International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Identifying Opportunities for Web Services Security Presentation Optimizations

[1]Dr.T.Swarnalatha, [2] J.Jayalakshmi

[1, 2] Department of Computer Science & Engineering, Narayana engineering college, Nellore Andhra Pradesh (A.P.), (India)

ABSTRACT

Data sharing has never been easier with the advances of cloud computing, and an accurate analysis on the shared data provides an array of benefits to both the society and individuals. Data sharing with a large number of participants must take into account several issues, including efficiency, data integrity and privacy of data owner. Ring signature is a promising candidate to construct an anonymous and authentic data sharing system. It allows a data owner to anonymously authenticate his data which can be put into the cloud for storage or analysis purpose. Yet the costly certificate verification in the traditional public key infrastructure (PKI) Setting becomes a bottleneck for this solution to be scalable. Identity-based (ID-based) ring signature, which eliminates the process of certificate verification, can be used instead. In this paper, we further enhance the security of ID-based ring signature by providing forward security: If a secret key of any user has been compromised, all previous generated signatures that include this user still remain valid. This property is especially important to any large scale data sharing system, as it is impossible to ask all data owners to re-authenticate their data even if a secret key of one single user has been compromised. We provide a concrete and efficient instantiation Of our scheme, prove its security and provide an implementation to show its practicality.

I. INTRODUCTION

Re-presentational state transfer or "REST" is a type of Application or a Service that are used to connected with different types of applications in a disconnected architecture. Data Flow from these server and client will be communicated by the means of some common language like Ex-tensible mark-up or Java script object notation. Due to its simplicity it is used as a wide range of applications. Security of these applications is maintained by passing credentials in Headers of the input Message.

Maintaining the service with two factor authentication is sending extra param in the existing parameters. This will attain the below factors: Two factor with Performance - how components interact affects performance .Two factor with Scalable - able to support large numbers of components, Two factor with Simple - between interacting interfaces ,Two factor with Efficiency - of components to meet changing needs ,Two factor with Transparency - clear communication between components ,Two factor with immediate Changes - of the data-filled code ,Two factor with Reliable - or resistance to fail at system level.

Everything in the REST-ful engineering is about assets. An asset is a question with its own related information. Assets have associations with different assets and an arrangement of techniques or verbs to work between these assets. At that point you can have an accumulation of assets which can cooperate as a gathering with one or more assets or accumulations. REST is straightforward as an idea since it takes after an essential dialect of

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

HTTP 1.1 hypertext exchange that the whole Web sees, to be specific the accompanying, plain as day activity verbs, which are generally composed in capital letters to emerge Action POST - to include information, as to a message board ,Action PUT - to spare an upgrade to the bound together asset identifiers (URI) ,Action PATCH - to roll out an improvement in a demand

In the blink of an eye, with simply these confined operations, REST essentially focuses on joint efforts between data segments and on what parts portions play, instead of focusing on unobtrusive components like lingo and executions. REST transformed into the bases on which HTTP benchmarks and URIs was arranged, which were moreover made by Fielding in parallel. Bringing each of the three things all around and REST easily transformed into the generally speaking and recognized programming building style for the World Wide Web the Term Web administrations portrays an institutionalized method for incorporating Web-based applications utilizing the XML, SOAP, WSDL and UDDI open benchmarks over an Internet convention spine. XML is utilized aggregate the information, SOAP is utilized to exchange the information, WSDL is utilized for portraying the administrations accessible and UDDI is utilized for posting what administrations are accessible. Utilized principally as a method for organizations to speak with each other and with customers, Web administrations permit associations to impart information without close learning of each other's IT frameworks behind the firewall.

Not in the slightest degree like customary client server models, for instance, a Web server Web page structure, Web organizations don't outfit the customer with a GUI. Web benefits rather share business basis, data and techniques through a programmed interface over a framework. The applications interface, not the customers. Originators can then add the Web organization to a GUI, (for instance, a Web page or an executable program) to offer specific value to customers. Web organizations allow different applications from different sources to talk with each other without dull custom coding, and in light of the way that all correspondence is in XML, Web organizations are not altering to any one working structure or programming lingo. For example, Java can talk with Perl, Windows applications can speak with UNIX applications.

The REST-ful administrations and Web API in the present business are utilized for web, versatile and desktop applications. So one must feel secure while presenting applications to end clients. A few applications like fiddlers can follow inflow and surge for the web related stuff, even those applications uncovered the information that flown out from the framework.

II. LITERATURE SURVEY

Literature survey is that the most significant step in package development method. Before developing the tool it's necessary to see the time issue, economy and company strength. Once this stuff square measure happy, 10 next steps square measure to see that package and language may be used for developing the tool. Once the programmers begin building the tool the programmers would like heap of external support. This support may be obtained from senior programmers, from book or from websites. Before building the system the higher than thought taken under consideration for developing the planned system.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

III. PROBLEM DEFINITION

It will coordinate to receive some authorization segments to the server to authenticate using the transformation of data between two different schemas. Such that the token will accept the piece of transformational data to prioritize the authentic level of data aggregation. Such that both the system will authenticate and transformation of data will point the segmented formatted data.

Shared token will authenticate from server to client. Unique key value also authenticates the primary level of authentication to survive. Transformation of data with encrypted and decrypted formats are used to assign the original assignments. Latent discussion will lead to the header based authentication. Body based parameter authentication.

Communicating with the REST-ful service will follow all the OSI layer communications that are used to transfer the content. While communicating with the server through HTTP or TCP IP protocol network may get attacked by the hacker to steal the content or the credentials. Even the Credentials are passed in such a way. Your WCF authentication options depend on the transfer security mode being used. For this reason, your authentication choices are partly determined by your transfer security mode. WCF offers the following transfer security modes. When using message security, the user credentials and claims are encapsulated in every message by using the WS-Security specification to secure messages. This option gives the most flexibility from an authentication perspective. You can use anytime of authentication credentials you want, largely independent of transport, as long as both client and service agree. When using transport security, the user credentials and claims are passed by using the transport layer. In other words, user credentials are transport dependent, which allows fewer authentication options compared to message security mixed security gives you the best of both worlds: transport security ensures the integrity and confidentiality of the messages, while the user credentials and claims are encapsulated in every message as in message security. This allows you to use a variety of user credentials that are not possible with strict transport security mechanisms, and to leverage transport security's performance. When using this option, the user credentials and claims are transferred at both the transport layer and message level. Similarly, message protection is provided at both the transport layer and message level. Note that this is not a common scenario, and only bindings that support the Microsoft Message Queuing (MSMQ) protocol support this security mode Security is applied on a point-to-point basis, with no provision for multiple hops or routing through intermediate application nodes. • It supports a limited set of credentials and claims compared to message security. • It is transport-dependent upon the underlying platform, transport mechanism, and security service provider, such as NTLM or Kerberos. In message Security This option may reduce performance compared to transport security because each individual message is encrypted and signed. It does not support interoperability with older ASMX clients since it requires both the client and service to support WS-Security specifications.

Token or credentials will be shared with company developers. That can misuse while developer changes the company or he tries with personal belief. Latent change of communication which leads to coordinate the static paradigm. It's difficult to change passwords frequently while developer changes the company. Password change from server takes some process.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 07, July 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

IV. PROBLEM ANALYSIS

After consistent study on the cryptography and information security I have found that it has been excessively iterative, making it impossible to sort out the information in each of the OSI layers to maintain the security customs to finish up the roll based security highlights for every framework. Before speaking with the arrange layer, each of the information change to be corresponded in every module to cover the first information with conceal information. And after that scramble into the strong un-indistinguishable information.

Expensive endorsement checks in the conventional open key base setting turns into a bottleneck for this answer for be versatile. Personality based (ID-based) ring mark, which disposes of the procedure of testament check, can be utilized. In this paper, we assist improve the security of ID-based ring mark by giving forward security: If a mystery key of any client has been traded off, all past created marks that incorporate this client still stay substantial. This property is particularly vital to any expansive scale information sharing framework, as it is difficult to ask all information proprietors to re-verify their information regardless of the possibility that a mystery key of one single client has been traded off. We give a solid and proficient instantiation of our plan, demonstrate its security and give an execution to demonstrate its common sense.

Dynamic generation of key will resultant to secrecy. No constant key will be passed to all the time such that it's difficult to hack the service. Each key has its own mechanism of cryptic script that results the latent scalable paradigm. Even after the developer has main token it will allocate the decent stable proposal to authenticate. Shared library of script will be same across client and server machines.

V. FLOW DIAGRAM

Flow diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

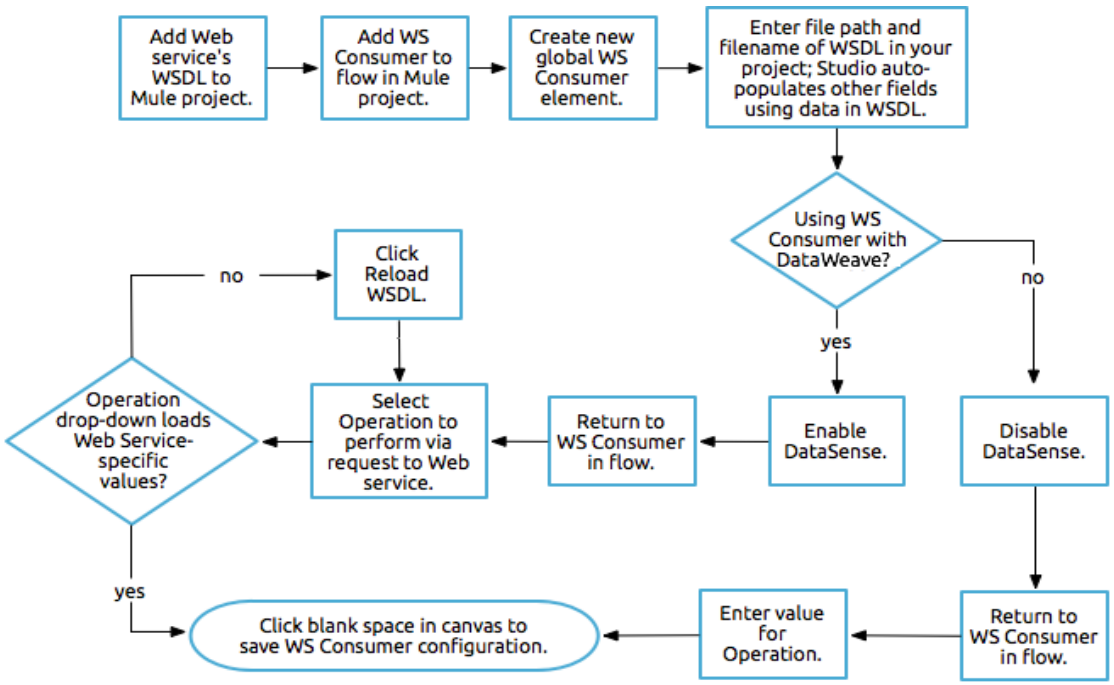

Fig: flow diagram

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. A use case diagram is a type of behavioral diagram created from a Use-case analysis. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts The upper part holds the name of the class,The middle part contains the attributes of the class, The bottom part gives the methods or operations the class can take or undertake. In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

VI. IMPLEMENTATION

6.1Use and Scenario

- **Situation 1**

On the off chance that we have a XML archive and we require the estimation of a property <attribute> for XML hub <node> into a variable <Attr_Value>, to show on the presentation layer then we can utilize the capacity as underneath:

Attr_Value=XMLUtility.GetXmlAttribute(<node>,< attribute>)

- **Situation 2**

In the event that we have to control any XML report like including a hub or erasing a hub then we can utilize the beneath capacities:

- **Including a sub hub**

newXML = XMLUtility.AddXMLSubNode(xml, xPath, subNode, subNodeValue);

Erasing a sub hub:

newXML = XMLUtility.DeleteXMLNode(xml, xPath);

where xml is the XML string to stack, xPath is the XPath to hunt hub into the XML string, subNode is the name of the new subnode, subNodeValue is the Value of the new subnode, namespaceURI is the Namespace URI of new subnode. The xml string with new subnode will be returned into "newXML" string variable.

- **Situation 3**

We have a question and we need to store this protest into our record framework (or other perseverance medium) we can utilize the underneath highlight to serialize that protest into XML and store into document framework.

XMLUtility.XMLFileSerialization (objVal, strXMLfile);

Where objVal is the nonspecific protest serialize and strXMLfile is the name of the xml document to be spared with physical way.

- **Situation 4**

On the off chance that we have put away a question into record framework and we need to reproduce that protest then we can utilize the element of Deserialization as underneath.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

XMLUtility.XMLFileSerialization (objVal, strXMLfile);

Where objVal is the bland question serialize and strXMLfile is the name of the xml document to be spared with physical way.

6.2 Object Serializer (Object Reader Writer Utility)

Object Reader Writer Utility is a reusable part fabricated utilizing C#.NET 4.0. It can read any sort of record (txt, CSV, altered length, exceed expectations and so on where document's information can be changed over into unthinkable shape) and returns record information as a protest gathering. The other way around it can read any kind of question gathering and compose its information into records (txt, CSV, settled length, exceed expectations and so on).

Benefits:

Object Reader Writer Utility is a reusable part and can be used inside any application with no change or with negligible coding overhead.

It decreases improvement and testing expense and upgrades efficiency.

Use and Scenario:

Situation 1

Change over document information into a question gathering by passing diagram and record data (as record way, File Type and so on) as parameter and utility will return protest accumulation as beneath:

List<Employee>objList== Object Reader Writer.Read Object<Employee>(info);

Situation 2

Change over a protest gathering into record information by passing mapping, document data and question accumulation as parameter and the utility will give back a record of coveted sort :

6.3. Extension techniques

This segment of nFactory system gives several Extension Methods which are absent from the .net structure and are exceptionally helpful for an engineer to code for any rehashed usefulness. This incorporates augmentations like Array Extensions, Datetime Extension, Generic Object Extensions et cetera. There are a few expansion techniques accessible; just a couple are portrayed in the use and situation area.

Benefits:

A scope of prepared to-utilize strategies by including a basic reference.

Reduces advancement exertion

Use and Scenario:

Situation 1:

There's a need to consolidate the consequences of two exhibits into one. Augmentation techniques have a capacity Combine which can do this in straightforward one line of code as beneath:

Array Type[ ] Combine<Array Type> (this Array Type] Array1, Array Type[ ] Array2)

Situation 2

On the off chance that there's a need to know what number of more days are left in the present month, this can be accomplished as beneath:

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Int Days Left In Month (this DateTime Date)

Situation 3

While working with the accumulations, there's a need to add a thing to the gathering, so the strategy AddIfUniquecan be utilized. It first checks if the thing isn't now there in the accumulation. It returns genuine if included effectively. Else, it returns false

6.4 Job Scheduler

The Job Scheduler is a full-highlighted work planning framework that can be utilized from little applications to vast scale venture frameworks.

- **Benefits**

It is light-weight, and requires almost no setup/arrangement - it can really be utilized 'out-of-the-crate' if the necessities are generally essential.

The Scheduler is blame tolerant, and can continue ('recall') the planned occupations between framework restarts. It decreases a decent measure of advancement time and exertion.

Use and Scenario:

" SimpleHelloScheduler" to keep running on a set date and time:

- **HTML to PDF**

HTML to PDF is a library used to make PDF records from a XML or a XHTML/CSS document. It gives the greater part of the primitive capacities important to make a PDF report.

- **Benefits**

Provides an ordinarily utilized element as a part of an application and which is not accessible in .net Framework. Does not depend on any outsider segments. Reduces advancement exertion.

- **Use and Scenario**

Making of PDF documents is a general prerequisite in numerous applications. This segment will diminish the improvement exertion to make a PDF record.

6.5 Identity

Identity management for Web services is similar to identity management for any IT system in that the subject (whether a person, machine, program, or abstrac-tion such as a process flow) is given a unique or unambiguous name within the security domain whose validity can be checked. The identity of a Web service requester is sometimes critical for a provider to establish trust because whether or not the requester is allowed to access the provider's service (or any other service, data resource, or device managed by the provider's service) depends upon the identity of the requester.

Identity management is complex for Web services, just like it is for the Web, because Web services can span departments and enterprises. Typically, identity management is performed locally, departmentally, or within an enterprise by ensuring that each employee's user name is unique on the network. Employees are responsible for keeping their passwords private because passwords are used to authenticate the user's identity and to determine the applications, directories, and data the user is allowed to access.

Identity management may need to be performed within a broader scope, such as the Microsoft Active Directory

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

or a corporate-wide LDAP solution. When an identity has to be uniquely managed across the Internet and across enterprises, the level of administration difficulty increases, as does the need for trust.

1. Various initiatives, such as those sponsored by the Liberty Alliance, are focused on establishing mechanisms for identity management for the Internet

6.6 Authentication

Authentication is the process through which an authority verifies a subject's identity, based on some set of proof such as a password or personal identification number (PIN). The authentication process creates a principal, which is an object that represents the authenticated subject, such as a credential or token that the subject can use later. On the Web, the subject is typically a user, but for Web services, it can be a machine, program, or other abstract entity represented by the Web service requester. Web services typically use some form of the user name/password mechanism for basic authentication, but stronger forms such as signatures also may be used.

Authentication can be described as the process of confirming that you (or your proxy service requester) are who you say you are. On the Web, this is most often seen as a popup user name/password box, which is called forms-based authentication, which uses a cookie returned on subsequent invocations.2 Only you know the correct user name and password, so you are authenticating your-self as someone who is allowed to access the Web site. The Web site will have to set up and manage a directory of authorized user name/password combinations so that it can verify the information you submit.

Web services requesters can include authentication information using user name/password information in SOAP headers that the service provider can check against its directory of authorized user name/password combinations. The user ID and password can also be sent via HTTP (no SOAP header required). The provider typically carries out a further refinement of this model to support specific checks for authorization to access specific services or specific data resources. Sometimes requesters are assigned certain roles that can be used as indexes into authorization information—meaning authorization is sometimes carried out according to specific roles such as administrator, clerk, or manager, but again, this is typically managed by the provider and may not appear in the SOAP header (and certainly not in the WS-Security header if it appears at all).

Authentication is needed in Web services to verify the identities of the service provider and service requestor. In some cases, mutual authentication may be needed (that is, the provider must authenticate the requester and vice versa).

6.7 Digital signature

A digital signature signs a message digest using a public/private key pair. A hash algorithm creates the message digest, and the encryption algorithm signs the digest (with the private key). The receiver decrypts the signature using the public key, recomputed the message digest, and compares the two. If the message has been altered, the results won't match, and the provider knows the message has been tampered with. As in other encryptions, symmetric or asymmetric key algorithms can be used to compute the signature, although for signing the user of asymmetric keys is more typical.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

6.8 Security Architecture

It's important to view the Web services security challenges and threats within their overall architectural context and determine solutions based not simply on a given technology but rather on looking at the overall solution context. That is, you can't just say "use SSL" without understanding the threat you're trying to defend against and without understanding the overall security context into which you'd like to deploy SSL. SSL may be sufficient, but it may not. Multiple security technologies often must be used in con-junction to provide a comprehensive solution to the big security concerns, and it is therefore important to understand how the technologies work together. The following sections detail some of the specific challenges and threats that the overall Web services security environment must address

6.9 Message Interception

The potential for SOAP message interception and decoding gives rise to a cate-gory of security threats that must be guarded against when deploying Web services, including message replay, alternation and spoofing. Unless specifically encrypted, Web services messages are transmitted in plain text, which can easily be intercepted and read. An intercepted message can be modified, potentially affecting all or part of the message body or headers. Additional bogus information could be inserted into a message header or body Parts. Any message attachment could also be modified or replaced. Altering the message or the attachment could cause bogus information to be sent to and received from a Web service, possibly including a virus. Reading an intercepted message can also give anyone access to confidential information within a message or message attachment, such credit card information, social security numbers, bank account numbers, and so on.

Protecting against message interception includes the use of encryption and digital signatures to preserve confidentiality and integrity.

Because SOAP messages can be routed through intermediaries, and because intermediaries are able to inspect the messages to add or process headers, it's possible for a SOAP intermediary to be compromised. Messages between the requestor and the ultimate receiver could therefore be intercepted while the original parties still believe they are communicating with each other.

Mutual authentication techniques can protect against this type of threat, but signed keys or derived keys provide even better protection.

Spoofing is a complex challenge in which an attacker assumes the identity of one or more trusted (i.e., authenticated) parties in a communication in order to bypass the security system. The target of the attack believes it is carrying on a conversation with a trusted entity. Usually, spoofing is a technique to launch other forms of attack such as forged messages that request confidential information or place fraudulent orders.

It's possible for spoofed Web service messages to include SQL or script tampering to attack through JSP or ASP script execution. Mutual authentication techniques can protect against this type of threat.

A replay attack is one in which someone intercepts a message and then replays it back to the receiver. Replays could also be used to gather confidential information or to invoke fraudulent transactions.

Strong authentication techniques together with message time stamp and sequence numbering can protect against this type of threat.

When an unauthorized intermediary or other attacker intercepts a SOAP message, the attacker can resend it

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

repeatedly in order to overload the Web services execution environment and effectively deny service to legitimate services that are trying to get through. An attacker can also blast a ton of messages to a Web service after the attacker gets its address. Even if the messages are rejected, the site can get overloaded with error processing.

In general, if someone wants to launch this type of attack, there's no real defence. However, firewall appliances are growing in popularity because they can help mitigate denial-of-service attacks.

The first level that needs to be secured is the communications transport. In the case of Web services, this is almost always TCP/IP, and this is certainly the case when using HTTP.

Firewalls map a publicly known IP address to another IP address on the internal network, thereby establishing a managed tunnel and preventing access by pro-grams at unauthorized addresses. Web services can work through existing fire-wall configurations, but this often means increased protection has to be added to firewalls to monitor incoming SOAP traffic and log any problems. Anotherpopular solution involves the use of XML firewalls and gateways that are capable of recognizing Web services formats and performing initial security checks, possibly deployed as intermediaries or within a "demilitarized zone" (i.e., be-tween firewalls).

VII. CONCLUSION

The Shared library of both client and Server will produce the secure authentication for Restful services in POST method in order to transform some data and reverent information to secure the data optimistic level of integration This will produce the high level interaction in the duplicate data process in logical stabilizing. In tenure the Authentic with token and unique identity will process hierarchy to abolish the standard level of integrated mechanism to coordinate in multiple ordinal data security. Also increase the performance due to avoiding of un authorized request. Securing the service with this shared library that contains the unique logic for each established library to provide the details in standard data library in accuracy and standard legalization.

IX. REFERENCES

[1] N. Abu-Ghazaleh and M. Lewis. Differential deserialization for optimized SOAP performance. In proceedings of the ACM/IEEE conference on Supercomputing, pages 21– 31, Los Alamitos, CA, 2005. IEEE Computer Society.

[2] N. Abu-Ghazaleh, M. Lewis, and M. Govindaraju. Differen-tial serialization for optimized SOAP performance. In pro-ceedings of the IEEE International Symposium on High Per-formance Distributed Computing, pages 55–64, Los Alami-tos, CA, 2004. IEEE Computer Society.

[3] S. Chen, J. Zic, K. Tang, and D. Levy. Performance eval-uation and modeling of Web services security. In Proceed-ings of the IEEE International Conference on Web Services (ICWS'07), pages 431–438, 2007.

[4] M. R. Head, M. Govindaraju, R. van Engelen, and W. Zhang. Benchmarking JSON processors for applications in Grid Web services. In proceedings of ACM/IEEE Supercomput-ing Conference, Los Alamitos, CA, November 2006. IEEE Computer Society.

[5] M. B. 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):689–700, 2006.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[6] H. Liu, S. Pallickara, and G. Fox. Performance of Web ser-vices security. In 13th Annual Mardi Gras Conference, Ba-ton Rouge, Lousiana, USA, Feburay 2005.

[7] W. Lu, K. Chiu, A. Slominski, and D. Gannon. A stream-ing validation model for SOAP digital signature. In In 14th IEEE International Symposium on High Performance Dis-tributed Computing (HPDC-14), 2005.

[8] S. Makino, K. Tamura, T. Imamura, and Y. Nakamura. Im-plementation and performance of WS-Security. Int. J. Web Service Res., 1(1):58–72, 2004.

[9] Moralis, V. Pouli, M. Grammatikou, S. Papavassiliou, and V. Maglaris. Performance comparison of Web services security: Kerberos token profile against X.509 token profile. In ICNS '07: Proceedings of the Third International Con-ference on Networking and Services, page 28, Washington, DC, USA, 2007. IEEE Computer Society.

[10] Network Working Group. The Transport Layer Protocol Version 1.1 (RFC4346). Available at http://www.faqs.org/rfcs/rfc4346.html.

[11] Oasis Consortium. WS-Security specification, 2004. Avail-able from www.oasis-open.org.

[12] Oasis Consortium. Web services secure conversation lan-guage (WS-SecureConversation), 2005. Available from www.oasis-open.org.

[13] Oasis Consortium. Web Services Trust Language (WS-Trust), 2005. Available from www.oasis-open.org.

[14] S. Shirasuna, A. Slominski, L. Fang, and D. Gannon. Per-formance comparison of security mechanisms for Grid ser-vices. In GRID '04: Proceedings of the Fifth IEEE/ACM In-ternational Workshop on Grid Computing (GRID'04), pages 360–364, Washington, DC, USA, 2004. IEEE Computer So-ciety.

[15] T. Suzumura, T. Takase, and M. Tatsubori. Optimizing Web services performance by differential deserialization. In ICWS '05: Proceedings of the IEEE International Confer-ence on Web Services (ICWS'05), pages 185–192, Washing-ton, DC, USA, 2005. IEEE Computer Society.

[16] R. van Engelen. The gSOAP toolkit for C and C++ Web ser-vices, 2001. Available from http://gsoap2.sourceforge.net.

[17] R. van Engelen. A framework for service-oriented com-puting with reusable C and C++ Web service components. accepted for publication in ACM Transactions on Internet Technologies, 2015.

[18] R. van Engelen and K. Gallivan. The gSOAP toolkit for Web services and peer-to-peer computing networks. In proceed-ings of the IEEE International Symposium on Cluster Com-puting and the Grid, pages 128–135, Los Alamitos, CA, May 2002. IEEE Computer Society.

[19] R. A. van Engelen. Constructing finite state automata for high performance JSON Web services. In proceedings of the International Symposium on Web Services (ISWS), Las Vegas, NV, 2004. CSREA Press.

[20] R. A. van Engelen. A framework for service-oriented com-puting with C and C++ Web service components. To appear in ACM Transactions on Internet Technologies, 2015.

[21] J. Viega, M. Messier, and P. Chandra. Network Security with OpenSSL. O'Reilly, 2002.

[22] W3 Consortium. JSON Encryption specification. Available from www.w3.org.

[23] W3 Consortium. JSON Signature specification. http://www.w3.org/TR/JSONdsig-core/.

[24] White Mesa. White Mesa Interop Lab. Available from www.whitemesa.com/interop.html.