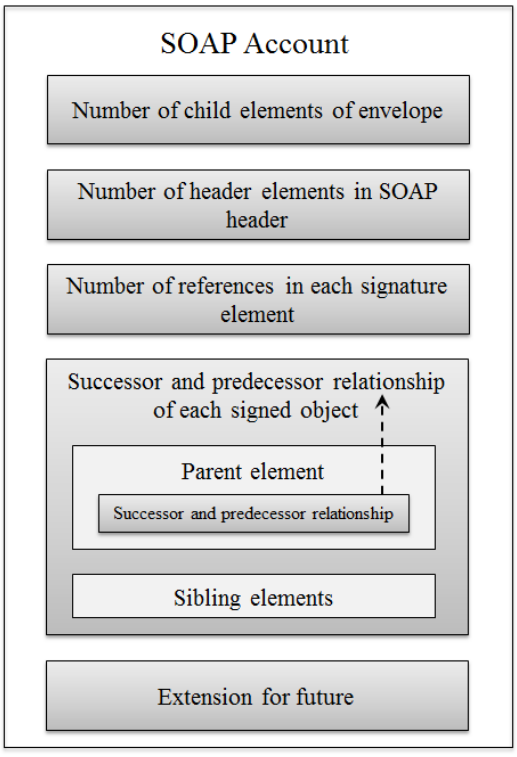


**Figure 5. SOAP Account**

### 4.3. String-Based Approaches

Gajek *et al.*, and Smriti *et al.*, [7, 8] analysed the potential scenarios that lead to the XML rewriting attack vulnerability, and they derived a new solution to the problem. Specifically, it is proposed to consider the absolute path from the root to the signed element using XPATH expressions. Specifically, authors proposed to use a subset of XPath, called FastXPath, instead of ID attributes for signature referencing in Web Services messages. They insist that if the proposed FastXPath is used, their approach is able to protect against XML rewriting attack in most reasonable scenarios without causing the performance impact associated with the use of complex XPath expressions.

### 4.4. Others

Fang *et al.*, [20] proposed FT-SOAP: Fault-tolerant Web Service framework, which protects SOAP messages in Web Service environment from security threats. In order to achieve this, they propose four major components, namely, Replication Management (RM), which performs the replication management including group constitution, and membership management Fault Detector, which performs monitoring; Fault Notifier that is responsible for the fault notification; Logging/Recovery mechanism that captures and logs requests for the recovery mechanism.

Pinzon *et al.*, [21] present the self-adaptive multi-agent architecture for dealing with Denial-of-Service (DoS) attacks in Web Service environments. DoS attacks are the class of attack, where an attacker aims to prevent legitimate users from accessing information or services. The common type of DoS attack happens when an attacker floods a network with too much requests to the target server until the server is unable to provide services to normal users. There are many methods to perform a DoS attack such as SYN flood or DTD bombing. In this paper, we deal with XML Rewriting attack. XML Rewriting attack is distinct class of attacks based on the malicious interception, manipulation, and transmission of SOAP messages. Using these types of attacks, in most cases, messages can be redirected to different location or by replaying the message, the same request can be processes several times and the server can be pushed to do redundant work.

## 5. Analysis of XML Rewriting Attack Detection Techniques

In this section, we discusses the limitations of these approaches and presents countermeasures for prevention and mitigation of XML Rewriting attacks.

Used correctly, policy-based approach can prevent XML Rewriting attacks by enforcing the location of the signed element in policy files. However, this approach is not efficient. In order to detect attacks of element deletion with policy-based approach, every element should be declared as mandatory. This reduces the flexibility of the XML document and causes the performance degradation in the validation phase. Inline approach can overcome pitfall of policy-based approach, however, is not efficient too. This is due to a high complexity of this approach, which needs a long calculation time in order to determine the structural information. The main pitfall of string-based approach is that if the depth of nodes grows, the size of the SOAP message becomes large which introduces additional overhead in processing the messages. Moreover, with the aforementioned approaches, the cost of performing attack detection is linear to the height of the XML tree (SOAP message), as with these approaches, each element of SOAP message needs to be accessed and checked.

In order to mitigate XML Rewriting attacks, we can analyze detection techniques according to the following abilities.

- Detection techniques should have detection ability that indicates their ability to catch various kinds of XML Rewriting attacks.
- Efficiency of these detection techniques is important as well. Because they have to not only detect attacks, but do so in a timely manner.
- Detection techniques should have warning analyzing ability, which would help to analyze the SOAP messages and throw a warning messages when the XML Rewriting attacks are detected.
- Recovery ability means to catch XML Rewriting attacks and recover from them. Recovery ability can be particularly useful when the Web Services operate in fault-tolerant environment.
- Detection technique should be adaptable to various XML Rewriting attacks. It has to have learning ability, which learns attack patterns and adapt to newly introduced attacks.
- The main reason why SOAP messages are used in Web Services is their flexibility. This is provided by the XML-syntax. Thus, detection techniques should be able to conserve this flexibility.

## 5. Conclusion

In this paper, we have proposed a study on detection techniques of XML Rewriting attacks in Web Services. We have investigated detection techniques by categorizing them into three groups, such as the policy-based approaches, the inline approaches and the string-based approaches, and described their limitations. Specifically, we found that policy-based approaches are not efficient, since these approaches reduce the flexibility of the XML document and causes the performance degradation in the validation phase. Inline approaches are not efficient as well due to a high complexity of these approach, which needs a long calculation time in order to determine the structural information. The main pitfall of string-based approach is that if the depth of nodes grows, the size of the SOAP message becomes large which introduces additional overhead in processing the messages. Finally, we have discussed general countermeasures for prevention and mitigation of XML Rewriting attacks. In order to detect the XML Rewriting attacks in SOAP messages, the detection mechanism should be adaptable and flexible, and have efficient detection ability, warning analyzing ability and recovery ability.

## Acknowledgement

This work was supported by the IT R&D program of MKE/KEIT. [10041854, Development of a smart home service platform with real-time danger prediction and prevention for safety residential environments].

## References

[1] H. Bae, “Web-Service Based Integration of Multi-organizational Logistic Process”, Proceedings of the Dynamics in Logistics (LDIC), **(2007)** August; Bremen, Germany.
[2] L. Liberti, “CA Survey Finds Security Concerns Slow SOA/Web Service Implementation”, CA Advisor Security Management Newsletter, **(2009)**.
[3] J. Rosenberg and D. Remy, “Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption”, Sams Publishing, Indiana, **(2004)**.
[4] M. McIntosh and P. Austel, “XML Signature Element Wrapping Attacks and Countermeasures”, Proceedings of the International Workshop on Secure Web Services (SWS), **(2005)** November; Fairfax, VA, USA.

 

[5] World Wide Web consortium, "Web Services Architecture Requirements", **(2008)**, http://www.w3.org/TR/wsa-reqs/.
[6] F. A. Kadir, "RewritingHealer: An Approach for Securing Web Service Communication", Master Thesis in KTH Royal Institute of Technology, Sweden, **(2008)**.
[7] S. Gajek, L. Liao and J. Schwenk, "Breaking and Fixing the Inline Approach", Proceeding of the ACM Workshop on Secure Web Services (SWS), **(2007)** November; Fairfax, VA, USA.
[8] K. S. Smriti and B. Azzadine, "A Formal Solution to Rewriting Attacks on SOAP Messages", Proceeding of the ACM Workshop on Secure Web Services (SWS), **(2008)** October; Alexandria, VA, USA.
[9] N. Gruschka and L. L. Iacono, "Vulnerable Cloud: SOAP Message Security Validation Revisited", Proceeding of International Conference on Web Services (ICWS), **(2009)** July, Los Angeles, CA, USA.
[10] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka and L. Iacono, "All Your Clouds are Belong to us - Security Analysis of Cloud Management Interfaces", Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW), **(2011)** October; Chicago, IL, USA.
[11] K. Bhargavan, C. Fournet and A. D. Gordon, "An Advisor for Web Services Security Policies", Proceedings of the 2nd ACM Workshop on Secure Web Services (SWS), **(2005)** November; Fairfax, VA, USA.
[12] P. P. Hung, A. Nasridinov, L. Qing and J. Y. Byun, "A Solution for Injection and Rewriting Attacks on SOAP messages in Web Services Security", Journal of KIISE: Computing Practices and Letters, vol. 18, no. 3, **(2012)**.
[13] M. A. Rahaman, M. Rits and A. Schaad, "An Inline Approach for Secure SOAP Requests and Early Validation", Proceeding of the Open Web Application Security Project Europe Conference (OWASP), **(2006)** May; Leuven, Belgium.
[14] M. A. Rahaman, M. Rits and A. Schaad, "Towards Secure SOAP Message Exchange in a SOA", Proceeding of the ACM Workshop on Secure Web Services (SWS), **(2007)** November; Fairfax, VA, USA.
[15] M. A. Rahaman and A. Schaad, "SOAP-Based Secure Conversation and Collaboration", Proceeding of International Conference on Web Services (ICWS), **(2009)** July, Los Angeles, CA, USA.
[16] A. Benameur, F. A. Kadir and S. Fenet, "XML Rewriting Attacks: Existing Solutions and their Limitations", Proceeding of the International Conference on Applied Computing, **(2008)** April; Algavre, Portugal.
[17] T. S. Barhoom and R. S. K. Rasheed, "Position of Signed Element for SOAP Message Integrity", International Journal of Computer Information Systems, vol. 2, no. 1, **(2011)**.
[18] A. Nasridinov, P. P. Hung, L. Qing and J. Y. Byun, "XaT-SOAP: XML-based Attacks Tolerant SOAP Messages", Journal of KIISE (Korea Institute of Information Scientists and Engineering): Computing Practices and Letters, vol. 18, no. 6, **(2012)**.
[19] A. Nasridinov and J. Y. Byun, "A Self-Adaptive Approach to Secure Integrity of SOAP Messages", Journal of KIISE (Korea Institute of Information Scientists and Engineering): Computing Practices and Letters, vol. 18, no. 9, **(2012)**.
[20] C. L. Fang, D. Liang, F. Lin and C. C. Lin, "Fault-Tolerant Web Services", Journal of System Architecture, vol. 53, no. 1, **(2007)**.
[21] C. I. Pinzon, J. Bajo, J. F. De Paz and J. M. Corchado, "S-MAS: An Adaptive Hierarchical Distributed Multi-Agent Architecture for Blocking Malicious SOAP Messages within Web Services Environments", Expert Systems with Applications, vol. 38, no. 5, **(2011)**.

## Authors

**Aziz Nasridinov**

Aziz Nasridinov received his B.S. degree in Computer Science from Tashkent University of Information Technology, and his M.S. and Ph.D. degrees in Computer Engineering from Dongguk University, Seoul, South Korea. He is currently working as Post-Doctoral Researcher at Sookmyung Women's University, South Korea. His research interests include Database Management Systems (DBMS), machine-learning techniques and Web Services.

**Jeong-Yong Byun**

Jeong-Yong Byunreceived the B.S. and M.S. degrees from Dongguk University, Seoul, Korea in 1980 and 1983, respectively, and the Ph.D. degree from Hongik University, Seoul, in 1994, all in computer science. From 1982 to 1987, he was with Electronic and Telecommunication Research Institute (ETRI), Daejon, Korea where he was involved in the development of UNIX systems. Since 1988, he has been with faculty of computer science and multimedia engineering of Dongguk University at Gyeonju. He had been visiting academy (by Post Doctor program of KOSEF) at University of York in England between 1995 and 1996. His researches has been concentrated on Korean alphabet according to Hunminjeongeum, Database Management Systems (DBMS), Semantic Web and Web Services.

**Young-Ho Park**

Young-Ho Park is an Associate Professor of the Multimedia Science at Sookmyung Women's University. His research interests include Database Management Systems (DBMS), Information Retrieval (IR), XML, and Telecommunication Systems. Young-Ho Park received his Ph.D. degree in Department of Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 2005. His Ph.D. research includes efficient query processing in heterogeneous XML documents. He received his B.S. and M.S. degrees in Computer Engineering from the Dongguk University in 1990 and 1992. He had worked for the Electronics and Telecommunication Research Institute ( ETRI ) as a senior research staff at the ISDN Administration & Maintenance Division for TDX-10 ISDN, the Real-Time DBMS Division and the Real-Time Operating System Division from 1993 - 1999. And, he had worked for the Advanced Information Technology Research Center ( AITrc ) , Korea Advanced Institute of Science and Technology ( KAIST ) as a Post Doctor at the from 2005 - 2006 after receiving Ph.D. degree.

 