Lightweight user access control in energy-constrained wireless network services

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

[1] Department of Computer Architecture and Electronics
University of Almería, Spain
{jmn843,leo}@ual.es
[2] Deusto Institute of Technology - DeustoTech
University of Deusto, Bilbao, Spain
{pablo.orduna,aitor.gomez,dipina}@deusto.es

Abstract. This work introduces a novel access control solution for infrastructures composed of highly constrained devices which provide users with services. Low energy consumption is a key point in this kind of scenarios given that devices usually run on batteries and are unattended for long periods of time. Our proposal achieves privacy, authentication, semantic security, low energy and computational demand and device compromise impact limitation on a simple manner. The access control provided is based on user identity and time intervals. We discuss these properties and compare our proposal to previous related work.

Keywords: access control, wireless network services, internet of things, sustainability

1 Introduction

The paradigm known as the Internet of Things (IoT) defends the benefits of everyday objects becoming first-class citizens of the Internet. To do so, these objects must be provided with connectivity to expose and to consume data from any other applications or services. Nevertheless, the embedded devices used to connect the objects face different problems and challenges compared to normal computers. Due to the large scale of objects that will populate the IoT, these devices are usually designed to be small and inexpensive, resulting in limited processing capability. Additionally, these devices are often running 24 hours a day so low power consumption is required to enable sustainable computing.

Security-related routines sometimes impose an increment of energy consumption due to expensive calculations. Indeed, little attention has been paid to the security aspects in the IoT, and commonly security is left as a *dispensable*, energy draining process. The focus of this contribution is to define a novel and low consumption solution for restricting access to legal users and securing communications among them and constrained devices. The solution enables a trustworthy communication on an insecure network at the cost of reduced energy consumption. The proposed model is compared with other solutions from the literature, showing how it is an advance in the state of the art on the field of efficient security with the restrictions commonly present in the IoT.

The contributions of this paper are twofold. First, a model is proposed to cover a typical scenario, adding the required access control layer, and second a review of the state-of-the-art and a comparison of this solution with different existing solutions to the date is provided. Section 2 provides some cryptographic background, Section 3 details the scenario we are focusing, while Sections 4 and 5 introduce our proposal and compare it with previous works. Section 6 concludes the paper.

2 Background

A *Message Authentication Code* (MAC) takes as input a message and a symmetric key shared by two or more communicating parties to produce an unequivocal bit string bound to the message and the key. A popular standard is HMAC [1].

Key Derivation Functions (KDFs) are used to derive proper symmetric cryptographic keys from a non-cryptographically strong secret input and additional (possibly) public information. The two existing general-purpose standards, NIST SP 800-108 [2] and HKDF [3], allow (and recommend) to include a public random value (namely the *salt*) and additional application-specific information in the key derivation process, such as the identity of the future key holder or a timestamp. A large number of privacy-related solutions and protocols rely on KDFs, among them [4, 5].

MAC and KDF computations are extremely efficient since their core routine is a hash function based on simple bit permutations. Special care should be taken when choosing the standards for an implementation of the work shown hereafter. We leave that question open in this work.

3 Assumed infrastructure and cases of use

The infrastructure considered in our solution involves three kinds of players: sensors, Base Stations and user devices (e.g. smartphones). Sensors are extremely constrained devices, frequently battery-powered and with reduced computational capabilities. Equipment and power shortage prevents sensors from performing on a frequent basis the complex arithmetic operations involved in public-key cryptography in order to achieve encryption and authentication. However, symmetric cryptography is an option, specially given that many 802.15.4/ZigBee compliant sensors have an Advanced Encryption Standard (AES) coprocessor installed.

Base Stations are better equipped devices that handle groups of sensors for message routing purposes and also for key management in our case. They are assumed to have a more powerful hardware, a permanent (or at least much longer) 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 smart devices, such as mobile phones or tablets.

In order to illustrate the scenario we are focusing let us assume a home automation infrastructure. A set of sensors is deployed within a house, e.g. lights control, alarm or TV. Different users of the house may enjoy different access privileges and therefore can also be separated into different groups (e.g. adult

$MSenc_S$, $MSauth_S$	Encryption and authentication master secrets for sensor S
$Kenc_{S,A}$, $Kauth_{S,A}$	Encryption and authentication keys for communication between sensor S and user A
$Kenc_{S,A}\{x\}$	x is encrypted with $Kenc_{S,A}$
$MAC_{Kauth_{S,A}}(x)$	A MAC is done on x with $Kauth_{S,A}$
$KDF(x, \{a, b\})$	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

Table 1. Notation

owners, children, friends or relatives). Each group has a different set of permissions. For example, adult owners should have the highest privilege to access and control every single sensor/actuator; children may have access to the TV actuators, but won't be able to purchase pay-per-view programs; and friends will be able to access the WIFI and turn on and off some lights of the house. These permissions are issued by members of the adult owners group.

4 Our proposal

Our main goal is to allow sensors and legal user devices only to establish encrypted and authenticated one-to-one channels while minimizing the intervention of the Base Station. The process should require a small amount of energy consumption and storage, specially on the sensor side. Minimizing storage requirements also implies that communications should be as stateless as possible, i.e., no inter-session information should be stored for long periods of time. Besides, it should be easy for the sensor to perform access control operations on user devices.

Our solution covers four phases: sensor bootstrapping, user join, regular communication and user eviction, all of which are described next. For the sake of simplicity and without loss of generality we focus on a simple scenario: one Base Station (namely BS), one sensor (namely S), and one user device (namely A). The extension of the proposed protocol to several users, sensors and base stations is straightforward from the protocol description below. Messages involved in the protocol are depicted in Figure 1 while Table 1 shows the notation used from now on.

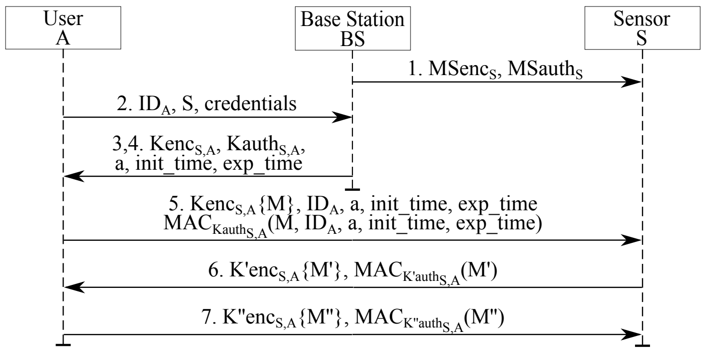
User A
Base Station BS
Sensor S
1. $MSenc_S$, $MSauth_S$
2. ID_A, S, credentials
3,4. $Kenc_{S,A}$, $Kauth_{S,A}$, a, init_time, exp_time
5. $Kenc_{S,A}\{M\}$, ID_A, a, init_time, exp_time $MAC_{Kauth_{S,A}}(M, ID_A, a, init\_time, exp\_time)$
6. $K'enc_{S,A}\{M'\}$, $MAC_{K'auth_{S,A}}(M')$
7. $K''enc_{S,A}\{M''\}$, $MAC_{K''auth_{S,A}}(M'')$

Fig. 1. Messages involved in the protocol

Sensor bootstrap. At the time of adding the new sensor S, the BS generates two master secrets: one for encryption and one for authentication, $MSenc_S$ and $MSauth_S$ respectively. These secrets are sent to S under a secure channel. See [6] for a good survey on the latter and [5] for a particularly smart solution.

1. $BS \to S$ (secure channel) : $[MSenc_S,\ MSauth_S]$

User join. Let us assume that S is already operating under normal conditions. User A arrives at the scenario handling her mobile device and wishes to request some information from S. First, A sends a request to BS asking for keying material to communicate with S (step 2). The message should include authentication and authorization information so BS can perform high-level access control on user A. For this we suggest the use of public key cryptography [7] given that *(i)* both the Base Station and the user device are assumed to handle it easily and *(ii)* it also allows to create a secure channel between them. In any case, let us remark that this step is only performed at user arrival and that many target sensors can be requested in the same message at step 2.

2. $A \to BS :\ [ID_A,\ S,\ credentials]$

If A's request is accepted then the Base Station generates appropriate keying material (step 3 can be repeated as many times as target sensors were requested) and sends it to A through a secure channel (step 4). The expiration time of this material is decided by the Base Station and cannot be changed by A.

3. BS computes:
   (a) a, random integer salt[1]
   (b) $(init\_time,\ exp\_time)$, keying material validity interval
   (c) $Kenc_{S,A} = KDF(MSenc_S, \{a,\ ID_A || init\_time || exp\_time\})$
   (d) $Kauth_{S,A} = KDF(MSauth_S, \{a,\ ID_A || init\_time || exp\_time\})$
4. $BS \to A$ (secure channel) : $[Kenc_{S,A},\ Kauth_{S,A},\ a,\ init\_time,\ exp\_time]$

Regular communication. A can now use the received keying material to encrypt and authenticate her first message M addressed to S (step 5).

5. $A \to S : [Kenc_{S,A}\{M\},\ ID_A,\ a,\ init\_time,\ exp\_time,$
   $MAC_{Kauth_{S,A}}(M, ID_A,\ a,\ exp\_time)]$

Upon reception of the message, sensor S computes the corresponding keying material as in steps 3c and 3d. S can now decrypt and authenticate the whole message. S's reply message, M', is encrypted with $K'enc_{S,A} = H(Kenc_{S,A})$ and authenticated with $K'auth_{S,A} = H(Kauth_{S,A})$ and needs not to contain any additional information (step 6).

6. $S \to A : [K'enc_{S,A}\{M'\},\ MAC_{K'auth_{S,A}}(M')]$

[1] We assume that $MSenc_S$, $MSauth_S$ and a are obtained from a secure pseudorandom number generator.

Subsequent messages within the same transaction are respectively encrypted and authenticated with $(K''enc_{S,A},\ K''auth_{S,A})$, $(K'''enc_{S,A},\ K'''auth_{S,A})$ and so on. If, for any reason, one of the players loses synchronization regarding which key to use for a given message it can always recover it by trying consecutive hashes until the proper key is found. This resynchronization process should not take long for short message exchanges. A very similar technique is used by the well known SNEP protocol [4].

So far we have considered the necessity of privacy between A and S. If the service provided by S does not require privacy then messages do not need to be encrypted, just authenticated with $Kauth_{S,A}$ (this is applicable to the rest of the paper). In any case, when the message exchange finishes the sensor can delete the keying material related to A since it can be easily recomputed in the next exchange. The protocol therefore does not require S to store any inter-session information.

User eviction. The inclusion of a validity time interval in the key derivation function input provides easy time-based access control. Before computing the keying material after step 5 the sensor S checks whether the expiration time has not yet been reached. This needs a very relaxed time synchronization with the BS, in the order of seconds, while other well known protocols impose much stronger requirements on this matter (centiseconds or milliseconds) [4, 8]. Also note that A cannot fake her $(init\_time,\ exp\_time)$ pair because the keys derived by the sensor will be different and communication will be impossible. Consequently, the user is *forced* to be honest.

Finally, the Base Station may decide to evict A before her expiration time in certain situations, e.g. due to misbehaviour or key exposure. This is more problematic: the only way of making S reject messages from A before exp_time is to maintain a blacklist with $(ID_A,\ exp\_time)$ items in every sensor, which requires an additional communication per item between the BS and the sensor. However, the scarce storage space at the sensor will not allow for long blacklists. In any case, items could be removed as soon as their corresponding expiration times were surpassed: from that moment on sensor S will reject step 5 messages basing on the obsolete exp_time value sent in step 5 rather than in ID_A.

4.1 Considerations on security

No keys are publicly disclosed, nor they even travel encrypted in the user-sensor message exchange. Following good cryptography practice, different keys are used for encryption and authentication so the use of a single key for more than one task is avoided. The impact that a sensor compromise makes on the rest of the network is reduced since there are no shared keys among sensors nor among users: every sensor owns a different master secret pair, so an attack on that node would not provide any knowledge about other sensors in the network. In a similar way, each user knows only those keys shared with a given set of sensors and, what's more, those keys are exclusive for her. A compromise on them would only allow to impersonate that user. This is not the case of [9] (see Section 5).

So far, the protocol suffers from a weakness that can be easily solved. We refer to the fact that, within the same key validity interval, different message exchanges at different transactions between A and S will use the same key chains: the first message $A \rightarrow S$ (step 5) will always be encrypted/signed with $(Kenc_{S,A},\ Kauth_{S,A})$, the first message $S \rightarrow A$ will use $(K'enc_{S,A},\ K'auth_{S,A})$, and so on. Reusing keys like that is obviously not desirable so we propose a simple, painless solution. Before introducing it let us recall that we are searching for a stateless proposal which does not require the storage of session data on a per-user base within the sensor. Now, the solution relies on performing a random number of hash operations, say h, on $(Kenc_{S,A},\ Kauth_{S,A})$ at the beginning of each message exchange. Then the key chain used in the given exchange will start in $(K^{h}enc_{S,A},\ K^{h}auth_{S,A})$. The downsides are that h must be communicated in step 5 and that different key chains may overlap (e.g., [h, h+15] and [h-10, h+5]). To avoid the latter the user can choose ever-increasing, sufficiently scattered values for h.

Note that the user decides on the value h to use. If the sensor does not trust user devices by default then it can choose a different value on step 6, say h', and include it in the response message. The device should then use $h'+1$ in the next message and so on. In any case, the sensor can delete this information along with the user-specific keying material when the transaction ends. The main benefit we obtain from this solution is achieving *semantic security* without needing to negotiate on initial parameters like in [4] (see Section 5).

4.2 Considerations on overhead and storage

Very little overhead is added to message length in user-sensor communications: the user device just needs to send a and time-related information in step 5 and optionally h (step 5) for enhanced security (we assume that ID_A must be sent in any case). The additional information sent by the weakest player, i.e. the sensor, is minimal: only h' in step 6 if the security extension is applied, nothing otherwise. This desirable feature helps reduce the energy used by the sensor when transmitting (by far the most energy-consuming operation in sensors).

The permanent storage requirements imposed on the sensor are also extremely reduced. It only needs to store *(i)* a symmetric session key for communications with BS and *(ii)* $(MSenc_S,\ MSauth_S)$. While exchanging messages with user A the sensor stores *(i)* the last received message M, which includes ID_A, a, $init\_time$, and exp_time, *(ii)* the last h value used (optional) and *(iii)* $(K^{h}enc_{S,A},\ K^{h}auth_{S,A})$. Any available space left can be used for a user blacklist if desired. On the other hand, the fact that the user device must keep a pair of keys per sensor might be seen as a downside. However, current smartphones and similar devices have huge long-term memories in comparison to sensors.

5 Related work

SPINS [4] provides lightweight symmetric encryption and authentication in WSNET scenarios in which a Base Station is actively involved. It is composed of two different protocols: SNEP and μTESLA. The SNEP protocol provides encryption, authentication and data freshness evidence between two parties, using symmetric encryption in counter mode to ease freshness verification and to thwart replay attacks. On the other hand, the μTESLA protocol provides symmetric authentication: the sender builds a hash-based key chain and discloses keys with and intentional delay. Based on that delay and on the one-way property of hash functions, message recipients can verify the authenticity of messages.

LEAP+ [5] proposes an authentication and encryption framework for the same scenarios addressed by SPINS: wireless networks of sensors communicating among them and with the Base Station. Apart from its own protocols, μTESLA is used to provide broadcast communications by the Base Station. LEAP+ is proven to be lightweight and low demanding in terms of energy consumption. Its cornerstone operation is an interesting proposal for limiting the impact of a single node compromise based on pseudo-random functions.

Ngo et al [9] proposed an access control system for the scenario we address here: wireless networks that provide services to users. Either service and user groups are supported with the help of an Authorization Service. Two computationally lightweight protocols are described: while the first one allows to prove a user group membership to a service, the second protocol is intended to authenticate against a service both individually and per-group by employing the user's individual key and the group key. Note that a stolen group key would compromise the whole group.

The very recent MAACE [10] addresses a similar scenario and is very similar to our proposal according to the key generation process and the security achieved. User-Base Station communications are secured with public-key cryptography, while user-sensor messages are encrypted with a freshly established symmetric session key. However, two drawbacks can be found in this scheme. First, the session key shared with the sensor is chosen by the user, which is not a desirable feature (she might use a deliberately weak key). In our scheme, the BS takes care of this task. Second, the sensor must store all keys shared with online users at a given time (which requires a large storage space), or involve the trusted device in frequent communication establishments (which would make the process less efficient). In our scheme, the sensor can generate any valid key on the fly.

Table 2 compares our proposal to the reviewed related work according to some relevant features. All protocols shown provide encryption and authentication.

	SPINS [4]	**LEAP+** [5]	**Ngo** [9]	**MAACE** [10]	**Ours**
Number of keys per sensor	1 for BS	1 for BS 1 per neighbour 1 for cluster	1 for group	1 for BS	1 for BS 2 for basic 2 per group
Number of keys per user	-	-	2	1 for BS	2 per sensor
Tight clock synchronization	Yes	No	No	No	No
BS is highly involved	Yes	No	Yes	Tradeoff	No
Limits impact of sensor compromise	Yes	Yes	No	Yes	Yes
Services scenario	No	No	Yes	Yes	Yes

Table 2. Feature comparison. LEAP+ is considered without μTESLA

6 Conclusions

This work introduces a simple user access control solution for wireless network services in a typical infrastructure composed of Base Stations and sensors, with users interacting through their smart devices (e.g. mobile phones) in an IoT scenario. We focus on a minimal use of computation, energy and storage resources at the sensor so as to address constrained devices: key distribution and access control rely on extremely fast key derivation functions and, for the same reason, memory usage is reduced since keys are computed on the fly when needed. This way, adding security to an IoT scenario does not imply a high energy consumption which would disable sustainability. Our solution provides encryption, authentication, semantic security and access control based on user identities and time intervals without requiring tight clock synchronization among devices. Finally, the intervention of the Base Station in user - sensor communications is minimal, which is also a desirable feature. Regarding future work, a sample scenario will be deployed so as to provide energy, CPU and memory consumption measurements compared with a straightforward insecure solution. This would be particularly interesting for measuring the extra energy consumption required by the secure solution, and how it might be affordable in most IoT scenarios.

Acknowledgments. 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. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Proceedings of CRYPTO'96, Springer-Verlag (1996) 1–15
2. Chen, L.: Recommendation for key derivation using pseudorandom functions. NIST Special Publication 800-108 (2008)
3. Krawczyk, H.: Cryptographic extraction and key derivation: the HKDF scheme. In: Proceedings of CRYPTO'10, Springer-Verlag (2010) 631–648
4. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: SPINS: security protocols for sensor networks. Wirel. Netw. **8** (2002) 521–534
5. Zhu, S., Setia, S., Jajodia, S.: LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. ACM Trans. Sen. Netw. **2**(4) (2006) 500–528
6. Zhang, J., Varadharajan, V.: Wireless sensor network key management survey and taxonomy. Journal of Network and Computer Applications **33**(2) (2010) 63 – 75
7. Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. ITU-T recommendation X.509 (2005)
8. Chowdhury, A.R., Baras, J.S.: Energy-efficient source authentication for secure group communication with low-powered smart devices in hybrid wireless/satellite networks. EURASIP J. Wireless Comm. and Networking (2011)
9. Ngo, H.H., Wu, X., Le, P.D., Srinivasan, B.: An individual and group authentication model for wireless network services. JCIT **5**(1) (2010) 82–94
10. Le, X.H., Khalid, M., Sankar, R., Lee, S.: An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare. Journal of Networks **6**(3) (2011)