Secure and Authenticated Access to LLN Resources Through Policy Constraints

Konstantinos Rantos[1], Konstantinos Fysarakis[2](✉), Othonas Soultatos[3], and Ioannis Askoxylakis[4]

[1] Department of Computer and Informatics Engineering,
Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece
krantos@teiemt.gr

[2] Department of Electronic & Computer Engineering,
Technical University of Crete, Chania, Crete, Greece
kfysarakis@isc.tuc.gr

[3] Department of Computer Science, University of Crete, Heraklion, Crete, Greece
sultatos@csd.uoc.gr

[4] Institute of Computer Science, Foundation for Research and Technology - Hellas
(FORTH), Heraklion, Greece
asko@ics.forth.gr

Abstract. Ubiquitous devices comprising several resource-constrained sensors and actuators while having the long desired Internet connectivity, are becoming part of many solutions that seek to enhance user's environment smartness and quality of living. Their intrinsic resource limitations however constitute critical requirements, such as security, a great challenge. When these nodes are associated with applications that might have an impact in user's privacy or even become life threatening, the security issues are of primary concern. Access to these resources should be appropriately controlled to ensure that such wearable nodes are adequately protected. On the other hand, it is very important to not restrict access to only a very closed group of entities. This work presents a service oriented architecture that utilizes policy-based, unified, cross-platform and flexible access control to allow authenticated entities consume the services provided by wearable nodes while protecting their valuable resources.

Keywords: Body sensor networks · Policy-based access control · XACML · SAML · DPWS · Web services · Security

1 Introduction

In recent years, we have experienced a lot of innovation in the Internet of Things (IoT) space. Collections of nodes typically bearing sensors and actuators are becoming part of a networking infrastructure and gain connectivity to the Internet. The corresponding technologies are becoming mature enough to start looking into more advanced and comprehensive solutions that can enable these nodes to integrate smoothly with existing infrastructures while, however, expanding existing attack surfaces.

R.N. Akram and S. Jajodia (Eds.): WISTP 2015, LNCS 9311, pp. 271–280, 2015.
DOI: 10.1007/978-3-319-24018-3_18

There are many application areas where these nodes flourish with even more being introduced to take advantage of the services that they can offer. They can be deployed as standalone nodes serving a single purpose, or as part of an infrastructure that consists of nodes with similar characteristics comprising a so called low power and lossy network (LLN). The current trend for all these nodes characterised by their limited resources in terms of computing power, memory, storage space and energy, is to adopt existing networking technologies and be reachable over the Internet, abandoning proprietary closed solutions.

Sensor nodes and Service Oriented Architectures (SOAs) have become convergent technologies with several standards emerging from these efforts. SOAs evolved from the need to have interoperable, cross-platform, cross-domain and network-agnostic access to devices and their services. At the same time, studies [1] and published reports[1] reveal that current deployments have not adequately considered the threats that these nodes face when connected to the Internet, hence the lack of the security measures. Such negligence is bound to inhibit any efforts made towards using these pervasive devices to handle our personal sensitive data. The expanded attack surface that results from the integration of LLNs with the Internet, needs new or adapted mechanisms to mitigate these new threats.

This paper defines an architecture that controls access to services provided by resource-limited nodes. Among the main concerns of the proposed architecture are the nodes' protection from unauthorised and unjustifiable use of their resources and the need to be able to control access through a well-established set of policy rules that can change and adapt to new environmental parameters. The work builds upon the eXtensible Access Control Markup Language (XACML) reference model for policy based access control infrastructures and proposes certain modifications to provide flexibility in terms of the authentication mechanism being used and satisfy requirements stemming from the limited resources of nodes.

2 Background and Related Work

Standardisation and research efforts in the area of Service Oriented Architectures have been taking place for more than a decade. Several schemes have been proposed and standardised regarding service discovery, registration, access and protection, and the corresponding communication protocols that enable the interoperable exchange of messages among remote participating entities.

In terms of the way that access to web services is controlled, the eXtensible Access Control Markup Language (XACML) [2], provides an access control language and a model for processing requests to resources while the Security Assertion Markup Language (SAML) focuses on the way the requester is authenticated and assertions are being transferred among participating entities. WS-Trust is another web services oriented that defines how security tokens are being issued,

[1] http://fortifyprotect.com/HP_IoT_Research_Study.pdf

renewed and validated (WS-Trust). This paper focuses more on the area of securing access to resources through policy-based access control, hence it is related and utilises these security related standards mentioned above, while proposing certain modifications mentioned below to fit best to the restricted environment of LLNs.

Many access control schemes have been proposed for wireless sensor networks, yet most of them focus on authentication and authorization schemes and on enhancing basic access control models to address privacy matters. Such schemes can be found in [3–6]. Little work has been carried out on policy-based access control (PBAC). The EU-project Internet-of-Things Architecture (IoT-A) worked on the adoption of XACML in the Internet of Things [7] and proposed a generic model whose functional modules are mapped to a set of well-defined components that comprise the IoT-A. The authors use a logistics scenario for demonstration purposes.

In [8] the authors also utilize XACML but focus on the privacy of e-Health data within the mobile environment. In contrast to the work presented here, a complete framework is not included and, moreover, the authors choose computationally intensive security mechanisms such as XML encryption digital signatures. In [9], the authors propose a lightweight policy system for body sensors but they do so by presenting a custom API and policy definitions, thus sacrificing interoperability with existing standards and infrastructures.

3 Requirements

IP based networking in LLNs changes the way that participating nodes can be accessed and their respective services can be consumed. For instance, there is no need for a dedicated application server that will intervene between a node and a remote party that wants to access the node's resources [10]. However, one of the problems that these nodes face in such a deployment, is that they have limited resources which do not suffice for the deployment of strong protection mechanisms. Without those mechanisms however, nodes are exposed to direct access from the Internet without having the capacity to handle unlimited requests. Therefore, several issues arise regarding the protection of nodes resources, that have to be addressed. The main aim is to protect the limited resources of a node that implements a service oriented architecture, to provide access to data and mechanisms that the node has under control. In this paper we are looking at these issues aiming for a smooth integration of web-services technology, adopted by serving nodes, with the Web.

Within this context, the proposed architecture is designed to satisfy the following requirements:

- Provide services using of Service Oriented Architecture technologies;
- Provide fine-grained access control to nodes' resources;
- Authenticate remote entities wishing to access protected nodes resources;
- Control access to nodes' resources through well-defined policies;

- Protect sensitive nodes from unauthorised access and unnecessary consumption of valuable resources including network and energy;
- Comply with existing standards to satisfy interoperability among the participating entities, such as between the identity provider chosen by the requester and the service orchestrator, regarding the exchange of authentication messages, assertions or user metadata and attributes.

In the following section we describe the proposed architecture that satisfies the above.

4 Proposed Architecture

The architecture proposed in this paper is an enhanced policy based access control scheme that seeks to provide flexibility regarding the chosen authentication mechanism while satisfying the aforementioned requirements, typically imposed by nodes' resource limitations. For this purpose, certain modifications to the OASIS standardised policy-based access control scheme are proposed to accommodate these needs.

The scheme utilizes and seeks compliance with the following technologies:

- XACML: an XML-based OASIS standard that defines a policy and an access control decision request/response language. An XACML-based architecture typically consists of the following main components:
  - *Policy Enforcement Point (PEP):* Performs access control, by making decision requests and enforcing authorization decisions [2,11].
  - *Policy Decision Point (PDP):* Evaluates requests against applicable policies and renders an authorization decision [2].
  - *Policy Administration Point (PAP):* Creates and manages policies or policy sets [2].
  - *Policy Information Point (PIP):* Acts as a source of attribute values [2].
- SAML 2.0 specification to protect, transport, and request XACML schema instances and other information needed by an XACML implementation [12].

In the XACML data-flow model the PEP, via the context handler, is considered as the device that orchestrates the exchange of messages among the requester, the PDP, the Attribute Authority and the Attribute Repository. According to the XACML specifications the PEP is considered as "part of a remote-access gateway, part of a Web server or part of an email user-agent, etc". Therefore all initial requests, valid or not, are sent to the PEP which will act as a routing device between the requester and the back-end key entities that examine the requests and make decision based on policy rules and other parameters, such as the requester's and/or resource's attributes.

While this model is appropriate for typical application gateways, it cannot be considered as such for resource-constrained nodes that only have the capacity to accept requests from a limited number of clients. Beyond this threshold, valuable node resource consumption is not acceptable as it leads to battery drainage and service unavailability. In this context, resource-constrained devices have to participate in the decision making process only if absolutely necessary and only to authorized entities to save valuable resources. As such, they cannot assume the role of a PEP as this is defined in the XACML standard [2].

Moreover, the flow model currently defined by XACML, considers that the PIP has all the required attributes for the requester, and that the PDP gets all the information from the PIP, which might be queried twice for the required attributes, once from the PEP and once from the PDP. Use of specific PIP implies that services will only be provided to entities subscribed to the specific scheme, thus narrowing down flexibility. This is in contrast to a more flexible approach where services are offered to a broader group of users, subject to policy restrictions.

The proposed architecture is depicted in Figure 1. In this proposal we assume that nodes bearing sensor and actuators, expose their functionality as web services. This can either be done through the device that the node is attached to, e.g. a mobile device, or directly by the node, assuming that it is powerful enough to accommodate such functionality. All these nodes are part of a dispersed environment where there is not necessarily a single gateway or web server to assume the role of PEP as this is defined in the XACML standard. Besides that, the service owner might want to register these services with multiple servers. As a result, the PEP functionality cannot be assigned to a gateway but it should be on the device that exposes this functionality, i.e. the mobile device or the micro/power node. For a given PEP, one of these web servers is assumed to play the role of the orchestrator as described below.

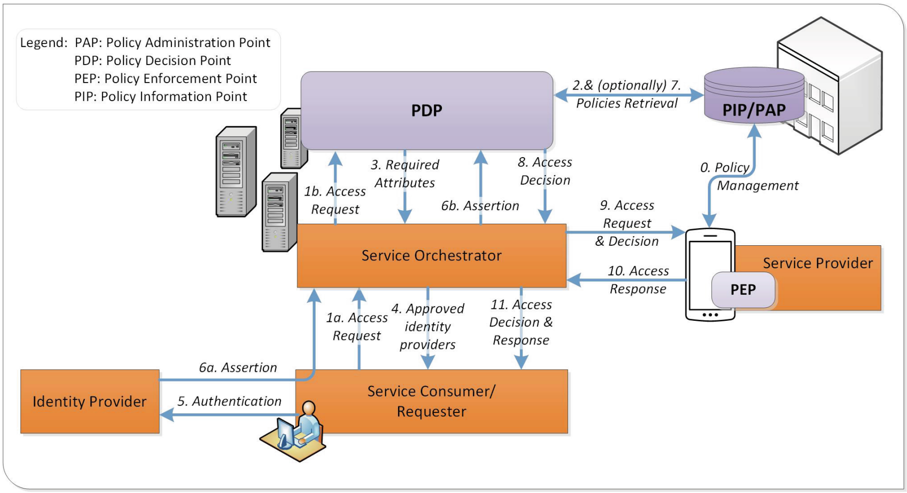
Legend: PAP: Policy Administration Point
PDP: Policy Decision Point
PEP: Policy Enforcement Point
PIP: Policy Information Point
PDP
2.& (optionally) 7. Policies Retrieval
PIP/PAP
0. Policy Management
1b. Access Request
3. Required Attributes
6b. Assertion
8. Access Decision
9. Access Request & Decision
10. Access Response
Service Orchestrator
Service Provider
PEP
1a. Access Request
4. Approved identity providers
11. Access Decision & Response
6a. Assertion
5. Authentication
Identity Provider
Service Consumer/ Requester

Fig. 1. Authenticated Access Control for LLNs

The core component of the proposed scheme is the Service Orchestrator (SO) which acts as a proxy for certain operations, such as relaying queries and messages exchanged among participating entities, yet not for handling the information the PEP exchanges with the requester.

Initially, the node, which assumes the role of a PEP, registers its services, defines the connection point to be the SO and sets the policy rules for its resources. This is accomplished once during the set-up phase. Following that, the data flow of the proposed architecture includes the following steps:

- A requester, who wants to access the service, formulates an appropriate request based on the advertised service rules, and sends it to the SO (step 1a). Note that this is in contrast to the XACML specifications which opted for sending the request directly to the PEP, introducing significant overhead that a limited-resources device cannot handle.
- The SO forwards the request to the PDP (step 1b) which, based on the requested target, fetches all applicable policies from the PAP (step 2) and informs the SO about the needed user attributes (step 3). As a result, the SO presents a list of approved Identity Providers (IdP) for the requester to authenticate (step 4).
- The requester chooses the appropriate IdP and the SO issues a (signed) authentication request (`<AuthnRequest>`) together with an attribute query (`<AttributeQuery>`) [12] to the chosen IdP. Upon successful authentication (step 5) the requester consents for the disclosure of certain attributes that the SO requires. Note that the IdP might be an entity that operates within the same environment as the SO. The authentication method used by the IdP is outside the scope of this paper.
- The IdP formulates a proper assertion for the necessary attributes and sends it to the SO via the Requester (step 6a). As a result, the SO forwards the received assertion to the PDP (Step 6b) [13].
- The forwarded assertion allows the PDP to establish a security context by combining the supplied attributes with the applicable policy rules which the PDP obtained from the PAP (step 2). Note that additional policy rules, might be obtained at this point (step 7), based on the requestes' attributes. The typical XACML decision making process can take place during this step.
- The access decision is sent to the SO (step 8). If the decision is to grant access, a signed or MAC-protected ticket is forwarded to the PEP together with details about the request (step 9). This is the first time that the node is contacted, and is only performed by an authorized party, hence not exposed to the outside world. If access is denied the decision is simply forwarded to the Requester. The Service Provider might also be informed on that based on appropriate pre-configurations.
- Now the PEP can respond to the service request through the SO (step 10). The SO can in turn send to the requester the Access Decision and the response to the Access Request. The Access Decision can be used as a token for re-accessing the same service without undergoing the authentication process.

5 Implementation Approach

There are many open-source implementations of the XACML handling and decision-making process that can be utilized for the proposed architecture. The authors chose Suns XACML [14] for this implementation, as it remains popular among developers and is actually the basis of various current open source and commercial offerings.

All of the frameworks entities are implemented using DPWS. This facilitates the discovery and description of the devices involved, and also offers control and eventing mechanisms which assist in the communication of the necessary information among the entities. Web Services for Devices (WS4D) [15] is an open source initiative which provides a number of toolkits for various platforms. The authors APIs of choice is the WS4D-JMEDS (Java-based) [16] stack as it is the most advanced and active work of the WS4D initiative, supporting almost all of the existing DPWS features and providing portability to a wide range of platforms.

The exact implementation of the frameworks entities and their communication interfaces are detailed below.

Service Orchestrator to Policy Decision Point. The SO is implemented as a DPWS peer (i.e. both a client and a server). Other than the necessary mechanisms needed to interface with the approved identity providers (which will vary depending on the specific scenario/deployment examined), it also features an "Attribute_Requirements operation. Similarly, the PDP has an "Access_Request_Operation. The latter is invoked by the SO as soon as an access request arrives from a service consumer, relaying the request for evaluation. As soon as the XACML decision-making process is completed, the PDP replies to the invocation with its access decision. As detailed in the information flow above, prior to providing a decision, it may need to invoke the "Attribute_Requirements operation on the SO, in order to inform it of the needed user attributes, getting the proper assertion as an answer.

Service Orchestrator to Policy Enforcement Point. The Policy Enforcement Point must reside on every device with resources that must be protected from unauthorized access. Other than the functional elements of the devices which the framework intends to protect (e.g. access to its sensors), one extra operation must be present on each DPWS device, namely the PEP_Operation. The SO, acting as a client, invokes this operation providing the service consumers access request along with the decision (pre-issued by the PDP) as input. If the decision accompanying the invocation is positive, the PEP replies to the SO with the resource (e.g. temperature reading) that the service consumer originally tried to access. This information is then relayed to the service consumer/requester. The above DPWS-based communication mechanisms are depicted in the figure below.

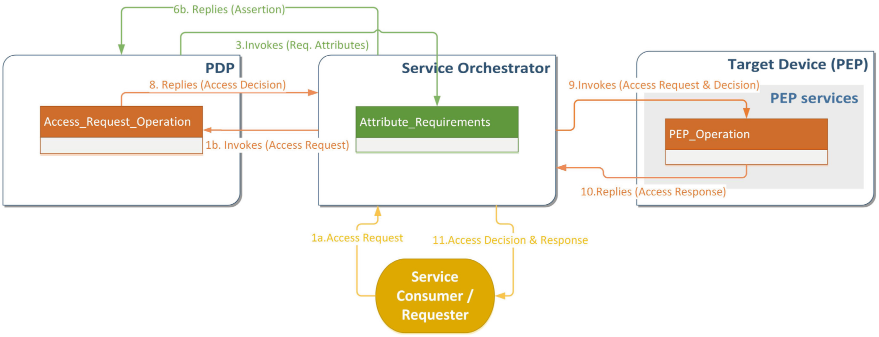
6b. Replies (Assertion)
3.Invokes (Req. Attributes)
PDP
Service Orchestrator
Target Device (PEP)
8. Replies (Access Decision)
9.Invokes (Access Request & Decision)
PEP services
Access_Request_Operation
Attribute_Requirements
PEP_Operation
1b. Invokes (Access Request)
10.Replies (Access Response)
1a.Access Request
11.Access Decision & Response
Service Consumer / Requester

Fig. 2. DPWS-based implementation of the authentication scheme

6 Security Analysis

One of the main concerns in accessing services and issuing commands, is the protection of the data being exchanged among the participating entities. In the proposed scheme the service provider has a pre-established relationship with the SO, PDP and PAP. Note that all these three entities are only functional components and therefore the exact needs in secure channel establishment depend on the actual deployment choice and cannot be specified. In a simplified approach, the SO, PDP and PAP can be part of the same entity and therefore a secure channel establishment using pre-shared keys is a viable and efficient option.

Regarding the underlying message security mechanisms, common methods that provide end-to-end security like TLS [17], (Transport Layer Security) [17] protocol and its counterpart proposed for securing UDP messages, namely DTLS [18], are considered suitable for this architecture. The cost of using TLS however, between the Requester and the SO is that the secure channel breaks at the SO and the SO has to re-encrypt the communication using the security parameters set for the link between the SO and the service provider. At the network layer solutions like the IPsec protocol and its variants that utilize header compression [19–21] can provide similar levels of protection. An alternative approach would be to utilize a subset of the mechanisms detailed in the WS-Security [22] specification, but the X509-based public key schemes included in said specification can impose a significant performance overhead [23].

7 Conclusions

As computing becomes ubiquitous, researchers and engineers aim to exploit the potential of pervasive systems, including nodes with sensors and actuators interconnected via LLNs, in order to introduce new types of services and address inveterate and emerging problems. Nevertheless, a key factor in the wide adoption and success of these new technologies will be the effectiveness with which the various security and privacy concerns are tackled. A necessary instrument in successfully addressing said issues is the presence of robust access control mechanisms.

To this end, this paper presents a work in progress on an architecture for providing access control services to heterogeneous resource-constrained devices. The authors chose the use of standardized access control mechanisms based on XACML. Moreover, the core PEP functionality is separated from the rest of the network and the decision-making process, keeping the core resource provision with the device that has the resources, while relieving it from the additional essential, yet very heavy computations that the XACML standard defines. Moreover, this approach shelters the device from direct user interaction, helping alleviate concerns that are typical to resource-constrained devices, like Denial of Service attacks.

Acknowledgments. This work was partially supported by the Greek General Secretariat for Research and Technology (GSRT), under the ARTEMIS JU research program nSHIELD (new embedded Systems arcHItecturE for multi-Layer Dependable solutions) project. Call: ARTEMIS-2010-1, Grant Agreement No.:269317.

References

1. Cui, A., Stolfo, S.J.: A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan. In: Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC 2010, pp. 97–106. ACM, New York (2010). http://doi.acm.org/10.1145/1920261.1920276
2. Parducci, B., Lockhart, H., Rissanen, E.: eXtensible Access Control Markup Language (XACML) Version 3.0, pp. 1–150 (2003). http://docs.oasis-open.org/xacml/3.0/
3. He, D., Bu, J., Zhu, S., Chan, S., Chen, C.: Distributed Access Control with Privacy Support in Wireless Sensor Networks. IEEE Transactions on Wireless Communications **10**(10), 3472–3481 (2011)
4. Yu, S., Ren, K., Lou, W.: FDAC: Toward Fine-Grained Distributed Data Access Control in Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems **22**(4), 352–362 (2011)
5. Askoxylakis, I.G., Markantonakis, K., Tryfonas, T., May, J., Traganitis, A.: A face centered cubic key agreement mechanism for mobile ad hoc networks. In: Granelli, F., Skianis, C., Chatzimisios, P., Xiao, Y., Redana, S. (eds.) MOBILIGHT 2009. LNICST, vol. 13, pp. 103–113. Springer, Heidelberg (2009)
6. Manifavas, C., Fysarakis, K., Rantos, K., Kagiambakis, K., Papaefstathiou, I.: Policy-based access control for body sensor networks. In: Naccache, D., Sauveron, D. (eds.) WISTP 2014. LNCS, vol. 8501, pp. 150–159. Springer, Heidelberg (2014)
7. Serbanati, A., Segura, A.S., Oliverau, A., Saied, Y.B., Gruschka, N., Gessner, D., Gomez-Marmol, F.: Internet of Things Architecture, Concept and Solutions for Privacy and Security in the Resolution Infrastructure. EU project IoT-A, Project report D4.2 (2012). http://www.iot-a.eu/

8. El-Aziz, A., Kannan, A.: Access control for healthcare data using extended XACML-SRBAC model. In: 2012 International Conference on Computer Communication and Informatics, Dept. of Information Science & Technology, Anna University, pp. 1–4. IEEE, January 2012
9. Zhu, Y., Keoh, S., Sloman, M., Lupu, E.: A lightweight policy system for body sensor networks. IEEE Transactions on Network and Service Management **6**(3), 137–148 (2009)
10. Colitti, W., Steenhaut, K., De Caro, N.: Integrating wireless sensor networks with the web. In: Proc. of Extending the Internet to Low Power and Lossy Networks, Chicago, IL, USA (2011)
11. Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., Waldbusser, S.: Terminology for Policy-Based Management, pp. 1–22 (2001). http://www.ietf.org/rfc/rfc3198.txt
12. Anderson, A., Lockhart, H.: SAML 2.0 Profile of XACML, Version 2.0 (2005). http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf
13. Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., Maler, E.: Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 (2005). http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
14. Sun Microsystems Laboratories, XACML. http://sunxacml.sourceforge.net
15. Web Services for Devices (WS4D). http://ws4d.e-technik.uni-rostock.de
16. WS4D-JMEDS DPWS Stack. http://sourceforge.net/projects/ws4d-javame/
17. Dierks, T., Rescorla, E.: RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2, pp. 1–104 (2008). http://tools.ietf.org/rfc/rfc5246.txt
18. Rescorla, E., Modadugu, N.: Datagram Transport Layer Security, pp. 1–31 (2012). http://tools.ietf.org/rfc/rfc6347.txt
19. Rantos, K., Papanikolaou, A., Manifavas, C.: Ipsec over ieee 802.15.4 for low power and lossy networks. In: Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access, MobiWac 2013, pp. 59–64. ACM, New York (2013)
20. Rantos, K., Papanikolaou, A., Manifavas, C., Papaefstathiou, I.: Ipv6 security for low power and lossy networks. In: 2013 IFIP Wireless Days (WD), pp. 1–8, November 2013
21. Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., Roedig, U.: Securing communication in 6LoWPAN with compressed IPsec. In: Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2011), Barcelona, Spain, June 2011
22. Lawrence, K., Kaler, C., Nadalin, A., Monzilo, R., Hallam-Baker, P.: Web Services Security: SOAP Message Security 1.1, pp. 1–76 (2006). http://docs.oasis-open.org/wss/v1.1/
23. Lascelles, F., Flint, A.: WS-Security Performance (2006). http://websphere.sys-con.com/node/204424