Policy-Based Access Control for Body Sensor Networks

Charalampos Manifavas, Konstantinos Fysarakis, Konstantinos Rantos,
Konstantinos Kagiambakis, Ioannis Papaefstathiou

Policy-based Access Control for Body Sensor Networks

Charalampos Manifavas[1], Konstantinos Fysarakis[2], Konstantinos Rantos[3], Konstantinos Kagiambakis[1], and Ioannis Papaefstathiou[2]

[1] Dept. of Informatics Engineering, Technological Educational Institute of Crete, Heraklion, Crete, Greece
harryman@ie.teicrete.gr,kostiskag@gmail.com
[2] Dept. of Electronic & Computer Engineering, Technical University of Crete, Chania, Crete, Greece
kfysarakis@isc.tuc.gr,ypg@mhl.tuc.gr
[3] Dept. of Computer and Informatics Engineering, Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece
krantos@teikav.edu.gr

Abstract. Sensor nodes and actuators are becoming ubiquitous and research efforts focus on addressing the various issues stemming from resources constraints and other intrinsic characteristics typically associated with such devices and their applications. In the case of wearable nodes, and especially in the context of e-Health applications, the security issues are exacerbated by the direct interaction with the human body and the associated safety and privacy concerns. This work presents a policy-based, unified, cross-platform and flexible access control framework. It adopts a web services-compliant approach to enable secure and authorized fine-grained access control to body sensor network resources and services. The proposed scheme specifically considers the very limited resources of so-called nano nodes that are anticipated to be used in such an environment. A proof-of-concept implementation is developed and a preliminary performance evaluation is presented.

Keywords: body sensor networks; policy-based access control; XACML; DPWS; web services; security

1 Introduction

Sensor nodes and wireless sensor networks constitute a well-established technology with many applications, ranging from home and industrial automation, to smart cities, agriculture and power metering, logistics, e-Health and assisted living monitoring. A leading solution adopted by many schemes for enabling interaction with and providing sensitive information to remote parties is Service Oriented Architecture (SOA). It constitutes an attractive design approach for many types of networks, including those that consist of nodes with limited capabilities. Such a network is a body sensor network (BSN) [29] which comprises a number of low-power implanted, wearable (on-body) or in close distance

wireless sensors and actuators. The environmental and physiological sensors of a BSN provide vital information to medical staff, who can remotely monitor and possibly control users medical treatment. For such an application there are many security concerns [8], including secure transmission of sensitive information to (remote) medical staff, unaltered instructions that reach patients actuators, robust entity authentication and access control mechanisms.

Among the access control schemes that have gained popularity are those where decisions are made based on policy restrictions, such as the standardized by the OASIS, eXtensible Access Control Markup Language (XACML), an XML-based general purpose policy decision language. Besides being used for representing authorization and entitlement policies for managing access to resources, XACML provides a processing model for evaluating requests and making decisions based on a well-defined set of policies. The architecture presented in this paper is an implementation of the XACML solution outlined in [19], adapted to the environment of a BSN. Access to BSN nodes resources is controlled by the use of XACML, facilitating the separate and scalable deployment of nodes on heterogeneous networks and platforms, based on patients' needs.

This paper is organized as follows: Section 2 analyzes the basic characteristics of a BSN network architecture and presents related work, section 3 details the proposed scheme and presents a proof-of-concept implementation and, finally, section 4 features the closing remarks.

2 The Body Sensor Network Case

In a typical BSN used for e-Health purposes, environmental and physiological sensors are deployed to gather and send medical staff important information, such as blood pressure and body and room temperature. Actuators controlled by authorized medical staff can also be deployed for remote treatment, such as an automatic insulin injection device. These sensitive actions need strict access control decisions before being authorized so that users privacy andor safety are not threatened. Security requirements related to BSNs are well documented in the literature [8, 22] and include data confidentiality, message authentication and availability.

The types of nodes, in terms of computing capabilities, found in a BSN include *power nodes*, i.e. nodes with medium to high performance computing power and no particular resources restrictions (e.g. a mobile phone, a laptop or a dedicated sink node) and *micro/nano nodes*, i.e. small devices with limited capabilities and resources, such as computational power, memory, storage space and energy. The latter are typically the on-body or implanted nodes found in a BSN and their resource constraints have been considered in the design of the proposed solution.

2.1 Related Work

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 [13, 12, 30, 16].

Some of the proposed mechanisms are based on the use of public-key cryptography, a choice that is very expensive for nano nodes found in a BSN. More importantly, 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 [25] 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, which has different requirements than a BSN considered in this paper.

In [11] 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 [31], 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 Proposed Framework

The framework presented in this paper is based on the standardised XACML architecture to provide a cross-platform solution that can typically be deployed in various types of embedded systems while satisfying interoperability, an important requirement for next-generation pervasive computing devices.

An XACML architecture typically consists of the following main components:

- *Policy Enforcement Point (PEP):* Performs access control, by making decision requests and enforcing authorization decisions [18, 27].
- *Policy Decision Point (PDP):* Evaluates requests against applicable policies and renders an authorization decision [18].
- *Policy Administration Point (PAP):* Creates and manages policies or policy sets [18].
- *Policy Information Point (PIP):* Acts as a source of attribute values [18].

In the proposed architecture the sensor nodes and actuators, which have direct access to resources, expose their functional elements to the PEP. These nodes are micro/nano nodes and are not expected to have the capacity to accommodate additional functionality. All the above XACML components can run on a single system or, in a more distributed approach, on different systems based on their distinctive capabilities. The latter is the model that fits the environment of a BSN comprising a number of nodes.

3.1 Implementation Approach

There is a variety of tools and APIs available to implement the presented access control framework, each with its own merits and peculiarities, although application development on micro and nano nodes is a challenging task due to inherent

resource constraints. For the entities deployed on power nodes, i.e. the PIP, PDP and PAP, the XACML handling and decision-making engine can be adopted from any open source implementation, including Suns XACML implementation [4], PicketBox XACML [2] and the Enterprise Java XACML project [1]. Considering the available options, Suns XACML is a solid choice, as it remains popular among developers and is actually the basis of various current open source and commercial offerings.

Regarding the use of web services to expose the various node features and services to the rest of the entities and users, the authors propose the adoption of the Devices Profile for Web Services (DPWS) specification [7]. A multifunctional sensor embedded on a patient's body will have a single hosting service but may feature various hosted services (e.g. a temperature service, a heart-rate service etc.). Discovery services are included [17], thus the device can advertise its presence on the network and search for other devices. Publish and subscribe eventing mechanisms allow clients to subscribe to services provided by devices.

In terms of opensource resources aimed at developing DPWS-compliant applications for resource-constrained devices, Service-Oriented Architecture for Devices (SOA4D, [3]) provides development toolkits in C and Java. Alternatively, Web Services for Devices (WS4D, [5]) is another open source initiative which provides a number of toolkits for various platforms.

For the proposed scheme to be operational each devices functional elements must be represented by an appropriate DPWS entity and its corresponding operations. Assuming a simple temperature sensor, for instance, a node is programmed as a DPWS device which hosts a temperature service featuring various operations:

- A *GetTemperature* operation which, when invoked, will return the patients current temperature.
- A more complex *TemperatureEvent* operation which, by exploiting the WS-Eventing [9] mechanism, allows a client device (e.g. doctors device) to subscribe to the service and get temperature updates at set intervals as well as event notification messages when the temperature exceeds a certain threshold.
- An additional *SetTemperatureThreshold* operation which, when invoked, allows setting/updating the abovementioned warning threshold.

Similarly, the XACML-related elements of each node must be represented as DPWS devices, clients or peers. The approach adopted by the authors involves a DPWS client on the temperature sensor node described above. DPWS is then used to discover and use the PDP service implemented on a control/gateway node. The process followed when a user tries to access a sensors functional elements (e.g. the temperature reading) is depicted in Fig. 1.

In more detail, assuming a doctor tries to access the temperature sensors features (Step 1), the request is automatically forwarded to the devices PEP (Step 2). The PEP can then invoke the *AccessRequest* operation on the control nodes PDP service (Step3), sending a properly formulated access request to the PDP. When the PDP is done evaluating (Step 4) the request based on subjects

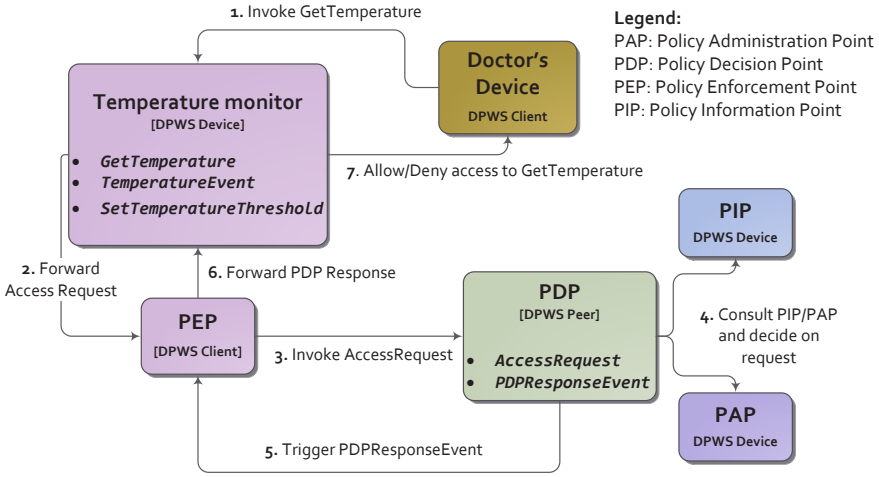

Fig. 1. PBAC implementation using DPWS

attributes and policy rules, it can, in turn, trigger its *PDPResponseEvent* (to which the sensors PEP client subscribes during initialization), returning the authorization decision. This decision is then conveyed to the functional operation of the device, thus granting or denying access to the *GetTemperature* operation the doctor tried to invoke. The PDP to PAP and PIP entities functionality are, equivalently, developed as DPWS devices and clients, exploiting the integrated discovery and subscription mechanisms, thus bypassing the need to use other protocols (e.g. LDAP).

Protection of PBAC messaging. Unprotected policy messages would expose the systems security, revealing private information to attackers who might also try to identify policy restrictions and do a mapping of the security measures taken for the specific environments. One could also masquerade as a legitimate entity or modify policy related messages, in an attempt to downgrade adopted measures and bypass access controls.

Security measures can be deployed on various layers of the network stack, with the most prominent being those that protect messages at the application or network layer and can provide end-to-end message protection. Well-known security mechanisms for these layers are the TLS (Transport Layer Security) [10] protocol and the variant proposed for securing UDP messages, namely DTLS [24], as well as IPsec and its variants that utilize header compression [21, 20, 23], for the network layer. An alternative approach would be to utilize a subset of the mechanisms detailed in the WS-Security [15] specification, but the X509-based public key schemes included in said specification can impose a significant performance overhead [14].

Communications between the context handler and the PIP can typically utilize any protection mechanism as it is anticipated they will operate on a power node, hence without significant restrictions.

3.2 Proof-of-Concept Implementation

A proof of concept implementation of the PBAC scheme presented in this work was developed using Suns XACML as a basis for the policy and access control mechanisms. The WS4D-JMEDS API [6] was used for the creation of the necessary DPWS devices. The implementation consists of the following modules:

- An application that runs on the sensors and which implements the access to the functional elements of the sensor (e.g. temperature reading) as well as the communication with the sink node. A security mechanism was also developed, based on the AES algorithm in CBC mode and pre-shared secret keys, to guarantee that only the legitimate sink node/bridge can access the sensors. When connected to the bridge, sensors ignore all other connection requests. Moreover the security mechanism protects the messages from eavesdropping on the sensors-sink node communications.
- A sink node application that bridges the BSN, which in this case operates over 802.15.4, to the standard network infrastructure. This application has to be deployed on a device equipped with dual 802.15.4 & Ethernet/wireless Ethernet functionality.
- The DPWS Provider module which discovers available sensors (via the sink node), probes said sensors to discover their functionality and then maps this functionality to a corresponding DPWS device. The DPWS device created for each of the discovered sensors includes the necessary operations to realize the PEP functionality, as well as the conversion of all low level messages transmitted to and from the sensors to a DPWS compatible form. The communication of the PEP(s) to the PDP must also be protected, as malicious tampering of the policy messages exchanged by the PBAC entities can compromise the access control efforts. To this end, a security mechanism based on the AES/CCM [28] authenticated encryption algorithm was implemented. Deployment of this mechanism guarantees that the PBAC-related messages exchanged between PEP and PDP (when the former seeks the authorization status of a specific clients request), are fully protected in terms of confidentiality, integrity and authenticity.

Performance Evaluation. The performance of the proof of concept implementation was evaluated on a test-bed featuring a SunSPOT mote [26] running the sensing application. Another Sun-SPOT mote was connected to a personal computer acting as a sink node. The DPWS Provider application was deployed on the same computer system. In a real-world application the bridge/DPWS Provider functionality could be deployed on any smart device with dual 802.15.4 & Ethernet/wireless Ethernet connectivity, even a small embedded or wearable

device, as depicted in Fig. 2. The PDP/PIP/PAP application was running on a separate computer system which also stored the policy files. This system also featured a client application developed to query the sensors for benchmarking purposes. SunSPOTs communicate via the 802.15.4 radio, while the personal computers communicated via wired Ethernet.

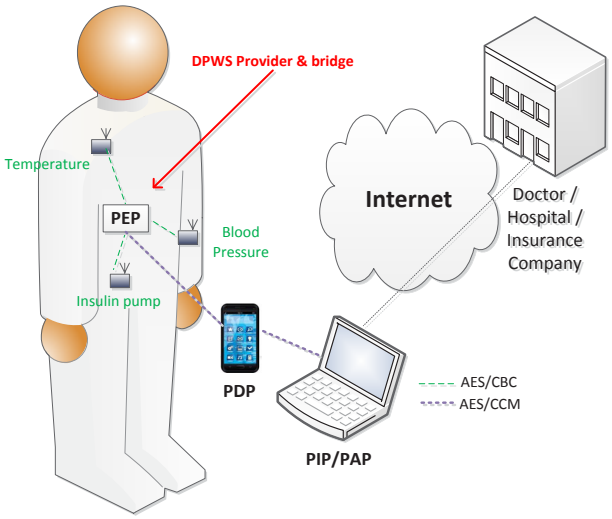

Fig. 2. Proposed deployment of the proof-of-concept PBAC application

A total of 50 consecutive requests were issued from the client application to the sensor. In order to evaluate the delay imposed by the proposed scheme, the sensor featured both a PEP-protected operation (*GetTemperature*) that the test client was allowed to invoke by the current policy set and an unprotected operation (*GetTemperatureUnprotected*) which could be invoked immediately (without going through the policy enforcement point for authorization). Aiming to also weigh the impact of the security mechanisms, the assessment included scenarios with and without encryption on both the SunSPOT-Provider link (plaintext vs. AES-CBC) and the PEP-to-PDP link (plaintext vs. AES/CCM). The response time, averaged over 50 requests, including the overhead when considering a totally unprotected (access control- and security-wise) operation as baseline, can be seen in Fig. 3.

The bulk of the delay can be attributed to the communication between the SunSPOT and the Provider, as was evident from timing tests run concurrently on the client side and the Provider side. E.g. the results of such a test, run with AES-CBC protection on the SunSPOT messages and no protection on the PEP-PDP communication, indicated that out of the 527,3ms client-side delay (on average, for 50 requests) when invoking an unprotected (i.e. no PBAC in-

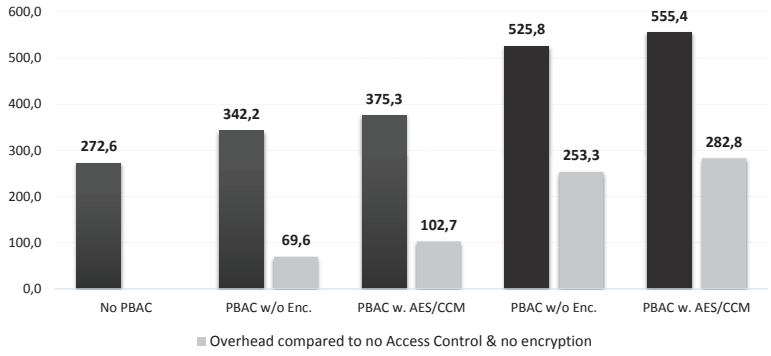

Fig. 3. Response time (in ms) for a single request, averaged over 50 data points. Columns in dark gray depict the scenario where there is no security between the sensor and the Provider, while black columns correspond to the scenarios where AES-CBC encryption was used to protect said link.

volved) operation, 449,45ms was the average time that the Provider had to wait until it got a reply from the SunSPOT. Therefore the overhead of the DPWS communication between client and the Provider (i.e. the DPWS device that mirrors the sensors functionality) was 77,85ms. Another important aspect is that when changing the policy so that the invocation of the protected operation by our test client is denied, the response time is negligible, as the request is rejected by the PEP and is never forwarded to the sensor. In a test run of 50 such unauthorized requests, the average response time of the DPWS device was just 8,39ms. It should be noted that, regarding policy look-ups, the authors chose to implement the system so that the PEP checks with the PDP for every single request, considering scenarios where policies change dynamically (even in an automated fashion when certain conditions are triggered), and where it is desirable to have the access control system enforce said changes in real-time. In deployments where policy changes are expected to be infrequent or less dynamic in nature, access tokens with a predetermined validity period (e.g. 30 minutes) could be introduced to reduce the load on the PDP.

4 Conclusions

In this paper we proposed a framework for controlling access in BSNs comprising of nodes with limited resources based on systems policy. Instead of proposing a proprietary solution typically applicable only to a network comprising of homogeneous nodes, the proposed framework is based on existing Internet and access control standards, facilitating the deployment of interoperable solutions. The aforementioned technologies and their applicability to various heterogeneous types of nodes have been investigated and relevant solutions have been identified. The results of these efforts include a proof-of-concept implementation, which is presented in this work along with an initial performance assessment. Further

evaluation, including deployment on alternative platforms, both in terms of the sensors (to include devices less capable than the SunSPOTs) as well as the bridge/Provider and access control entities (to include embedded systems and smart devices), also considering alternative security mechanisms and comparison to existing schemes, will be presented in future work. This paper focused on authorization aspects, but another important building block is the user authentication, which will also be investigated in future work, along with suggestions on adapting the utilized standards to better facilitate BSN and IoT deployments in general.

Acknowledgments. This work has been 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. Enterprise Java XACML, http://code.google.com/p/enterprise-java-xacml/
2. PicketBox XACML, https://community.jboss.org/wiki/PicketBoxXACMLJBossXACML
3. Service-Oriented Architecture for Devices (SOA4D), http://cms.soa4d.org/
4. Sun Microsystems Laboratories, XACML, http://sunxacml.sourceforge.net
5. Web Services for Devices (WS4D), http://ws4d.e-technik.uni-rostock.de
6. WS4D-JMEDS DPWS Stack, http://sourceforge.net/projects/ws4d-javame/
7. Devices profile for web services, version 1.1 (2009), http://docs.oasis-open.org/ws-dd/dpws/1.1/os/
8. Alhaqbani, B., Fidge, C.: Access control requirements for processing electronic health records. In: Arthur Ter Hofstede Boualem Benatallah, Paik, H.Y. (eds.) Proceedings of the 2007 international conference on Business process management. pp. 371–382. Springer-Verlag (2007)
9. Box, D., Cabrera, L.F., Critchley, C., Curbera, F., Ferguson, D., Graham, S., Hull, D., Kakivaya, G., Lewis, A., Lovering, B., Niblett, P., Orchard, D., Samdarshi, S., Schlimmer, J., Sedukhin, I., Shewchuk, J., Weerawarana, S., Wortendyke, D.: Web Services Eventing (WS-Eventing) (2006), http://www.w3.org/Submission/WS-Eventing/
10. Dierks, T., Rescorla, E.: RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2 (2008), http://tools.ietf.org/rfc/rfc5246.txt
11. El-Aziz, A.A.A., Kannan, A.: Access control for healthcare data using extended XACML-SRBAC model. In: 2012 International Conference on Computer Communication and Informatics. pp. 1–4. Dept. of Information Science & Technology, Anna University, IEEE (Jan 2012)
12. Faye, Y., Niang, I., Noel, T.: A survey of access control schemes in wireless sensor networks. Proc. World Acad. Sci. Eng. Tech (Laboratory LID), 814–823 (2011)
13. 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 (Oct 2011)

14. Lascelles, F., Flint, A.: WS-Security Performance (2006), http://websphere.sys-con.com/node/204424
15. Lawrence, K., Kaler, C., Nadalin, A., Monzilo, R., Hallam-Baker, P.: Web Services Security: SOAP Message Security 1.1 (2006), http://docs.oasis-open.org/wss/v1.1/
16. Maerien, J., Michiels, S., Huygens, C., Hughes, D., Joosen, W.: Access Control in Multi-party Wireless Sensor Networks. In: Demeester, P., Moerman, I., Terzis, A. (eds.) Wireless Sensor Networks SE - 3, Lecture Notes in Computer Science, vol. 7772, pp. 34–49. Springer Berlin Heidelberg (2013)
17. Nixon, T., Regnier, A., Jeyaraman, R.: SOAP-over-UDP Version 1.1 (2009), http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/
18. Parducci, B., Lockhart, H., Rissanen, E.: eXtensible Access Control Markup Language (XACML) Version 3.0 (2003), http://docs.oasis-open.org/xacml/3.0/
19. Rantos, K., Papanikolaou, A., Fysarakis, K., Manifavas, C.: Secure policy-based management solutions in heterogeneous embedded systems networks. In: 2012 International Conference on Telecommunications and Multimedia (TEMU). pp. 227–232. IEEE (Jul 2012)
20. Rantos, K., Papanikolaou, A., Manifavas, C., Papaefstathiou, I.: Ipv6 security for low power and lossy networks. In: Wireless Days (WD), 2013 IFIP. pp. 1–8 (Nov 2013)
21. 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. pp. 59–64. MobiWac '13, ACM, New York, NY, USA (2013)
22. Ray, P., Wimalasiri, J.: The need for technical solutions for maintaining the privacy of EHR. In: Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society. vol. 1, pp. 4686–4689. IEEE (2006)
23. 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 (Jun 2011)
24. Rescorla, E., Modadugu, N.: Datagram Transport Layer Security (2012), http://tools.ietf.org/rfc/rfc6347.txt
25. 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/
26. Smith, R.: SPOTWorld and the Sun SPOT. 2007 6th International Symposium on Information Processing in Sensor Networks (2007)
27. 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 (2001), http://www.ietf.org/rfc/rfc3198.txt
28. Whiting, D., Housley, R., Ferguson, N.: Counter with CBC-MAC (CCM) (2003), http://tools.ietf.org/rfc/rfc3610.txt
29. Yang, G., Yacoub, M.: Body sensor networks, vol. 6. Springer London (2006)
30. 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)
31. 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 (Sep 2009)