Extending a user access control proposal for wireless network services with hierarchical user credentials

J.A.M. Naranjo[1], Aitor Gómez-Goiri[2], Pablo Orduña[2], Diego López-de-Ipiña[2], and L.G. Casado[1]

[1] Dpt. of Computer Science,
University of Almería,
Agrifood Campus of International Excellence (ceiA3), Spain
{jmn843, leo}@ual.es

[2] Deusto Institute of Technology - DeustoTech,
University of Deusto, Spain
{aitor.gomez, pablo.orduna, dipina}@deusto.es

Abstract. We extend a previous access control solution for wireless network services with group-based authorization and encryption capabilities. Both the basic solution and this novel extension focus on minimizing computation, energy, storage and communications required at sensors so they can be run in very constrained hardware, since the computations involved rely on symmetric cryptography and key derivation functions. Furthermore, no additional messages between users and sensors are needed. Access control is based on user identity, group membership and time intervals.

Keywords: access control, group-based authorization, wireless network services, Internet of Things, ubiquitous computing

1 Introduction

The Internet of Things (IoT) initiative advocates for providing identities to everyday objects by representing them on the Internet. A way to achieve that is to physically connect them to the Internet so the objects can interact with Internet services and vice-versa. Together with mobile computing, IoT constitutes the clearest sign of the Ubiquitous Computing prominence in our current lives [1]. On the other hand, security has been an ever-present concern in Internet communications, and will keep being in the new scenario: if we want the IoT paradigm to reach all its possibilities then we need to provide reliable routines for information encryption and user authentication and authorization. Furthermore, these routines must be able to run seamlessly in very constrained hardware: small and cheap devices with limited processing capabilities and sometimes energy restrictions. For example, a typical mote in a wireless sensor network is not able to make use of public key cryptography (on a frequent manner at least) given the high computational and energy demands of the latter. Hence, very lightweight security routines are needed. In [3, 4] we presented an access control solution for wireless environments in which users access services offered by constrained devices (e.g. wireless sensors). This solution provides efficient encryption, authentication and authorization on a per-user basis, i.e. a given user can access the services offered by a given sensor based on her identity. Furthermore, it needs no additional messages in the user-sensor communication. In this work, we extend the solution in order to differentiate groups of credentials in the authorization process, i.e. users can access the services offered by a group of sensors when they have the corresponding group credentials. The groups can be either hierarchical or non-hierarchical. In the latter, members in different privilege groups enjoy different non-hierarchical sets of services. In the former, members in higher privilege groups enjoy more services than lower level users. We present the basic protocol, the novel group credentials extension, and a discussion in terms of security and both message overhead and storage requirements. Experimental results in [4] confirm the applicability of our proposal.

The article is organized as follows. Section 2 discusses some proposals from the literature. Sections 3 and 4 present the scenario we are addressing here and recall the basic protocol, respectively. Section 5 introduces the novel groups extension, while Sections 6 and 7 discuss it in terms of security, message overhead, storage and efficiency. Section 8 concludes the article.

2 Related work

The popular SPINS solution [6] provides lightweight symmetric encryption and authentication in wireless sensor networks where a Base Station is actively involved. It is composed of two sub-protocols: SNEP, which provides encryption, authentication and data freshness evidence between two parties, and μTESLA, used for authenticating broadcast messages to the whole network. LEAP+ [8] proposes an authentication and encryption framework for similar scenarios. Apart from its own protocols, μTESLA is used for authentication of broadcast messages from the Base Station. Ngo et al [5] proposed an access control system for the scenario we address here: wireless networks that provide services to users supported by an Authorization Service. It allows both individual and group-based authentication thanks to the combination of user keys and group keys. The recent MAACE [2] also focuses on the same scenario with individual and per-group authentication. However its storage requirements at every sensor are very large (sensors must store all keys shared with online users at a given time). The authors solve the storage problem by involving the Base Station in frequent communications, which is not a proper solution from our point of view since sending information is by far the most energy-consuming operation for sensors.

3 Scenario

The scenario we address in this work involves three kinds of players: sensors, Base Stations and user devices (e.g. smartphones), interacting together in a given facility (buildings, factories, greenhouses, homes, etc).

Sensors are extremely constrained wireless devices, frequently battery-powered and with reduced computational capabilities, which provide users with services of any kind. Their reduced equipment and power supply prevents them from carrying out the complex arithmetic operations involved in public-key cryptography.

However, symmetric cryptography is an option, either in software or hardware since many sensor models include an AES coprocessor. Note that under this category we also consider actuators, which are devices able to perform actions related to physical access control (opening gates to authorized users), ventilation, emergencies, etc.

Base Stations are better equipped devices that handle groups of sensors for message routing purposes, data collection and also for key management in our case. They are assumed to have a more powerful hardware and a permanent (or at least much larger) power supply and large storage space. They are also assumed to handle public-key cryptography routines and certificates.

Finally, users communicate with Base Stations and sensors through their powerful smart devices, such as mobile phones or tablets.

The key point here is that sensors need to perform access control on users, however they have to face several limitations: 1) they are not able to handle complex public-key authentication nor encryption routines and 2) they do not have enough memory space so as to keep large sets of user keys. The goal of our basic protocol is to provide an access control mechanism with symmetric encryption and authentication routines which minimizes storage requirements. On the other hand, the goal of the groups extension introduced in this work is to manage users on a per-group basis: each user group has a different set of privileges, meaning that they can access different sets of the services provided by the sensors. Table 1 shows the notation used throughout the article.

MS_S	Master secret for sensor S
$Kenc_{S,A},\ Kauth_{S,A}$	Encryption and authentication keys for communication between sensor S and user A
$Kenc_{S,A}\{x,\ ctr\}$	x is encrypted in counter mode using key $Kenc_{S,A}$ and counter ctr
$MAC_{Kauth_{S,A}}(x)$	A MAC is done on x using $Kauth_{S,A}$
$KDF(x,\ \{a,\ b\})$	A Key Derivation Function is applied to master secret x using a as public salt and b as user-related information
$H(x)$	A hash function is applied to x
$x\|\|y$	Concatenation of x and y
ID_A	Identifier of user A
a	Random integer salt
$init\_time,\ exp\_time$	Absolute initial and expiration time of a given key
MS_p	Master secret for privilege group p
$Kenc_{p,A},\ Kauth_{p,A}$	Encryption and authentication keys between sensors offering services for group p and user A
ID_p	Identifier of privilege group p
$A \rightarrow *$	User A sends a message to any listening sensor
$S_p \rightarrow A$	One sensor giving services from privilege group p sends a message to A

Table 1. Notation

4 The basic protocol

Here we briefly summarize the initial version of the protocol as showed in [3, 4]. It provides encryption and user access control to user ↔ sensor one-to-one communications. The Base Station, a more powerful device, performs high-level authentication on the user (with authorization certificates based in public key cryptography, for example) and provides her with two symmetric keys (for encryption and authentication, respectively) and parameters for their generation at the sensor. If those parameters are attached to the first message of a conversation then the sensor can input them to a Key Derivation Function in order to obtain an identical pair of symmetric keys that make communication possible. Figure 1 depicts the message exchange in the protocol. Let us explain it with more detail.

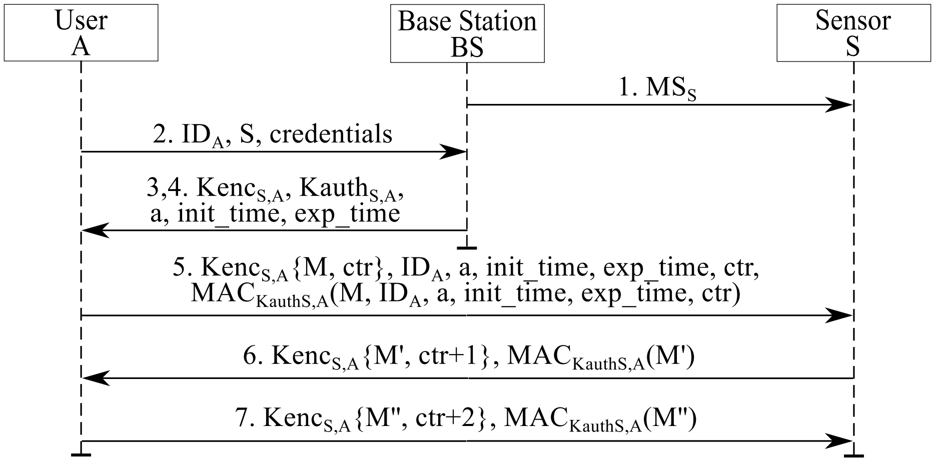

User A
Base Station BS
Sensor S
1. MS_S
2. ID_A, S, credentials
3,4. $Kenc_{S,A}$, $Kauth_{S,A}$, a, init_time, exp_time
5. $Kenc_{S,A}\{M, ctr\}$, ID_A, a, init_time, exp_time, ctr, $MAC_{KauthS,A}(M, ID_A, a, init\_time, exp\_time, ctr)$
6. $Kenc_{S,A}\{M', ctr+1\}$, $MAC_{KauthS,A}(M')$
7. $Kenc_{S,A}\{M'', ctr+2\}$, $MAC_{KauthS,A}(M'')$

Fig. 1. Messages involved in the original protocol

1. At the time of sensor deployment, the latter receives a master secret MS_S, which is *secretly shared* (see Section 6) by the Base Station BS and the sensor S. This step is run only once in the life of the sensor (unless the master secret needs to be changed).
2. Upon arrival, user A sends her credentials (e.g. an authorization certificate) to BS so high-level access control can be performed, and the list of sensors she wants to communicate with (in Fig. 1 we only consider S). This step is run only at user arrival.
3. BS computes:
   (a) a, random integer salt
   (b) $(init\_time,\ exp\_time)$, keying material validity interval
   (c) $Kenc_{S,A},\ Kauth_{S,A} = KDF(MS_S, \{a,\ ID_A || init\_time || exp\_time\})$

4. BS sends the information generated in the previous step to A under a secure channel (see Section 6).
5. A encrypts her first message to S with $Kenc_{S,A}$ in counter mode (thus using a fresh counter ctr), attaches parameters ID_A, a, $init\_time$, exp_time, ctr in plain text and a MAC obtained with $Kauth_{S,A}$.
6. Upon reception of the message, S obtains the key pair $Kenc_{S,A}$, $Kauth_{S,A}$ by feeding the Key Derivation Function with the attached parameters; S can now decrypt the message. The reply is encrypted in counter mode with $Kenc_{S,A}$ and $ctr + 1$ and authenticated with a MAC using $Kauth_{S,A}$.
7. Any subsequent message is encrypted and authenticated with the same key pair after increasing the counter by one.

When the message exchange finishes the sensor deletes all information related to the user since it can be easily and quickly recomputed at the beginning of the next exchange, thus saving space at the sensor. The sensor is sure of the authenticity of the user since the only way of knowing ($Kenc_{S,A}$, $Kauth_{S,A}$) is either knowing MS_S (which is kept secret) or obtaining it from the Base Station (which is actually the case). What is more, the MAC at the end of the message provides integrity assurance in addition to authentication. We refer the reader to [3, 4] for more considerations on security, efficiency, message overhead and storage.

5 A groups extension

In this section we address a scenario with different groups of users, each group giving its members access privilege to a given set of services provided by sensors. Services provided by a sensor may (but not necessarily) belong to more than one group. The associated access control routines should not be intensive in terms of computations or message exchanges.

Let us assume that there are $l > 0$ groups. The main idea is that there exists a different master secret MS_p for every privilege group $p \in [1,\ l]$, hence sensors should only reply to service requests encrypted and/or authenticated with a key pair derived from the corresponding master secret. From here, we propose two different approaches based on how services are arranged into groups. In Approach 1 privilege groups are not hierarchical, like in the case of employees that are allowed to enter different areas of a facility based on their activity (though some services might be in more than one group). In Approach 2 privilege groups are hierarchical, hence a user with privilege level p should enjoy all privileges from groups $[1,\ p]$. An example of this scenario is a smart house with different privilege groups based on age: children would have access to certain services of the house, while parents should have full control of the house.

5.1 Approach 1: non-hierarchical privilege groups

In this case, the Base Station generates l independent random master secrets $MS_1, \ldots, MS_l$ assuming there exist l different privilege groups. Sensors offering services from any privilege group p receive MS_p from the Base Station under a secure channel. In this scenario, users will typically belong to one group only, and sensors will provide services to one group as well. Figure 2(a) shows an example with three users and three sensors. However, if a sensor offers services to different privilege groups (or if a given service is included in more than one group), then the sensor should store each group's master secret. In a similar way, users assigned to more than one group (if that occurred) should receive a different pair of keys per group, and use the appropriate one to the requested service.

When user A arrives at the system the Base Station authenticates her and generates a different pair of symmetric keys ($Kenc_{p,A},\ Kauth_{p,A}$) for the privilege group A belongs to (group p in this case). These keys are generated by the BS and sensors assigned to group p in the same way as in the basic protocol: the user identifier, a random salt a and a key validity interval ($init\_time,\ exp\_time$) are fed to a Key Derivation Function along with the corresponding master secret as shown in Eq. (1).

$$Kenc_{p,A},\ Kauth_{p,A} = KDF(MS_p, \{a,\ ID_A||init\_time||exp\_time\}) \quad (1)$$

These keys are sent to A by the BS under a secure channel (see Section 6). When user A wants to request a service from privilege group p she needs to encrypt and authenticate her message with that pair of keys like in the basic protocol (note that ID_p has been added).

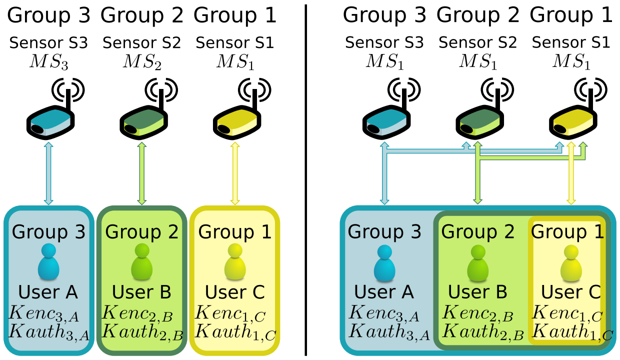
$$\begin{aligned} A \rightarrow * : [&Kenc_{p,A}\{M,\ ctr\},\ ID_A,\ ID_p,\ a,\ init\_time,\ exp\_time,\ ctr, \\ &MAC_{Kauth_{p,A}}(M,\ ID_A,\ ID_p,\ a,\ init\_time,\ exp\_time,\ ctr)] \quad (2) \end{aligned}$$

Any nearby sensor providing services from group p (let us name it S_p) can now reply to A after deriving the appropriate pair of keys from the received information and MS_p. The counter is explicitly stated on plain text so synchronization is not lost due to an arbitrary sequence of messages if more than one sensor is involved in the conversation.

$$S_p \rightarrow A : [Kenc_{p,A}\{M',\ ctr+1\},\ ctr+1,\ MAC_{Kauth_{p,A}}(M',\ ctr+1)] \quad (3)$$

5.2 Approach 2: hierarchical privilege groups

In this case, services are arranged in hierarchical groups: users assigned to privilege group p should be granted access to all services in groups $[1,\ p]$. Here every sensor in the system receives the lowest group's level master secret MS_1 from the BS. The rest are obtained by hashing the immediately lower master secret, i.e. $MS_p = H(MS_{p-1})$. This requires lower permanent storage requirements at the cost of a slightly higher computational demand and more security risks as



(a) Approach 1 (b) Approach 2

Fig. 2. Examples of the two approaches with three groups.

we will see later. Note that every sensor can obtain the master secret for any privilege level. Figure 2(b) shows an example with three users and three sensors.

Thanks to this modification, user devices need to store only one pair of keys, that of the highest privilege level they are granted. For example, a user A in group 3 will only receive $(Kenc_{3,A},\ Kauth_{3,A})$ from the Base Station. However the use of this key pair is enough for being granted access to any service in groups 1 to 3.

The verification of user credentials at the sensor side goes as follows. After receiving a message encrypted and authenticated with $(Kenc_{p,A},\ Kauth_{p,A})$ (see Eq. (2)) the sensor derives $MS_p = H(...H(MS_1))$. From MS_p and user-bound parameters the sensor obtains $(Kenc_{p,A},\ Kauth_{p,A})$ as in Eq. (1). Communications can now be established as in Eq. (3).

5.3 Combining hierarchical authentication with individual privacy

The basic protocol provides one-to-one authentication and encryption between a user and a sensor. On the other hand, approaches 1 and 2 allow to perform one-to-many authentication and encryption: all sensors holding the affected master secret will be able to authenticate the user and decrypt the conversation. Next, we consider the possibility of having services that demand one-to-one private communications and group-based authorization at the same time. For achieving this we we base on Approach 1, however the extension to Approach 2 is straightforward.

In this case sensor S is assigned by the Base Station an individual master secret MS_S (as in the basic protocol) and one master secret MS_p for each privilege group p the sensor provides services from (in Approach 2 the sensor would be assigned MS_1 and would derive the rest by hashing).

User A is assigned a pair of keys for individual communication with S, i.e. $(Kenc_{S,A},\ Kauth_{S,A})$, and a pair of keys $(Kenc_{p,A},\ Kauth_{p,A})$ for the privilege group she is entitled to, say p. Like before, these keys are generated for A by the Base Station by feeding MS_p and user-related parameters $ID_A,\ a,\ init\_time,\ exp\_time$ to a Key Derivation Function.

Now, when A wants to communicate only with S while proving her authorization level, she encrypts her messages with $Kenc_{S,A}$ and computes the corresponding MAC with $Kauth_{p,A}$ as in Eq. (4). S replies using the same pair of keys and incrementing the counter, which needs not to be included on plain text given that the message exchange takes place between two players only:

$$\begin{aligned} A \rightarrow S : [&Kenc_{S,A}\{M,\ ctr\},\ ID_A,\ ID_p,\ a,\ init\_time,\ exp\_time,\ ctr, \\ &MAC_{Kauth_{p,A}}(M,\ ID_A,\ ID_p,\ a,\ init\_time,\ exp\_time,\ ctr)] \quad (4) \\ S \rightarrow A : [&Kenc_{S,A}\{M,\ ctr+1\}, MAC_{Kauth_{p,A}}(M,\ ctr+1)] \quad (5) \end{aligned}$$

6 Considerations on security

Similar considerations to those made for [3, 4] can be made here. Both the basic protocol and the extensions provide semantic security (different encryptions of the same plain text produce different ciphertexts) thanks to the use of counter mode encryption. At this point we remark that it is the user who chooses the initial counter to be used during each message transaction. If the sensor does not trust the user then she can choose a new counter in her reply (Eq. (3) or (5), in the latter the counter should be hence added on plain text), thus forcing the user to increment that new one.

Regarding how to install master secrets on sensors, it can be done if a symmetric key is pre-installed at every sensor at deployment time. This key should be different for every sensor and shared with the Base Station, thus obtaining private communications with the latter. Furthermore, master secrets can be updated at a given frequency to enhance security, but once a master secret is updated the symmetric pairs of keys generated from the old version will be no longer valid in the system. To solve this problem, the *exp_time* value associated to every key pair can be made to match the master secret's update time, thus forcing the user to obtain a new pair from the Base Station. Doing so, the new pair will be derived from the new master secret.

Regarding how to communicate the user-related key pairs to the user, we assume that both user devices and the Base Station can handle public key encryption, hence a temporary secure channel is easy to establish (e.g. by using public key certificates).

Let us conclude this section with a discussion on key compromise. Given that user key pairs are bound to a specific user, stealing them will allow to impersonate a single user only, thus limiting the impact of a security breach at the user side. At the sensor side we can differentiate between stealing a sensor-only master secret or a privilege group master secret. In the first case, an attacker that steals MS_S from a sensor will only be able to impersonate that given sensor. In the second case, we can distinguish between approaches 1 and 2. In Approach

1 a sensor receives only the master secrets it is entitled to. Stealing a master secret MS_p will allow an attacker to understand and forge messages related to privilege group p, thus impersonating any sensor within that group, but not within other groups. In Approach 2, a sensor can obtain the master secret from any privilege group from the lowest group's. Thus, compromising a single sensor would allow an attacker to impersonate any sensor at any privilege group. The conclusion is that Approach 1 offers more security at the cost of more permanent storage requirements.

7 Other considerations

The most power-demanding operation in a sensor is airing messages through its antenna [6, 8], hence protocols intended for sensors (and not only those related to security) should try to minimize the number of messages needed as well as their length. Our protocol and its extensions do not require any additional message in a service request from the user to the sensor. Regarding message length, the additional overhead is values $(ID_p,\ a,\ init\_time,\ exp\_time, ctr)$ in the first message and a MAC (we assume ID_A must be sent anyway). The sensor needs only to attach the counter on plain text in approaches 1 and 2 (and in Section 5.3 in the case the user's counter is discarded and a new one is used).

Speaking of storage, the basic protocol requires that the user stores a pair of keys $(Kenc_{S,A},\ Kauth_{S,A})$ per sensor S and values ID_p, a, $init\_time$, exp_time (again we consider ID_A is needed anyway). The counter ctr must also be stored during a message exchange. The sensor needs only to permanently store a symmetric session key for communications with the Base Station and MS_S. During a message exchange, the sensor needs to keep the pair of keys used for that user and values $(ID_p,\ a,\ init\_time,\ exp\_time,\ ctr)$. That information may be erased after the exchange since it is easily recomputed each time needed.

In addition to the storage requirements of the basic protocol, approaches 1 and 2 require the user to store permanently a pair of keys for the group. At the sensor side, Approach 1 requires the sensor to store a master secret for every privilege group it might be assigned to. In Approach 2, however, the sensor can decide whether to permanently store a single master secret (that of the lowest level, thus needing to compute the needed master secret at every message exchange) or to store all master secrets once derived (thus saving computations at the cost of space). We see this tradeoff as an interesting open future workline.

Passive participation is a typical behaviour used by sensors in order to save energy based on overheard messages [8]: if a node receives a user query and a subsequent reply from a different sensor then the first node can decide to not to reply in order to save the energy spent in transmission. Approach 1 allows for this type of behaviour within a given group, while Approach 2 allows for it within the group and those below in the hierarchy.

Regarding efficiency in computations, the experimental results shown in [4] for the basic protocol are equally valid for the groups extension, since the latter adds no overhead apart from the inclusion of the group identifier in MACs.

8 Conclusions

Here we present a group-based extension for an access control protocol for wireless network services. We address infrastructures populated by constrained devices (such as wireless sensors) that are arranged in different groups of services: users are granted access to these groups depending on their privileges. We consider two different scenarios, depending on whether privilege groups are hierarchical (entitlement to a privilege group implies access to all services down to the lowest group) or not (users can only access services contained in the very privilege group they are entitled to). Also, we show a way of combining individual encryption with group-based authorization. Regardless of the approach chosen, the authentication and authorization processes are performed efficiently and with no additional messages between the user and the addressed sensor. We discuss our proposal in terms of security and message and storage overhead. Also, experimental results shown in previous work [3, 4] prove its applicability. Future worklines include the formal validation of the protocol in AVISPA [7] and its implementation on an extremely constrained platform such as MICAz or Arduino.

Acknowledgements

This work was funded by the Spanish Ministry of Economy and Competitiveness (TIN2008-01117, IPT-2011-1558-430000 and TIN2012-37483-C03-03), Junta de Andalucía (P11-TIC-7176), and partially by the European Regional Development Fund (ERDF).

References

1. Aitor Gómez-Goiri, Pablo Orduña, Javier Diego, and Diego López-de-Ipiña. Otsopack: Lightweight Semantic Framework for Interoperable Ambient Intelligence Applications. To appear in Journal of Computers in Human Behavior.
2. Le, X.H., Khalid, M., Sankar, R. and Lee S. "An Efficient Mutual Authentication and Access Control Scheme for Wireless Sensor Networks in Healthcare". Journal of Networks (2011) 6:3 pp: 355-364.
3. Naranjo, J. A. M. and Orduña, Pablo and Gómez-Goiri, Aitor and López-de-Ipiña, Diego and Casado, L. G. Lightweight User Access Control in Energy-Constrained Wireless Network Services. In Ubiquitous Computing and Ambient Intelligence. Lecture Notes in Computer Science, volume 7656, pp: 33–41. Springer, 2012.
4. J.A.M. Naranjo, Pablo Orduña, Aitor Gómez-Goiri, Diego López-de-Ipiña and L.G. Casado. Enabling user access control in energy-constrained wireless smart environments. To appear in Journal of Universal Computer Science.
5. Ngo, H.H., Xianping W., Phu D.L. and Srinivasan, B. "An Individual and Group Authentication Model for Wireless Network Services". JCIT (2010) 5:1, pp: 82-94.
6. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V. and Culler, D. E. "SPINS: security protocols for sensor networks". Wireless Networks (2002) 8:5, pp: 521-534.
7. The AVISPA Project. http://www.avispa-project.org/
8. Zhu, S., Setia, S. and Jajodia, S. "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks". ACM Transactions on Sensor Networks, (2006) 2:4, pp: 500-528.