Secure Policy-Based Management Solutions in Heterogeneous Embedded Systems Networks

Konstantinos Rantos
Technological Educational Institute of Kavala,
Kavala, Greece
Email: krantos@teikav.edu.gr

Alexandros Papanikolaou
Technological Educational Institute of Larissa,
Larissa, Greece
Email: alpapanik@teilar.gr

Konstantinos Fysarakis and Charalampos Manifavas
Technological Educational Institute of Crete,
Heraklion, Crete, Greece
Email: {fisarakis, harryman}@epp.teicrete.gr

***Abstract*—Managing a large number of heterogeneous nodes in a network of embedded systems is a challenging task, mainly due to differences in requirements and resources. Nano nodes with very limited capabilities, such as the nodes of a Wireless Sensor Network (WSN), may not be suitable for adopting solutions designed for power nodes that have no such constraints. Using these devices in dynamic, ad-hoc infrastructures that feature a plethora of characteristics, has brought up the need for appropriate management of participating nodes to satisfy the corresponding policy restrictions. Many schemes have been proposed for various types of devices in terms of resources, ranging from the well-studied policy-based management in computer networks to the more challenging management in sensor networks. This paper identifies these schemes and proposes a framework for the secure and interoperable policy-based management of heterogeneous, resource-constrained, embedded systems networks.**

***Index Terms*—Embedded systems, policy-based management, security, XACML**

I. Introduction

Although literature lacks a precise definition of embedded systems, there is a scientific consensus on the fact that they are computer systems, often integrated into larger devices, designed to perform certain dedicated functions. Some of these devices have constrained resources, in terms of power, computing capacity, memory, and bandwidth, hence limited capabilities such as the nodes of a Wireless Sensor Network (WSN).

This heterogeneity in large scale deployments combined with the dynamic nature of these systems requires utilisation of specialised techniques in order to manage their resources and corresponding services. It should be noted that embedded systems might have the capability to support different requirements to adapt to the needs of the corresponding environments and therefore provide services that adhere to the policy of the network they join.

There is a variety of network and service management schemes available in the literature, some already in operation, as well as efforts to standardise such mechanisms. Some of the above have been specifically designed for embedded systems, whereas others have been ported to resource-constrained environments. This paper proposes an architectural framework employing, wherever possible, currently-available standards, with an emphasis on the interoperability of heterogeneous embedded systems and on the mechanisms deployed at the application layer for the protection of communicated messages, both management- and service-related.

The paper is organised as follows: Section II presents the related work on policy-based management, web services security and the currently available standards for resource-constrained devices. Section III presents the proposed management framework, as well as a critical review of the available options for securing the communicated management messages. The paper concludes in Section IV.

II. Related Work

A. Policy-Based Management

Policy-based management has successfully been implemented in various types of sensor networks. One such instance is presented in [1] and involves the use of policy-based management in body-area networks (BAN), where autonomous adaptation to changing conditions (failures, user activity, patients' clinical condition) is a requirement. The toolkit that was developed and deployed, *Ponder2* [2], allows the specification of rules in the form of event-condition-action, which enforce a given policy. Additional functionality of this toolkit includes the logical grouping of components in domains, as well as the dynamic loading of new functionality and communication protocols.

Another application of policy-based management was the SNOWMAN framework presented in [3] that allowed nodes of a WSN to autonomously organise themselves. A lightweight policy distribution protocol was developed, TinyCOPS-PR, as well as a policy information base (PIB). For facilitating scalable and localised management of WSNs, nodes are organised into three logical groups: regions, clusters and sensor nodes. The simulation results revealed that the scheme features lower power consumption compared to other schemes.

The work in [4] proposes the use of an extra middleware layer that introduces a level of abstraction, thus making it easier to describe and enforce both functional and non-functional business requirements of different end-users.

An architecture for policy-based WSN management was proposed in [5]. It distributes the management functionality across the sensor network and employs clustering for improved

management. The scheme also includes functionality for hierarchy maintenance management and cluster maintenance.

In [6] a framework for implementing policy-based management in WSN is proposed that makes use of the *Finger2* policy system which, although derived from the *Ponder2* [2] policy system, it is considerably simplified so as to run on motes. Furthermore, examples of policies are given that deal with the self-healing aspects of sensor networks. Policy-based reconfiguration is thus able to deal with various network faults.

B. Web Services Security

The Web Services Security Specification (WS-Security or WSS) [7] is part of the WS-* family of specifications published by OASIS. The protocol specifies enhancements to existing SOAP (Simple Object Access Protocol) messaging, integrating security features in the header of SOAP messages (working in the application layer), in order to provide message-level confidentiality, integrity and authentication. The main mechanisms detail signing SOAP messages (integrity, non-repudiation), encrypting SOAP messages (confidentiality) and attaching security tokens to SOAP messages (authentication). There is a variety of supported encryption, signature and security token formats (e.g. SAML Assertions [8], Kerberos tickets [9], X.509 Certificates [10], Rights Expression Language (REL) Tokens [11], UserID/Password credentials [12], as well as custom tokens).

C. DPWS – Implementing the OASIS Standards on Resource-Constrained Devices

The need to implement dynamic and secure discovery of devices and Web Services (including messaging, description, interactions, event-driven changed etc.) on resource constrained devices led to the development of the Devices Profile for Web Services specification (DPWS) [13].

The profile's architecture includes hosting and hosted services, where the former are associated to a device and are essential for device discovery and the latter are functional and reply on the hosting device for discovery. Moreover discovery services are included, enabling devices to "advertise" their presence on the network and search for other devices. Meta-data exchange services provide dynamic access to services hosted on a device and their meta-data and publish/subscribe eventing services allow other devices to subscribe to messages provided by a certain service.

The EU research project SIRENA [14] was one of the earliest implementations of DPWS on embedded devices. Their results were a foundation for the EU projects SODA [15] and SOCRADES [16] that followed, but also led to the introduction of the Service-Oriented Architecture for Devices (SOA4D) [17] and Web Services for Devices (WS4D) [18] open source programs.

SOA4D is an open source architecture which provides development toolkits (in C and Java), simplifying the development of DPWS-compliant applications (and thus the implementation of the WS-* family of protocols) for embedded devices.

WS4D is another open source initiative which provides a number of toolkits aimed at developing DPWS-compliant applications for resource-constrained devices in ad-hoc networks which are interoperable with regular W3C-specified Web Services. A detailed overview of the WS4D initiative can be found in [19].

III. Proposed Framework

A. Overview

Among the studied schemes proposed for systems with different requirements and properties, a cross-platform solution that meets the requirements of all types of embedded systems and provides interoperability, crucial for next-generation pervasive computing devices, is based on eXtensible Access control Markup Language (XACML) policies. XACML is an XML-based general-purpose access control policy language used for representing authorisation and entitlement policies for managing access to resources. However, it is also an access control decision request/response language. As such, it can be used to convey policy requirements in a unified and unambiguous manner, hence interoperable and secure, if appropriately deployed.

The above fit well into the model of a network of heterogeneous embedded systems where access to resources is provided by nodes as a service, and into the management architecture developed by IETF Policy Framework. This typical policy based management architecture combined with XACML, is mapped to a Service Oriented Architecture (SOA) network of nodes to provide protected access to their distributed resources. XACML is designed to accommodate the policy management architecture which consists of [20], [21]:

- Policy Enforcement Point (PEP): The system entity that performs access control, by making decision requests and enforcing authorisation decisions.
- Policy Administration Point (PAP): The system entity that creates a policy or policy set.
- Policy Decision Point (PDP): The system entity that evaluates applicable policy and renders an authorisation decision.
- Policy Information Point (PIP): The system entity that acts as a source of attribute values.

The types of nodes that the proposed framework aims to cover are the following:

1) Power Nodes: Nodes with high performance in terms of computing power and no particular resources restrictions.
2) Micro/Personal nodes: Nodes which do not have the capabilities of power nodes yet neither suffer from the restrictions that nano nodes face, e.g. a smart card.
3) Nano nodes: Small devices with limited capabilities and resources in terms of computations or power supply, e.g. a sensor node.

It is worth noting that by the term "power node" we imply a node possessing the functionality of a micro node, without any resource constraints.

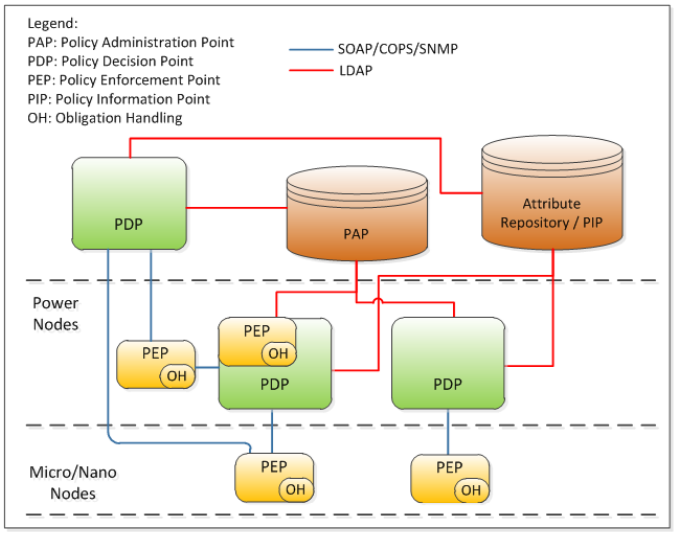

Fig. 1. Policy-Based Management (PBM)

In a typical data flow model, authorisation requests for accessing nodes' resources, are forwarded to PDPs, through the context handler which is responsible for orchestrating the communications and collecting the required attributes from the PIP. The PDP evaluates the request against the policy restrictions taken from the PAP and issues an authorisation decision which the PEP has to enforce together with some (optional) obligations and/or advices. Given its potentially limited resources, a PEP might not have the capacity to support the context handler functionality, which can be offloaded to a PDP or another infrastructure component, such as a base station in a WSN architecture.

XACML defines that requests to context handler should be sent by a PEP in its native request format. However offloaded to PDPs context handler functionality and interoperability in dynamically formulated environments, demand abandoning proprietary communications and adopting common messaging formats.

The architectural framework proposed here is an enhancement of the one adopted by pSHIELD [22], also considering the different types of nodes and their corresponding capabilities, as well as the requirements for adequately securing the connections between the policy repository/orchestrator and the participating nodes. These security measures are considered essential in a wireless and hostile environment where policy information might prove a valuable source for attackers to exploit. According to the proposed architecture, which is also depicted in Fig. 1, different approaches are adopted for the different types of nodes met in heterogeneous systems, with diversified capabilities:

1) Power Nodes that incorporate both PEP (typically on a node component) and PDP functionality, serving requests originating from other nodes or for its own needs.
2) Power Nodes that are only acting as PEPs while the PDP functionality is provided by another system, such as a base station.
3) Power Nodes that only act as PDPs serving requests originating from other nodes.
4) Micro or Nano nodes acting only as PEPs enforcing authorisation decisions rendered by a PDP. Such nodes, due to their resource restrictions should typically not support the PDP functionality.

Given the dynamic nature and the need for self-configurability, there will be situations where there is no coordination or central control over the network of nodes. Therefore, nodes that just joined the network have to discover which entity is responsible for making access decisions. In this case the corresponding systems, such as power nodes or base stations, have to advertise their capabilities regarding PDP functionality while PEPs have to be able to discover PDPs and their corresponding provided services.

While XACML defines the structure and content of access requests and responses exchanged among PEPs and PDPs, it does not provide any details regarding mechanism(s) used to transfer these messages, thus providing the necessary flexibility to adapt to diversified environments. Protocols that have been proposed for the communications among PEPs and PDPs are COPS, SNMP and LDAP, which matches the requirements for accessing the policy repository and PIP.

Resource-constrained nodes participating in a SOA should typically avoid connection overhead caused by expensive protocols such as TCP, although this option is at the expense of reliability which has to be provided by other layers protocols. Therefore, the choice between SOAP over UDP or HTTP/TCP for resource-constrained devices is quite straightforward with OASIS characterising SOAP over UDP solution as the natural choice [23]. SOAP over UDP specifies the way SOAP envelopes are carried in user datagrams while respecting the requirements for UDP messages being received in one chunk, which demands valuable resources on small

embedded systems. If this is not an issue or reliability is a first order requirement on more powerful nodes, alternatives such as HTTP/TCP can be used instead. So choosing the most appropriate communication protocol(s) is actually a case of communication overhead as opposed to single message demands.

Whatever the chosen communication protocol, one of the issues that need considering is the protection of policy messages exchanged among PAP, PEP, PDP, and repository.

B. Protecting Policy Messages

The exchange of unprotected policy messages might reveal useful information to attackers who will try to easily identify policy restrictions and do a mapping of the security measures taken for the specific environments, hence exploit potential vulnerabilities. Moreover, in a more active approach, an attacker might modify policy related messages, such as authorisation requests and/or decisions, obligations or advices in an attempt to downgrade adopted measures and bypass access controls. Masquerading is another threat to the system's integrity, where an attacker sends forged messages pretending to be a legitimate user. Lack of authorisation requirements on requests might allow an attacker to make such message injections.

To avoid the aforementioned problems, security measures have to be taken to protect message confidentiality, integrity and authentication. These measures can be deployed in various layers in the OSI protocol stack, such as the application layer with the mechanisms described herein, the network layer with IPsec, including light versions to satisfy restricted resources systems requirements [24], and lower layers, such as the IEEE802.15.4 link-layer inherent security protocols [25]. In a comparative analysis made between IPsec and IEEE802.15.4 link-layer security [26], in some cases IPsec scales better while also offering end-to-end security. The choice is also affected by the node capabilities, especially when protection is required for micro/nano nodes with very limited resources. This paper focuses on solutions that apply to the application layer instead.

Communications between a PDP (running on a power node) and the repository can be protected using Transport Layer Security (TLS), as the power node can support the heavy computations required by TLS. Given that there are no resources restrictions on power nodes there are no major obstacles, besides key management, to deploy TLS or even IPsec on these nodes.

TLS/SSL is adopted by DPWS as the mechanism for providing confidentiality, integrity and authentication of messages exchanged between a client and a service, running on a node. However, the inherent expensive computations of TLS or IPsec (such as the TLS handshake protocol) do not match the requirements of resource-constrained environments, for the protection of the communications that take place among the less powerful micro/nano nodes (PEPs), or between PEPs and power nodes (PDPs) [27], [28]. A TLS implementation on the Sun SPOT (Small Programmable Object Technology) Java-enabled WSN platform, has shown that the network lifetime is reduced by 70% [29].

Several lightweight alternatives based on TLS/SSL have been proposed for resource-constrained environments. One such approach was proposed in [30], which uses ECC for key exchange and authentication, RC4 for encryption and MD5 for integrity check. According to the presented experimental results, it was able to complete a full SSL handshake within 2 seconds. Tiny 3-TLS proposed in [31] is an extension and adaptation of the TLS handshake sub-protocol, tailored for securing communications between sensing nodes and remote monitoring terminals. This protocol relies on the existence of an intermediate node, the sink node, which in the proposed framework can be assigned to a power node.

In [32], an implementation of SSL was attempted through an enhanced version of *Sizzle* (a tiny-footprint HTTPS stack) [33]. Measurements were performed on Telos motes and it was concluded that the exploitation of features such as session reuse and persistent HTTP(S) can spare multiple executions of the key exchange phase, which is the most energy-demanding part of the protocol. What is more, it was also shown that the extra cost for encrypting/authenticating application data with SSL is around 15%. Again, the key exchange phase is performed via ECC since it is significantly more efficient than the RSA alternative.

In [30] a scheme for implementing SSL-based lightweight security on the SNAIL [34] protocol for WSN is proposed that uses ECC for key management.

Considering the less powerful micro or nano nodes, deployed protection mechanisms have to be based on lightweight cryptographic protocols that satisfy the needs and match the capabilities of constrained environments. A solution that can be adopted for this purpose is WS-Security and the corresponding cryptographic mechanisms, i.e. XML encryption and signatures (enveloped, enveloping or detached) or a combination of those depending on the particular requirements to provide confidentiality and integrity of the exchanged messages.

A structured exchange of secure XACML messages using XML encryption and signature is provided by SAML specifications (Security Assertion Markup Language) [35], thus offering the required protection at the application layer. SAML is a plaform-independent, XML-based standard for exchanging authentication and authorisation information. SAML assertions are typically transferred embedded using HTTP or XML-encoded SOAP messages that are transferred over HTTP or UDP [23]. OASIS has defined a profile in [36] for the integration of SAML with XACML and, among the others, the use of SAML for the secure transmission of XACML requests and responses.

Therefore, considering the networking requirements and the corresponding security mechanisms, the most appropriate options for securing XACML messages at the application layer, while providing interoperability, are the following:

- SAML-integrated XACML messages transferred using the SOAP protocol (over UDP).
- SOAP-encapsulated XACML messages protected with TLS. Such an approach requires using expensive TCP

TABLE I
WS-SECURITY AND TLS BENCHMARK RESULTS FOR 25 CONCURRENT REQUESTORS [28]

Security Mechanism	Messages per second	CPU load	Throughput (kB/s)
X.509 XML Signature & Encryption	352	99	2,403
WS Secure Conversation XML Signature & Encryption	798	98	5,679
SSL with HTTP Basic	2,918	95	3,181
None (message routing only)	5,088	96	5,419

communications to transmit the resulting TLS messages. Another option is to use an adaptation of TLS over UDP, namely DTLS, specified in [37].

Both of the above solutions are typically independent of the protocols used at the underlying layers, allowing them to adapt in many environments, hence satisfying the interoperability requirement.

In either of the aforementioned approaches, one of the main problems related to the exchanged messages' protection is key management, especially when considering the following:

1) Communications might take place ad-hoc between nodes that do not have an established trust relationship, hence they do not (pre-)share any secrets. Dynamic structures and self-configuration capabilities demand for more flexible mechanisms.
2) Some nodes might not support public-key technologies, which further complicates the processes of establishing trust relationships and keys.

Web Services Secure Conversation [38] is a WS-Security add-on which introduces, similarly to TLS, a session key to secure communication across one or more messages. The aim of the specification is to establish security context, share, renew, amend or cancel said context as well as derive (potentially more efficient) sessions keys from the aforementioned context. When multiple message exchanges are involved WS-SecurityConversation has proven to be more efficient than a plain WS-Security implementation [39], but the former requires the presence of other WS-* protocols as well, like WS-Trust, so the added complexity should also be considered.

In situations where point-to-point confidentiality and integrity are adequate, TLS could be considered as an alternative. Unlike WSS though, TLS cannot offer end-to-end (message-level) security and it is not as flexible when application-level proxy servers are involved. Still, the performance overhead is significant with the standard WS-Security implementation and this is an area where further research is required (and already being conducted) in order to improve its usability in resource-constrained devices (see Table I).

These inherent key management problems, which might affect the decision between the two aforementioned solutions for securing XACML messages, especially in resource-constrained environments, have attracted a lot of attention in the research community and many schemes have been proposed within this context. A survey and taxonomy on proposed wireless sensor networks key management schemes is provided in [40].

A working model that combines the use of XACML and DPWS, in order to add fine-grained secure service policies to the latter, has been introduced in [41]. The proposed model is based on X.509 certificates and tailored to the needs of a "smart home" environment but can easily be adopted to other scenarios (e.g. in our proposed scenarios, power nodes could adopt the CA roles described in the above work). It is demonstrated that processing time overhead (especially in the XACML lookup phase) is affected by the number of policies maintained at the PDP, it is therefore imperative to fine-tune the amount of queries the PDP needs to lookup or even the number of PDPs available on the network (e.g. segregating service discovery and application PDP duties), depending on application requirements.

IV. CONCLUSION

Managing nodes in a heterogeneous network of embedded systems is a challenging task, due to the differences in characteristics among the nodes. Policy-based management is an effective and standardised option for managing such networks. However, securing the exchanged policy messages across the network introduces new challenges, due to the fact that embedded systems nodes are usually resource-constrained and thus classical cryptographic approaches and/or mechanisms are either not applicable or require modifications.

This paper proposes a policy-based management framework for embedded systems, based on XACML, offering a cross-platform, general-purpose solution that can be used for specifying and enforcing access control policies. Emphasis is given on the security of the communicated request/response and service provision messages. These messages can be transmitted according to a web-service-based approach, which also offers a variety of options for securing the transmission. Each entity type may be assigned more than one role, in scenarios where this is feasible, thus allowing the system to offload computationally demanding tasks to other infrastructure components. A critical issue is the protection of policy messages exchanged among the framework's entities. The mechanisms deployed for this task are closely dependent on the application requirements (e.g. the need for point-to-point or message-level confidentiality), as well as the potential support for more advanced security characteristics, such as node trust-sharing schemes and security context awareness. Moreover, from a cryptographic point of view a scheme may seem far more efficient but the overall overhead may actually be quite significant and render the scheme impractical for large-sized networks. Before making a decision, several parameters need to be considered in order to adapt the proposed framework to a specific application and its requirements.

ACKNOWLEDGMENT

This work was funded by the European Community's Seventh Framework Programme Artemis nSHIELD (new em-

bedded Systems arcHItecturE for multi-Layer Dependable solutions) project. Call: ARTEMIS-2010-1, Grand Agreement No.: 269317.

REFERENCES

[1] S. L. Keoh, K. Twidle, N. Pryce, A. E. Schaeffer-Filho, E. Lupu, N. Dulay, M. Sloman, S. Heeps, S. Strowes, J. Sventek, and E. Katsiri, "Policy-based management for body-sensor networks," in *4th IEEE International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*. AAchen, Germany: Springer-Verlag, March 2007, pp. 92–98.

[2] "Ponder2," http://www.ponder2.net.

[3] S.-H. Cha, J.-E. Lee, M. Jo, H. Y. Youn, S. Kang, and K.-H. Cho, "Policy-based management for self-managing wireless sensor networks," *IEICE Transactions on Communications*, vol. E90-B, no. 11, pp. 3024–3033, 2007, special section: Next Generation Network Management.

[4] N. Matthys and W. Joosen, "Towards policy-based management of sensor networks," in *3rd International Workshop on Middleware for Sensor Networks*, ser. MidSens '08. Leuven, Belgium: ACM, 2008, pp. 13–18.

[5] Z. Wenbo and X. Haifeng, "A policy based wireless sensor network management architecture," in *3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 1–3 November 2010, pp. 552–555.

[6] T. Bourdenas and M. Sloman, "Starfish: Policy driven self-management in wireless sensor networks," in *ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, ser. SEAMS '10. Cape Town, South Africa: ACM, 2010, pp. 75–83.

[7] OASIS, "Web services security: SOAP message security 1.1 (WS-seurity 2004)," http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, 1 February 2006.

[8] ——, "Web services security: SAML token profile 1.1," http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf, 1 February 2006.

[9] ——, "Web services security kerberos token profile 1.1," http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf, 1 February 2006.

[10] ——, "Web services security 3 X.509 certificate token profile 1.1," http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf, 1 February 2006.

[11] ——, "Web services security rights expression language (REL) token profile 1.1," http://www.oasis-open.org/committees/download.php/16687/oasis-wss-rel-token-profile-1.1.pdf, 1 February 2006.

[12] ——, "Web services security usernametoken profile 1.1," http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf, 1 February 2006.

[13] ——, "Devices profile for web services version 1.1," http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.pdf, 1 July 2009.

[14] "Service Infrastructure for Real time Embedded Networked Applications (ITEA SIRENA)," http://www.sirena-itea.org, 2003–2005.

[15] "Service-Oriented Device & Delivery Architecture (SODA)," http://www.soda-itea.org/.

[16] "Service-Oriented Cross-layer Infrastructure for Distributed Smart Embedded Devices (SOCRADES)," http://www.socrades.eu/Home/default.html.

[17] "Service-Oriented Architecture for Devices (SOA4D)," http://cms.soa4d.org/.

[18] "Web Services for Devices (WS4D)," http://ws4d.e-technik.uni-rostock.de/.

[19] E. Zeeb, G. Moritz, D. Timmermann, and F. Golatowski, "WS4D: Toolkits for networked embedded systems based on the devices profile for web services," in *39th International Conference on Parallel Processing Workshops*. IEEE, 2010, pp. 1–8.

[20] OASIS, "eXtensible Access Control Markup Language (XACML) version 3.0," http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf, 10 August 2010.

[21] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, "Terminology for policy-based management," Internet Engineering Task Force, RFC 3198, November 2001, http://www.ietf.org/rfc/rfc3198.txt.

[22] A. Fiaschetti, A. Morgani, R. Baldelli, M. Chowdhury, A. Taglialatela, V. Suraci, A. Palo, and S. Drakul, "SPD middleware and overlay functionality report," Artemis Joint Undertaking, FP7, Deliverable D5.4, 2011, http://www.pshield.eu/index.php?option=com_docman&task=doc_download&gid=248&Itemid=37.

[23] OASIS, "SOAP-over-UDP version 1.1," http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/os/wsdd-soapoverudp-1.1-spec-os.pdf, 1 July 2009.

[24] S. Raza, S. Duquennoy, A. H. M. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communicqation in 6LoWPAN with compressed IPsec," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops*, ser. DCOSS, 27–29 June 2011, pp. 1–8.

[25] N. Sastry and D. Wagner, "Security considerations for ieee 802.15.4 networks," in *3rd ACM workshop on Wireless security*, ser. WiSe '04. Philadelphia, PA, USA: ACM, 2011, pp. 32–42.

[26] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of things – a comparison of link-layer security and IPsec for 6LoWPAN," *Security and Communication Networks*, 2012, in press.

[27] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *SenSys '04 Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM, Ed., 2004, pp. 162–175.

[28] F. Lascelles and A. Flint, "WS security performance. Secure conversation versus the X509 profile," http://websphere.sys-con.com/node/204424, 17 April 2006.

[29] D. E. Boyle and T. Newe, "On the implementation and evaluation of an elliptic curve based cryptosystem for java enabled wireless sensor networks," *Sensors and Actuators A: Physical*, vol. 156, no. 2, pp. 394–405, 2009.

[30] W. Jung, S. Hong, M. Ha, Y.-J. Kim, and D. Kim, "SSL-based lightweight security of IP-based wireless sensor networks," in *2009 International Conference on Advanced Information Networking and Applications Workshops*, 2009, pp. 1112–1117.

[31] S. Fouladgar, B. Mainaud, K. Masmoudi, and H. Afifi, "Tiny 3-TLS: A trust delegation protocol for wireless sensor networks," in *Security and Privacy in Ad-Hoc and Sensor Networks*, ser. LNCS, L. Buttyan, V. Gligor, and D. Westhoff, Eds. Springer Berlin / Heidelberg, 2006, vol. 4357, pp. 32–42.

[32] V. Gupta and M. Wurm, "The energy cost of SSL in deeply embedded systems," Sun Labs, Sun Microsystems, 16 Network Circle, Menlo Park, CA 94025, USA, SMLI TR-2008-173, June 2008.

[33] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, and S. C. Shantz, "Sizzle: A standards-based end-to-end security architecture for the embedded internet," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 425–445, December 2005.

[34] S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J.-E. Kim, "SNAIL: an ip-based wireless sensor network approach to the internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, December 2010.

[35] OASIS, "Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0," http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 15 March 2005.

[36] ——, "SAML 2.0 profile for XACML, version 2.0," http://docs.oasis-open.org/xacml/3.0/xacml-profile-saml2.0-v2-spec-en.pdf, 10 August 2010.

[37] E. Rescorla and N. Modadugu, "Datagram transport layer security," Internet Engineering Task Force, RFC 4347, April 2001, http://www.ietf.org/rfc/rfc4347.txt.

[38] OASIS, "WS-SecureConversation 1.4," http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.4/os/ws-secureconversation-1.4-spec-os.pdf, 2 February 2009.

[39] H. Liu, S. Pallickara, and G. Fox, "Performance of web services security," in *13th Annual Mardi Gras Conference*, Baton Rouge, LA, USA, 2005, pp. 72–78.

[40] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63–75, March 2010.

[41] A. Müller, H. Kinkelin, S. K. Ghai, and G. Carle, "A secure service infrastructure for interconnecting future home networks based on DPWS and XACML," in *2010 ACM SIGCOMM Workshop on Home Networks*, ser. HomeNets '10. New Delhi, India: ACM, 2010, pp. 31–36.