Journal of Applied Mathematics & Bioinformatics, vol.3, no.4, 2013, 107-123
ISSN: 1792-6602 (print), 1792-6939 (online)
Scienpress Ltd, 2013

Performance of Web Services Security Mechanisms: Analysis and Evaluation

D.P. Iracleous[1,2], K. Moutsakis[2] and O.B. Efremidis[2]

Abstract

Web services interact with sensitive data such as credit card numbers, it is very important for web services to be able to ensure security. To this end, they include a wide range of technologies and specifications that when combined can provide integrity, confidentiality and authentication.

This paper discusses the basic concepts of WS-Security and the related technologies. The complexity caused by the overhead to the overall process of web services transactions is analysed and evaluated. Extended experiments have been conducted in order to measure and compare the security mechanisms.

Keywords: web services, security evaluation, encryption algorithms, WSIT, security overhead

[1] Informatics and Computer Engineering Lab, Faculty of Mathematics and Engineering Science, Department of Military Sciences, Univ. Military Sciences - Hellenic Army Academy, Vari - 16673, Greece. E-mail: iracleous@ieee.org

[2] Computer Science Department, University of Hertfordshire, IST College, Pireos 72 str., Moschato 18346, Athens, GREECE. E-mail: obe@ist.edu.gr

Article Info: *Received* : August 1, 2013. *Revised* : October 20, 2013.
Published online : December 10, 2013.

1 Introduction

Web services are a way of integrating applications by overriding their specific implementation or platform by using technologies and protocols such as SOAP, XML, WSDL [1] [3]. The usage of web services to sectors such as e-commerce, e-banking and other applications that manipulate sensitive data and information require special attention and it is very important to minimize the security risks that may occur by transferring these kinds of data [2].

Two types of security mechanisms can be defined; transport level security and message level security. Transport level security includes SSL/TLS (Secure Sockets Layer/Transport Layer Security) [4] [5] [6] [7] [8].

Even though SSL is very fast and performs very well at securing connections it fails to provide the kind of security that the web services require. SSL is a point-to-point security mechanism and can't guarantee end-to-end security. Another characteristic of SSL is that it encrypts all the information that is transmitted and cannot encrypt a specific part of information. This is a problem for big piece of information as it is known that cryptography is very intense for the CPU and can cause serious performance issues. Furthermore web services can be used with a variety of protocols such as FTP, SMTP and TCP. SSL is not able to provide security to all of these transmission mechanisms thus reducing web services potential. Although SSL can be proven insufficient for securing a web service, it can be used in conjunction with other security mechanisms in order to achieve maximum security.

WS-Security is a standard of OASIS (Organization for the Advancement of Structured Information Standards) [9] [10] [11]. It provides security to SOAP based web services by ensuring confidentiality, integrity and authentication. This is a message level security; this means that the SOAP message itself is responsible for its own security by specifying the appropriate security information within the <head> part of the SOAP envelope. WS-Security uses standard security mechanisms in order to achieve end-to-end security. Security tokens such as

SAML and Kerberos are used in order to provide authentication and XML Signature and XML Encryption ensure integrity and confidentiality of the message [12].

WSIT, a project by formerly Sun and now Oracle, and Microsoft, provides an implementation various open web services specifications to support enterprise features [13]. It has been renamed from project Tango and comes as a part of the METRO web services stack and is responsible for providing interoperability with .NET services, security and reliability.

In this paper, several security mechanisms are presented and implemented using a variety of standards in order to provide security to web services and analyse their performance by conducting experiments and measurements.

The overhead performance that is produced by securing a web service after implementing several web services security mechanisms and techniques introduced by the WSIT, is measured, analysed and compared.

2 Problem Formulation

Java programing language has been selected along with its web service stack METRO for evaluation experiments. WSIT services and Glassfish application server for the deployment of web services are integrated with it.

The following list provides a brief description of the security mechanisms that WSIT provides and that experiments will be conducted on. The explanation and description of these mechanisms have been studied from the official WSIT tutorial [32] that Oracle provides and gives a detailed explanation on how to implement the security mechanisms and how they can be used.

2.1 No security

Web service is implemented that will have no security at all in order to have a point of reference for the rest of the mechanisms.

2.2 Endorsing certificate

This security mechanism uses certificates that need to be authorized by a special identity for authorization and identification purposes. The endorsing certificate provides proof for the token that the message is associated with. Furthermore, this mechanism uses symmetric key cryptography for confidentiality and integrity of the message.

2.3 Message authentication over SSL

This mechanism uses SSL (Secure Sockets Layer) to provide confidentiality and integrity protection for the message at the transport level. Furthermore this mechanism uses either a Username Supporting Token or a X.509 Supporting Token for authentication purposes by attaching this token to the message.

2.4 Mutual certificates

This security mechanism requires for both the client and the server to authenticate each other using certificates. When a client asks a web service provider to access a resource that is secured the server provides the client its certificate, the client verifies the certificate and if it is successful it sends its certificate to the server. If the server verifies the certificate then it grants the client the permission to access the secured resource.

2.5 SAML authorization over SSL

This mechanism is based on transport level security for confidentiality. It also uses SAML tokens in order to provide the service provider with authorization information about the client.

2.6 SAML holder of key

This mechanism uses SAML assertions that have been signed from a trusted authority. These SAML assertions hold important authorization information for the requester that mutual certificates ensure integrity and confidentiality. Furthermore the Holder of Key method of this SAML-based mechanism creates a relationship between the SOAP message and the SAML assertions that are added to the SOAP message.

2.7 SAML sender vouches with certificates

With this mechanism the message is protected with a Sender Vouches SAML token for authorization and mutual certificates for confidentiality and integrity.

2.8 Transport security

Transport security mechanism uses SSL in order to protect the SOAP messages that are exchanged. In contrast with all the other security mechanisms that WSIT provides SSL is a point to point security mechanism. This mechanism ensures integrity and confidentiality of the message by establishing a secure connection between two points. The disadvantage of this mechanism that makes

SSL insufficient to provide security to web services by its own is that the message is considered secure only while the message is transmitted. When the message reaches its destination it gets decrypted by the client or the server and the message stops being secured. In order to make sure that the message is secured all the time message level security must be used in conjunction with SSL if needed.

2.9 Username authentication with symmetric key

This mechanism provides integrity and confidentiality using symmetric keys which can be proven important for the performance measurements as they perform faster than public key cryptography. This mechanism uses the username and password pair of the client instead of certificates.

2.10 Username authentication with password derived key

This security mechanism is almost the same with username authentication with symmetric key with the exception that the shared secret key which is used for signing and encryption is generated using a password, a 16 byte random array as salt and an integer value.

3 Problem Solution

In order to measure the overhead that each security mechanism produces a simple echo service has been implemented that takes as a parameter a randomly generated array of bytes and returns it without any processing taking place. It is chosen to use this type of web service in order to minimize the effect of factors that are out of scope of this experiment. Furthermore the byte array is randomly

generated at each web service call in order to avoid any caching mechanism that could exist either to the client or the service provider. Both the service provider and the client have been implemented and deployed locally on the same machine because network delay is out of scope of this experiment and also the latency can be an unpredictable and unstable resulting to imprecise measurements that would lead to wrong assumptions.

Two separate projects have been implemented for each security mechanism, the service provider and the client. Each experiment run corresponds to a different security mechanism. An experiment consists of five call cycles. These cycles run 100 times each and also have an increased byte array size.

Table 1: Experimental Cycles

Cycle No	**Byte Array Size**
Cycle 1	100b
Cycle 2	1Kb
Cycle 3	10Kb
Cycle 4	100Kb
Cycle 5	1Mb

The round trip time (RTT) of each message exchange, have been chosen to measure. This includes from the time that the client creates the request and sends it to the service provider until the service provider responds back with the message and the message is available for the client. In order to measure this time a Java built-in function is used called currentTimeMillis() which returns the current time in milliseconds.

3.1 Testing environment

A Dell machine with a Intel Core i5 2.55 Ghz processor / 4GB of RAM was used. The operating system is Windows 7 64bit. The NetBeans IDE has been used for the development of the web services and the implementation of the experiments using the embedded Metro WSIT wizard for applying the appropriate security for each scenario. Furthermore both the service provider and the client have been implemented as Java servlets and have been deployed on a Glass Fish server.

3.2 Additional Information

In order to implement the security mechanisms that use SAML a callback handler should be implemented. Oracle through its WSIT tutorial provides a sample callback handler that it has been implemented and used for the purpose of our experiments.

For the algorithmic operations the Java Cryptography Extension (JCE) Unlimited Strength 7 was downloaded and installed.

4 Evaluation results

The following table presents the average RTT (Round Trip Time) in milliseconds for each security mechanism and for all the message sizes after conducting the experiments.

The increment percentages of the average RTT of each security mechanism in contrast with the web service mechanism average timings that had no security at all have been gathered.

For message sizes until 1Kb the SAML holder of key mechanism has the

biggest increment percentage of 163% for 100b and 165% for 1Kb. At 10Kb message size the mutual certificates mechanism presents the worst performance with an increment of 104%. SAML holder of key has again the worst performance with 109% at 100Kb message size and SAML sender vouches with certificates mechanism has the biggest increment of 116.47% at 1Mb message size.

In Figure 1 a comparison has been made between some of the security mechanisms and their performance in contrast with the performance of a web service that has no security at all. The security mechanisms included in this chart use message level security to explore the overhead that WS-Security produces which provides message level security. The transport level security (SSL) that some security mechanisms use, are not shown.

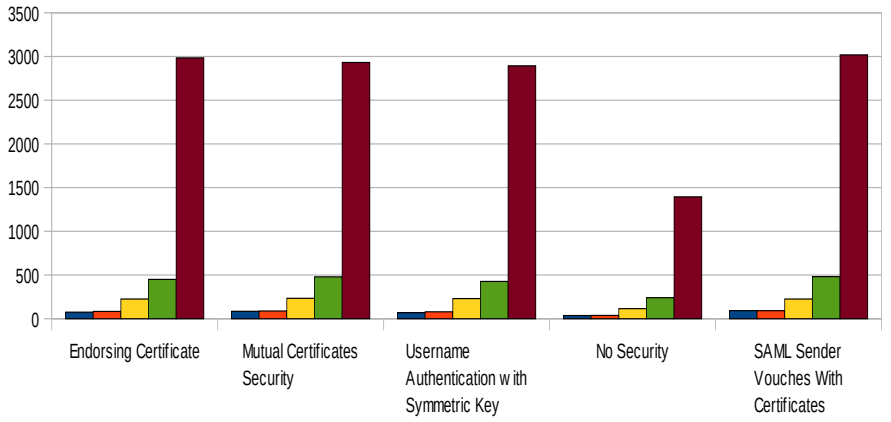

Figure 1: WS-Security overhead

As shown in Figure 1, it can be easily understood that the overhead that can be produced by web services security may exceed the 100% percent increase. SAML sender vouches has an average of 3018.31 ms RTT and is together with the username authentication with password derived key the only mechanisms that exceed the limit of 3000 ms. At 100Kb message size the SAML holder of key

mechanism has the worst performance with 500.67 ms RTT. SAML holder of key mechanism has also the worst performance for message sizes between 100 byte to 1 Kb. At 10 Kb message size the mechanisms mutual certificates security and username authentication with symmetric key present the biggest overhead with 233.72 ms and 230.37 ms respectively.

4.1 SAML-based mechanisms

The Figure 2 shows the three SAML security mechanisms in comparison.

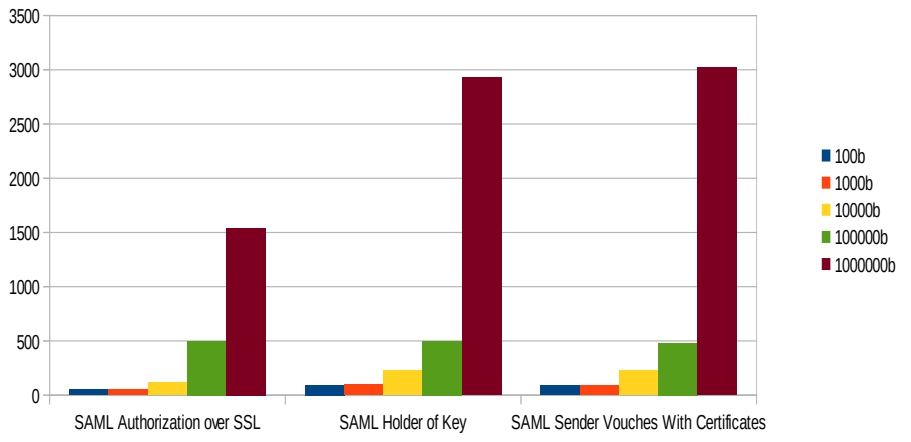

Figure 2: SAML security mechanisms

As shown in Figure 2 the SAML Authorization over SSL mechanism shows a much better performance from the other two mechanisms that use SAML assertions with the exception of the 100Kb message size where the SAML authorization over SSL mechanism has an average RTT of 499.37ms, significantly higher than the average of 481.83ms for SAML sender vouchers but still lower from the average of 500.67ms of SAML holder of key. No justification can be argued for this weird behavior at this specific message size but an assumption

could be possible external problems that occurred from outside the experimental scope causing this anomaly.

SAML holder of key and SAML sender vouches with certificates show about the same performance with the second one having a better performance until 100 Kb message size with a slightly worse performance than SAML holder of key at 10 Kb for 1 ms. At 1Mb message size though the SAML holder of key mechanism performs better than the Sender vouches with certificates with an average of 2931.74 ms while the SAML sender vouches have an average of 3018.31 ms. From the experimental data of the security mechanism that use SAML it is discovered that SAML itself contributes only a minor portion of the overall overhead that is produced.

The most important factor for the performance loss that occurs from the SAML mechanisms is found at the underlying mechanism that they use. These differences are more easily observer at the 1Mb message sizes where it can be seen that the SAML authorization over SSL mechanism has an average of 1542.79 ms when the SAML holder of key has 2931.74 ms and the SAML sender vouches with certificates 3018.31 ms. This big difference between the first SAML mechanism and the other two can be justified by the fact that SAML authorization over SSL as its name suggests uses transport level security (SSL) instead of message level security that the other two mechanisms use. In order to justify this assumption the Figure 3 shows a comparison between the SAML authorization over SSL mechanism and the transport security mechanism which protects the messages by only using SSL.

The two mechanisms that are compared in Figure 3 have very similar performances. With SAML authorization over SSL having an average of 115.32 ms at the 10Kb message size and the transport security mechanism having 114.68 ms at the same message size it can be seen that the two mechanisms perform almost the same. At bigger message sizes the same behavior occurs with the transport security mechanism having an average of 486.46 ms at the 100 Kb

message size and the SAML mechanism with an average of 499.39 ms. At the 1Mb message size the two mechanisms are very close two with the SAML mechanism having an average of 1542.79 ms and the transport security having an average of 1520.91 ms.

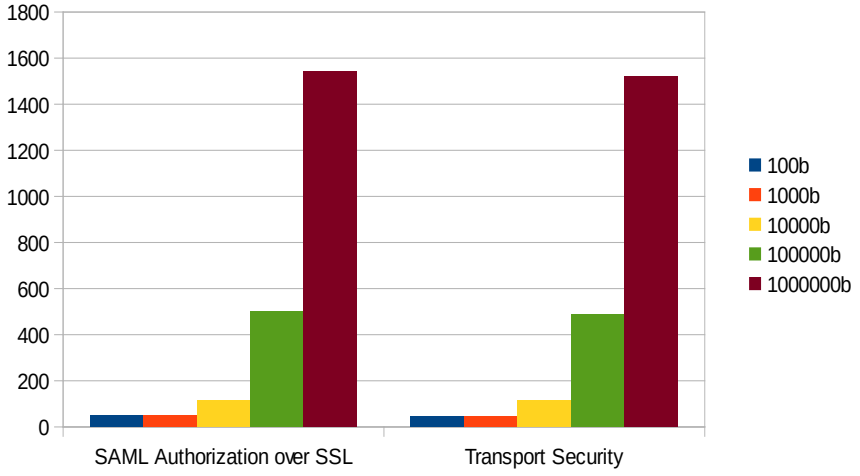

Figure 3: SAML vs. Transport Security

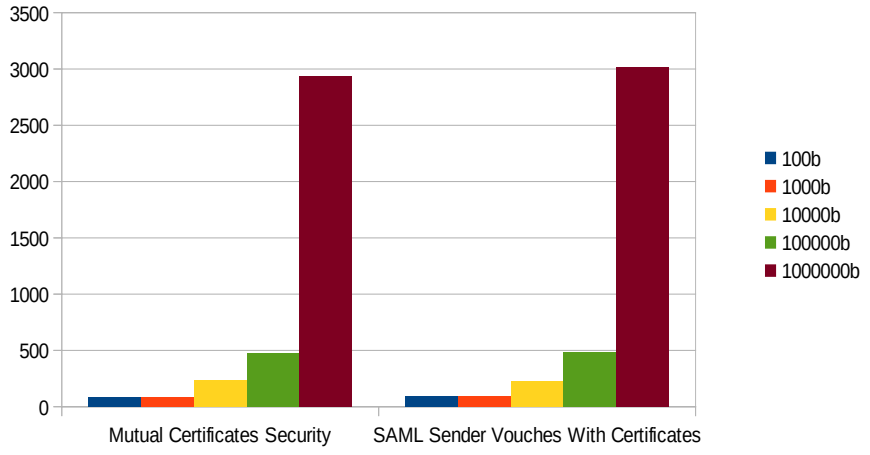

Figure 4: SAML sender vouches – Mutual certificates

The slightly bigger values that the SAML mechanism has can be justified from the SAML assertions operations that increase the overall message size. From this comparison it can be seen that the SAML mechanisms performance is mainly dependent from the underlying security mechanism they use, in this case the SAML mechanism uses transport level security (SSL). Figure 4 confirms this with a comparison with another SAML-based mechanism that uses mutual certificates for integrity and confidentiality. The two mechanisms have almost identical average RTT with the SAML sender vouches mechanism being a bit slower mostly at the 1mb message size with an average RTT of 3018.31 ms.

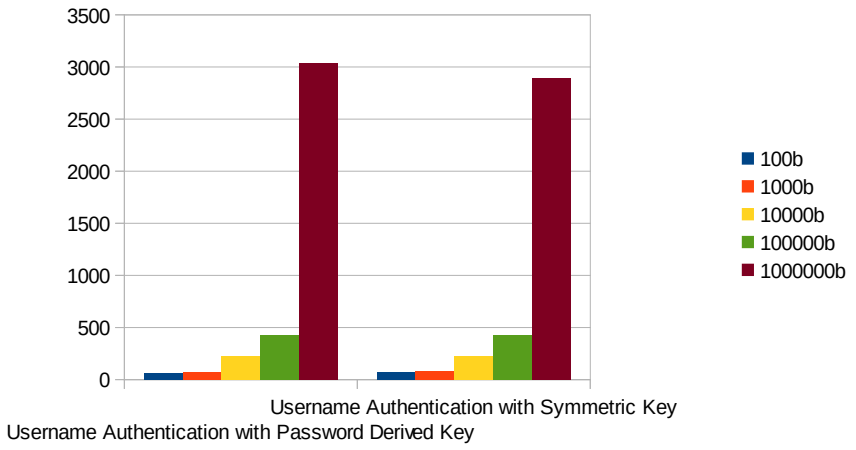

Figure 5: Username token mechanisms

In Figure 5 the two security mechanisms that use username and password pairs for authentication are compared. From the experimental data the password derived key mechanism shows better performance until it reaches the message size of 10Kb. At 100Kb message size the average RTT of the symmetric key mechanism is 426.72 ms, slightly better than the password derived key mechanism which reaches an average of 429.38 ms. The difference between these two mechanisms continues with the same rate until the maximum message size of

1Mb.

It can be justified the better performance of the Username authentication with password derived key mechanism to the fact that is encrypts the username token. It is assumed that the drop of performance for this mechanism has been caused by some possible malfunction of the experiment after a point in time as the architecture of this mechanism does not prove any connection to the message size.

4.2 Transport level security vs. Message level security

In Figure 6 a comparison is made between transport level security and message level security. The transport security mechanism uses SSL in order to provide integrity and confidentiality to the SOAP messages. Message level security is represented by the username authentication with password derived key mechanism which uses username tokens for authentication and symmetric keys for XML encryption for integrity and confidentiality.

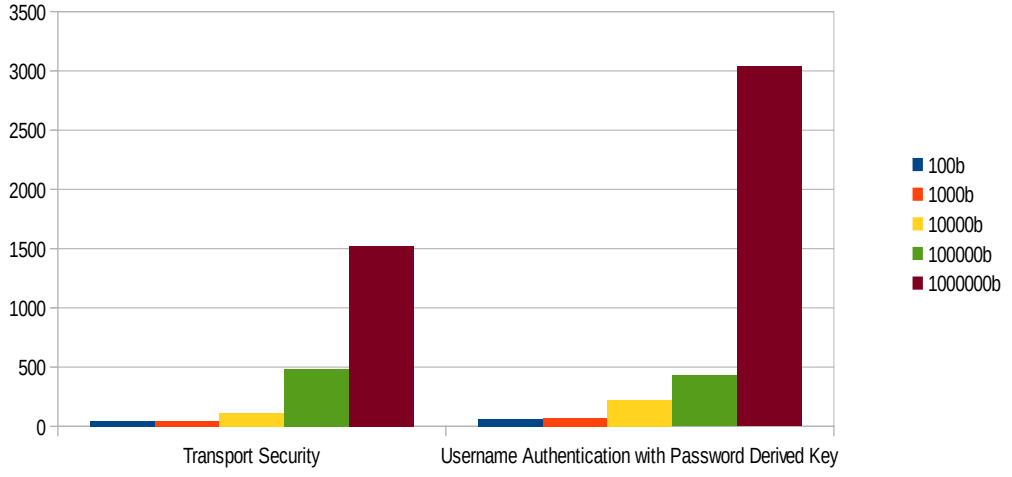

Figure 6: Transport level security vs. message level security

As it can be easily observed the transport level security mechanism performs much better than the message level security mechanism. The difference between the two mechanisms increase even more when the message size reaches 1Mb.

5 Future work

Based on the findings about the performance of web services security it could be suggested for future work a more detailed approach and analysis to specific mechanisms that present bad performance. Through this analysis process and by understanding the work flow, the XML parsing and the algorithmic operations that these security mechanisms use, optimization could be invented. The optimization could focus either at the algorithmic choice and combination of algorithms either at the XML signature and XML encryption it-self or at the authorization part of the message level security. In addition, the same experiments that were conducted here could also be tested to network situations.

6 Conclusion

The purpose of this work was to measure the performance of web services security and specifically the performance of the security mechanisms that WSIT provided by implementing several specifications and technologies. Furthermore the provided charts allowed to compare and group mechanisms together in order to come to conclusions for their behaviour. The main objectives of this work which were to conduct experiments in order to compare the security mechanisms were met. Furthermore some inconsistent behaviours of some specific security mechanisms could not be explained which could be a result of false data retrieved from the experiments.

References

[1] B. Alrouh, G. Ghinea, A Performance Evaluation of Security Mechanisms for Web services, *Fifth International Conference on Information Assurance and Security*, (2009).

[2] N.G. Bardis, N. Doukas and K. Ntaikos, Design and Development of a Secure Military Communication based on AES Prototype Crypto Algorithm and Advanced Key Management Scheme, *WSEAS Trans. on Information Science & Applications*, **5**(10), (Oct., 2008), 1501-1510.

[3] D. Rodrigues, J.C. Estrella and K.R. Banco, Analysis of Security and Performance Aspects in Service-Oriented Architectures, *International Journal of Security and Its Applications*, **5**(1), (January, 2011).

[4] N.G. Bardis, N. Doukas, O.P. Markovskyi and A. Drigas, Two level efficient user authentication scheme, *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, Dubai, (2010), 470 - 474.

[5] K. Moutsakis, *Performance of Web Services Security Mechanisms: Evaluation and Analysis*, MSc. Thesis, University of Hertfordshire, IST College, 2012.

[6] S. Shirasuna, et al., Performance comparison of security mechanisms for grid services, *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, GRID'04, IEEE Computer Society Press, (2004).

[7] M. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko, Comparison of Performance of Web Services, WS-Security, RMI, and RMI-SSL, *Journal of Systems and Software*, **79**(5), (2006).

[8] S.N. Srinama, M. Jarke and W. Prinz, A performance evaluation of mobile web services security.

[9] R.A. van Engelen and W. Zhang, Identifying Opportunities for Web Services Security Performance Optimizations, *IEEE Congress on*, **10**, (2008), 209-210.

[10] R.A. van Engelen and W. Zhang. ”An Overview and Evaluation of Web Services Security Performance Optimizations”, *Proceedings of the 2008 IEEE International Conference on Web Services*, (2008), pages 137-144.

[11] J. Tekli, E. Damiani, R. Chbeir, and G. Gianini. “Similarity-based SOAP Processing Performance and Enhancement”, *IEEE Transactions on Service computing*, (2011).

[12] A. Moralis, V. Pouli, M. Grammatikou, S. Papavassiliou, V. Maglaris. ”Performance Comparison of Web Services Security: Kerberos Token Profile against X.509 Token Profile”, *Proceedings of the Third International Conference on Networking and Services*. (2007).

[13] S. Chen, J. Zic, K. Tang, D. Levy. “Performance Evaluation and Modeling of Web Services Security”, *Proceedings of the IEEE International Conference on Web Services,* (2007), pp. 431-438.