On securing IEEE 802.15.4 networks through a standard compliant framework

Savio Sciancalepore, Giuseppe Piro, Elvis Vogli, Gennaro Boggia, and Luigi Alfredo Grieco
Department of Electrical and Information Engineeering (DEI)
Politecnico di Bari, Italy
Email: {name.surname}@poliba.it

***Abstract*—The IEEE 802.15.4 standard is widely recognized as one of the most successful enabling technologies for short range low rate wireless communications. Although it provides all the details of both MAC and PHY layers of the protocol stack, the standard also explains specific methodologies to protect MAC packets through symmetric-key cryptography techniques under several security options. However, the orchestration of available security profiles and the management of key negotiation schemes is delegated to upper layers. In support of this feature, this work describes a standard compliant security framework aimed at proposing: (i) different kind of security architectures, (ii) an efficient mechanism for initializing a secure IEEE 802.15.4 domain, and (iii) a lightweight mechanism to negotiate link keys among devices. Experimental tests have been conducted to demonstrate the behavior of the proposed solution in real environments. Obtained results clearly show that the enabling of security features in constrained nodes requires additional computational efforts, which involve a not negligible growth of communication latencies. Such findings have to be carefully considered when designing and developing enhanced applications in future and secured Internet of Things (IoT) systems.**

***Keywords*—IoT, IEEE 802.15.4, security, key negotiation, experimental evaluation**

I. Introduction

The Internet of Things (IoT) paradigm refers to a system where different objects (i.e., sensors, machines, vehicles, smart phones, tablets, motes etc.) realize a capillary networking infrastructure connected to the Internet [1], thus offering the opportunity to develop many novel pervasive services in a number of application domains, like health care, smart city, energy management, military, environmental monitoring, industry-automation [2][3]. Such evident potentials have driven researchers, industries, and standardization bodies to define novel and efficient protocol stacks, which are more suitable for constrained devices [2][4][5].

Among of all the most important proposals, the IEEE 802.15.4 standard is widely recognized as one of the most successful enabling technologies for short range low rate wireless communications and provides all the details for both MAC and PHY layers [6]. More recently, the IEEE 802.15.4e specification introduced some amendments to the aforementioned standard, among which we can consider the Time-Slotted Channel Hopping (TSCH), i.e., a novel MAC protocol properly designed to better support multi-hop communications in industrial applications [7][8]. However, to actualize the IoT vision and easy plug and play operations of smart devices in IPv6 networks, if properly combined in a communication stack for Low-power and Lossy Networks (LLNs) based on IEEE 802.15.4 radios, the Internet Engineering Task Force (IETF) has recently proposed and standardized novel interesting protocols at different layers of the protocol stack (i.e., for taking into account several issues, like routing [9], compression [10], security [8], and so on).

The risks arising from a potential lack of security and privacy of the involved stakeholders could actually hinder the deployment of the IoT. To overcome such an issue, several solutions have been introduced in both IEEE and IETF contexts and in literature (Tab. I reports a summary of the most important proposals). First, the IEEE 802.15.4 specification allows the possibility to protect MAC packets by means of symmetric-key cryptography techniques, based on the AES-CCM* algorithm, with several security options. Nevertheless, it does not explain how to handle the initialization of a secure IEEE 802.15.4 domain, the generation and the exchange of keys, and the management of joining operations in a secure 802.15.4 network already configured in the past. While a practical solution devoted to the distribution of keys in 802.15.4 networks has been conceived within the ZigBee IP specification, a number of scientific publications have suggested, so far, to adapt well-known security solutions, already highly exploited in conventional IP networks, to the IoT domain, as well as specific implementation of Key Management Protocols (KMPs) more suitable for LLN domains. In addition, very valuable proposals have been formulated in various IETF working groups focusing on different layers of the protocol stack of a LLN. Unfortunately, at the time of this writing, such works are still in progress or they have not been yet translated in implementations on real testbeds, thus preventing the demonstration of their universal effectiveness.

To provide a significant breakthrough in this direction, we developed a simple and standard compliant framework supporting a number of security features in the IEEE 802.15.4 MAC (this work has been presented in its preliminary version in [11] and [12]). The code is open-source and freely available from http://telematics.poliba.it/security-iot. It covers: (i) the identification of potential security configurations that could be supported in an IoT domain, (ii) the definition of schemes enabling the data confidentiality and integrity protection of messages, (iii) the design of an efficient mechanism to configure and maintain a secured IoT domain, and (iv) the development of a lightweight KMP used by *smart* objects to negotiate link keys. Moreover, to demonstrate its concrete effectiveness, this novel proposal has been also implemented in the well known OpenWSN stack [13] and experimental tests

were carried out with the TelosB Platform [14] to evaluate its behavior in real environments. Obtained results show that the enabling of security features in constrained nodes requires additional computational efforts, which involve a not negligible growth of communication latencies.

The rest of the paper is structured as follows: the security framework proposed hereby is described in Sec. II; Sec. III illustrates the experimental evaluation of the presented solution and, finally, Sec. IV closes the paper and draws future works.

II. The conceived security framework

The framework described in this section has been properly designed to provide security features at the MAC layer of a LLN based on the IEEE 802.15.4 standard. In general, a LLN architecture could be extended over a large area, thus leading to multi-hop communication links among motes. However, the devised framework focuses the attention to a specific portion of the network, namely *secured domain*, where nodes are directly connected at the datalink layer. In line with the IEEE 802.15.4 specification, it is composed by a number of remote nodes directly connected to a specific coordinator. However, the developed approach can be easily extended also to multi-hop scenarios. In that case, the network is composed by multiple *secured domains*. Hence, starting from the one handled by the coordinator and progressing towards the farthest, the setup of the secured domain is locally executed.

A. Envisaged Secured configuration

Five different security configurations have been conceived to support a wide range of heterogeneous IoT networks, that are *Fully Secured*, *Unsecured*, *Partial Secured*, and *Hybrid Secured* networks. In a *Fully Secured* network, both confidentiality and data integrity are provided for all packets. In a *Unsecured* network, instead, no security service is supported. Only the message integrity is guaranteed in a *Partial Secured* network. An *Hybrid Secured* network protects unicast communications between devices supporting security capabilities. Finally, when the *Flexible Secured* option is used, a network may move from the *Fully Secured* configuration to the *Hybrid Secured* one in the case at least one node does not support security capabilities.

B. Initialization of a secured domain

When *Fully Secured*, *Flexible Secured*, or *Hybrid Secured* configurations are enabled, a secured domain is initiated through the execution of *Setting-up* and *Bootstrapping* phases. The former is used to store into the device all the secrets required to set up a secured domain. The latter, instead, is exploited to initialize the secured domain and to compute a key that will be adopted to protect broadcast messages at the MAC layer.

The Setting-up phase
The *Setting-up* phase consists in storing, within the device, parameters and initial secrets useful to set up the secured domain (this operation may be performed by the manufacturer or by the network administrator). They include:

- The *Master Key*, which is an initial secret shared among all the motes. It is not directly used to encrypt and decrypt messages, but it is exploited, together with other time-varying parameters (that can be unique in each secured domain and periodically updated during the time) to generate all the required key materials. The *Master Key* can be used to generate two different keys: the *Default Key*, adopted to protect *broadcast* messages (i.e., the beacon frame) and the *Link Key*, used to encrypt and authenticate *unicast* packets (i.e., those exchanged between only two specific nodes).
- The *GlobalSecurityLevelsTable*, that is used to store the minimum security level and the list of allowed security levels that must be adopted for each kind of MAC frame and for each security configuration.
- The *PrimeNumbersTable*, which stores a set of N prime numbers and their respective primitive roots used during the *Key Negotiation* phase to generate *Link Keys* according to the DH algorithm [32].

Bootstrap phase for the coordinator
As soon as the device becomes the coordinator of a given portion of the LLN, it generates the *Default Key* and updates, accordingly, security-related parameters at the MAC layer.

In particular, the *Default Key*, D_k, is generated starting from the *Master Key* (i.e., M_k), the coordinator MAC address (i.e., $MAC_{addr,c}$), and the network ID (i.e., PAN_{ID}), by using a 128-bit hash function ($H_{128}\{\cdot\}$):

$$D_k = H_{128}\{PAN_{ID}|MAC_{addr,c}|M_k\}. \quad (1)$$

Note that the security level may be improved by adopting a keyed-hash message authentication code (HMAC) scheme instead of a simple hash function. In this case, the secret cryptographic key used to calculate the message authentication code is the Master Key, M_k.

Bootstrap phase for the remote mote
To join the network, the remote device should associate itself with the coordinator. Once the association phase is completed, the node should be able to generate the *Default Key* by using Eq. (1). This task can be handled without any problem because such parameters are stored within the MAC header and, hence, transmitted in clear.

C. Key Management Protocol

Since resource-constrained devices are unable to perform complex algorithms and protocols [33][34], a simple key agreement protocol, based on the Station-to-Station protocol [35], has been conceived for enabling a couple of nodes to negotiate a *Link Key*. To handle the KMP, a number of high-level commands have been defined. In line with IEEE 802.15.4e specifications, they are mapped into specific *Header Information Elements*, each one identified by an unique element ID. They are:

- *Control Information Element* (element ID set to $0x17$). It stores all the parameters that control the execution of the KMP and it is always sent together with one of the Information Elements discussed in the sequel. In particular, it contains the *KeyGenMode* (2 bits long), which describes the algorithm adopted for generating the key, the boolean *KeyFlag* (1 bit long), which is

TABLE I. CONTRIBUTIONS FOCUSING ON SECURITY ASPECTS IN LLNS

Contribution	Main covered features
IEEE 802.15.4 standard [6]	• a symmetric-key cryptography techniques based on the AES-CCM* algorithm; • 8 security levels, and a number of security-related attributes at the Media Access Control (MAC) layer; • a specific *Auxiliary Security Control* field within the MAC header, which contains all the parameters allowing the destination node to properly decrypt the received message; • a mechanism for uniquely identifying a given key among all the available ones, based on the knowledge of the MAC address of the node that generated the packet and the value of *KeySource* and *keyIndex* attributes associated to that key; • dedicated procedures handling the encryption and the decryption of MAC farmes, namely *outgoing frame security* and *incoming frame security* procedures, respectively.
ZigBee IP specification [15]	• security services at both network (i.e., *Network Level Security*) and application (i.e., *Application Level Security*) layers; • a dedicated entity, namely Zigbee Device Object (ZBO), which is in charge of handling security functionalities in a device; • two operational modes, i.e., high-security and low-security, that differ among them for the length of the key adopted to protect messages; • key distribution and joining procedures handled, in a centralized fashion, by the *Trust Center*; • three different keys, that are: the Master Key, used to finalize authentication procedures, the Network Key, used to protect management messages, and the Link Key, exploited to protect the communication between a couple of devices.
I-D presented within the IETF IPv6 over Networks of Resource-constrained Nodes (6lo) WG [16]	• accurate analysis on security threats; • Elliptic Curve Cryptography (ECC)-based key negotiation techniques for the establishment of secured communication links.
I-Ds presented within the IETF Low-power and Lossy Network (CORE) WG [17]-[19]	• definition of main security threads affecting a Wireless Sensor Network (WSN) at the application layer and the identification one well-known IP-based solutions able to increase the overall level of the network security [17]; • the design of a framework for protecting Constrained Application Protocol (CoAP) messages, which is based on the Datagram Transport Layer Security (DTLS) protocol [18]. • lightweight and robust authentication and key management scheme based on the DTLS protocol [19].
I-D presented within the IETF Routing Over Low power and Lossy networks (ROLL) WG [20]	• accurate analysis on threats and attacks that compromise the security level at the network layer • definition of some countermeasures to fix all the identified issues.
I-Ds presented within the IETF IPv6 over the TSCH mode of IEEE 802.15.4e (6tisch) WG [21][22]	• design of a more complete security architecture for industrial environments, which covers minimal security features for both layer-2 and layer-4 of the 6tisch protocol stack [21]; • definition of a secure and scalable key management framework to adopt in 6tisch networks, including the set of requirements that a key management protocols should satisfy in that framework [22].
Scientific contributions presented in [23]-[27]	• adaption of well-known security approaches at the application layer [23], [24], at the network layer [25], [26] and at the MAC layer [27].
Scientific contributions presented in [28]-[31]	• implementation of a KMP scheme based on centralized approaches [28] [29] and on distributed techniques [30] [31].

set to TRUE in the case the following Information Element will deliver key materials or to FALSE otherwise, the boolean *AuthFlag* (1 bit long), which is set to TRUE in the case the following Information Element will deliver an authentication field or to FALSE otherwise, the *KeySize* (5 bits long), which indicates the size of the transported key material, expressed in bytes, the boolean *FragEnabled* (1 bit long), which is set to TRUE when the key material contains a fragment of the certificate storing the public key of a mote, and the *FragNumber* (3 bit long), which indicates the fragment number associated to the key material field. Note that the *KeyGenMode* parameter is set to 00 when the key is computed through the anonymous DH algorithm. Other values are reserved and can be used for future upgrades.

- *Crypto Information Element* (element ID set to $0x18$). It is used to deliver the crypto suite needed to negotiate the key, composed by the *KeyMaterial*, which is the public key of the device, and *RAND*, which is a random value exploited to finalize the mutual authentication.
- *Authentication Information Element* (element ID set to $0x19$), which stores the *AuthField* used to execute the mutual authentication.

It is assumed that all devices store into the *PrimeNumbersTable* the same set of N prime numbers and their primitive roots, each one having size equal to S. The number of bits needed to identify each available prime number is equal to $N_p = log_2(N)$. We note that the size and the number of prime numbers can be chosen by the network administrator (or directly by the manufacturer). By increasing the size (or the number) of prime numbers it is possible to make the system more resilient to external attacks, thus improving the overall security level. Nevertheless, the higher is the size of prime numbers, the higher is the computational complexity required to generate keys. As a consequence, a good compromise between security and complexity is highly required in networks composed by constrained devices.

The KMP is initialized by the remote device, that has already completed the join procedure and that wants to establish a secured link with the coordinator. As illustrated in Fig. 1, it consists of the following six consecutive steps.

Step 1

The remote node selects a prime number, P, and the corre-

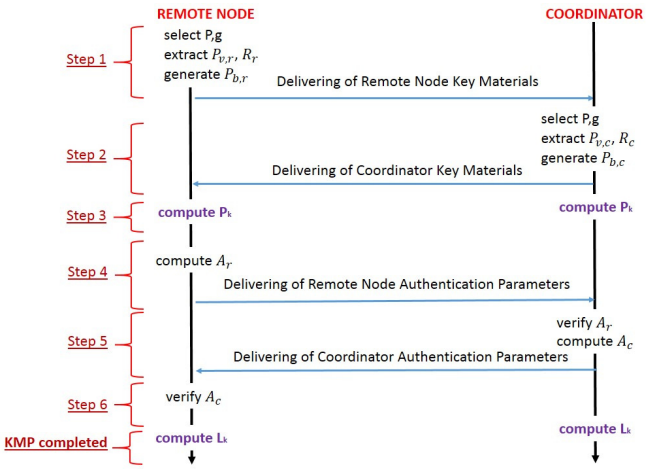

Fig. 1. Messages exchanged during the Key Negotiation Phase, between the Remote Node and the Coordinator .

sponding primitive root, g, from the *PrimeNumbersTable* by considering the latest N_p bits from the output of the following hash function:

$$H_{128}\{PAN_{ID}|D_k\}. \tag{2}$$

Then, it extracts two random numbers, i.e., $P_{v,r}$ and R_r, which represent its private key and a random value used for the mutual authentication, respectively. The public key, i.e., $P_{b,r}$, is generated according to the Diffie-Hellman (DH) algorithm:

$$P_{b,r} = g^{P_{v,r}} \cdot modP. \tag{3}$$

Hence, it sends to the coordinator a MAC message containing both the *Control Information Element* and the *Crypto Information Element*. Parameters of the *Control Information Element* are set as in the following: *KeyGenMode*=00, *KeyFlag*=TRUE, *AuthFlag*=FALSE, *KeySize*=S, *FragEnabled*=FALSE, and *FragNumber*=0. The *Crypto Information Element* will store, instead, *KeyMaterial*=$P_{b,r}$ and *RAND*=R_r. In the case the *Fully Secured* configuration is enabled, this message is encrypted with the *Default Key*, D_k. Otherwise it is sent in clear.

Step 2
Similarly to the remote node, also the coordinator selects a prime number, P, and the corresponding primitive root, g, from the *PrimeNumbersTable* by considering the latest N_p bits from the output of the Eq. (2). Then, it extracts two random numbers, i.e., $P_{v,c}$ and R_c, which represent its private key and a random value used for the mutual authentication, respectively. The public key, i.e., $P_{b,c}$, is generated according to the DH algorithm:

$$P_{b,c} = g^{P_{v,c}} \cdot modP. \tag{4}$$

Hence, it sends to the remote node that has initialized the procedure a MAC message containing both the *Control Information Element* and the *Crypto Information Element*. Parameters of the *Control Information Element* are set as in the following: *KeyGenMode*=00, *KeyFlag*=TRUE, *AuthFlag*=FALSE, *KeySize*=S, *FragEnabled*=FALSE, and *FragNumber*=0. The *Crypto Information Element* will store, instead, *KeyMaterial*=$P_{b,c}$ and *RAND*=R_c. In the case the *Fully Secured* configuration is enabled, this message is encrypted with the *Default Key*, D_k. Otherwise it is sent in clear.

Step 3
The remote node and the coordinator compute a *Pre Link Key*, P_k, using Eq. (5) and Eq. (6), respectively.

$$P_k = P_{b,c}^{P_{v,r}} \cdot modP. \tag{5}$$

$$P_k = P_{b,r}^{P_{v,c}} \cdot modP. \tag{6}$$

Step 4
The remote mote computes the authentication parameter, A_r, through the 128-bit hash function, as in the sequel:

$$A_r = H_{128}\{P_k||R_c||R_r\}. \tag{7}$$

Then, it sends a new MAC message to the coordinator to complete the mutual authentication. This message is composed by the *Control Information Element* and the *Authentication Information Element*. Parameters of the *Control Information Element* are set as in the following: *KeyGenMode*=00, *KeyFlag*=FALSE, *AuthFlag*=TRUE, *KeySize*=0, *FragEnabled*=FALSE, and *FragNumber*=0. The *Authentication Information Element* will store, instead, *AuthField*=A_r. This message is protected by using the *Pre Link Key* computed before.

Step 5
The coordinator verifies the validity of the received A_r parameter. In affirmative case, it computes the authentication parameter, A_c, through the 128-bit hash function, as in the sequel:

$$A_c = H_{128}\{P_k||R_r||R_c\}. \tag{8}$$

A new MAC message is hence generated by the coordinator and sent to the remote mote to complete the mutual authentication. This message is composed by the *Control Information Element* and the *Authentication Information Element*. Parameters of the *Control Information Element* are set as in the following: *KeyGenMode*=00, *KeyFlag*=FALSE, *AuthFlag*=TRUE, *KeySize*=0, *FragEnabled*=FALSE, and *FragNumber*=0. The *Authentication Information Element* will store, instead, *AuthField*=A_c. This message is protected by using the *Pre Link Key* computed before.

Step 6
The remote mote verifies the validity of the received A_c parameter.

Once all the steps have been completed, the remote node and the coordinator are able to generate the *Link Key*, L_k. The standard imposes to use the CCM* algorithm and a 128-bit key to protect MAC frames. At the same time, the CCM* algorithm assumes that each key must be used for a specific number of block ciphers (i.e., until the frame counter associated to a given communication reaches its maximum value). For each $i-th$ group of block ciphers, the *Link Key*, L_k, is computed according to Eq. (9).

$$L_k = H_{128}\{i||PAN_{ID}||P_k\}. \tag{9}$$

III. Experimental evaluation

To evaluate the impact that the conceived security framework has on communication latencies, an extensive experimental analysis has been conducted in real testbeds. To this end, we first implemented the designed framework in the well known OpenWSN stack [13]. In particular, starting from the work presented in [36], which extends the OpenWSN framework by introducing all data structures and variables referring to MAC PIB security parameters and security operations exploited to encrypt and decrypt MAC frames as defined in IEEE 802.15.4, we have implemented all the parameters and functionalities defined in Sec. II. Then, TelosB motes have been used to build testbeds. Despite the very limited capabilities offered by such a platform (16-bit microcontroller working at a maximum speed of 8 MHz, 48 kB Flash Memory, 10 kB RAM, and CC2420 radio module), it is highly used in todays' research to evaluate protocols and algorithms in LLN environments with extreme constraints. In our tests, we set the power level of each node in order to ensure a transmission range equal to 30 cm. To achieve lower computational efforts, the MD5 Hashing Function and the AES-CTR algorithm have been integrated within implanted security operations. Moreover, the Random Number Generator function already implemented in OpenWSN, based on a 16 linear shift registers, is used to generate random variables introduced in our scheme to provide the mutual authentication.

A. Testbed 1: time required to execute encryption and decryption operations

In order to analyze the impact that security features have on the total amount of communication latency, we analyzed the time employed to encrypt and decrypt MAC frames. The test is performed by varying the payload size from 117 bytes (the minimum MTU value) to 127 bytes (the maximum MTU size). As reported in Fig. II, the time required to perform cryptographic operations increases with the packet size because there are more plaintext blocks to encrypt. However, all values are included in the range $[295, 330]$ ms.

TABLE II. Mean Time to encrypt and decrypt MAC frames, varying the payload size

Payload Size	**Time to Encrypt the Payload**	**Confidence Interval at 95%**
117 byte	299.24 ms	$\pm$ 2.49
119 byte	303.71 ms	$\pm$ 2.11
123 byte	326.03 ms	$\pm$ 1.31
127 byte	328.25 ms	$\pm$ 1.55

B. Testbed 2: time required to initialize a secured domain and to negotiate a Link Key

We have investigated the time required to execute the *Bootstrap phase* and to create a secured link between the remote node and the coordinator. Obtained results have been reported in Fig. 2. As expected, the time required to complete the *Bootstrap* phase for the remote node is much larger than the one measured for the coordinator. In fact, if from one side the coordinator is able to execute the *Bootstrap* phase immediately after its startup (time instant in which it recognizes to be the coordinator), from the other hand, instead, the remote node should run the joining process before being able to start and

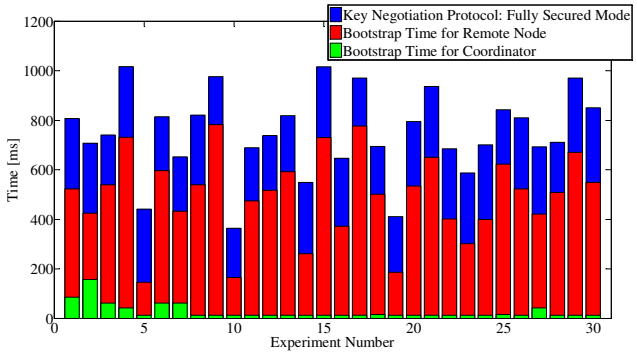

Fig. 2. Time required to setup a *Fully Secured Network*.

complete the *Bootstrap* phase. Furthermore, the configuration of a secured link, which is offered by the *Key Negotiation* phase, requires, on average, additional 260 ms from the end of the *Bootstrap* phase of the remote device. In any case, however, the secured link is available in less than 1.2 s. The variability of values reported in Fig. 2 is due to fluctuating delays introduced by the channel access procedure.

C. Testbed 3: study of communication latencies in more complex scenarios

To conclude, we have measured the application end-to-end packet delay obtained in a LLN network by varying the number of nodes and the application transmission rate, when the payload size is set to the maximum allowed value, that is 127 byte. As expected, the communication latency increases when both the application transmission rate and the number of nodes increases (Fig. 3). We can furthermore observe that the highest impact on the end-to-end application delay is essentially due to the cryptographic operations, which causes an additional workload at the coordinator side.

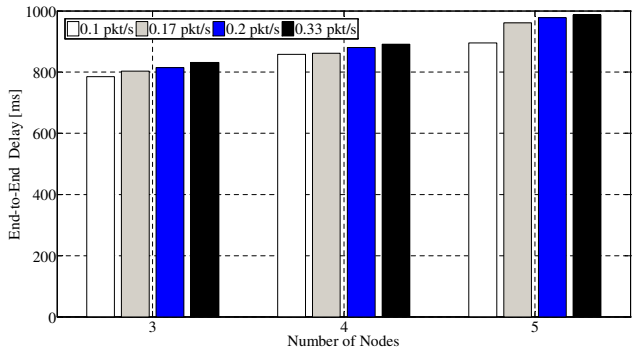

Fig. 3. Communication latencies vs application transmission rates, evaluated when the application payload size is set to 127 byte.

IV. Conclusions and future works

In this paper we presented a standard compliant and open source security framework for IEEE 802.15.4 networks. Moreover, we evaluated its impact on communication latencies through real experiments. Obtained results demonstrated that secured operations provided by the proposed framework, which include the setup of a secured link and the encryption/decryption of a MAC packet, involve to not negligible

communication delays and computational efforts. Such important findings have to be carefully taken into account when designing enhanced services in future IoT domains. In this context, our future research activities will cover the study of the framework feasibility in more capable devices, the analysis of its impact on the quality of service offered to real applications properly conceived for future IoT systems, the improvement of cryptographic operations through hardware accelerators, and the comparison with respect to other schemes presented in literature.

V. Acknowledgments

This work was supported by the PON projects (RES NOVAE, DSS-01-02499 and EURO6-01-02238) funded by the Italian MIUR and by the European Union (European Social Fund).

References

[1] O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key Applications and Protocols*, 2nd ed. Wiley, 2012.

[2] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, and M. Dohler, "Standardized Protocol Stack for the Internet of (Important) Things," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, Third Quarter 2012.

[3] L. A. Grieco, A. Rizzo, S. Colucci, S. Sicari, G. Piro, D. Di Paola, and G. Boggia, "Iot-aided robotics applications: technological implications, target domains and open issues," *Elsevier Computer Communication*, 2014, to appear.

[4] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of Things: Vision, Applications & Research Challenges," *Ad Hoc Networks*, vol. 10, no. 7, Sep. 2012.

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, October 2010.

[6] IEEE std. 802.15.4, *Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Standard for Information Technology Std., 16 June 2011.

[7] *802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Std., 16 April 2012.

[8] N. Accettura and G. Piro, "Optimal and secure protocols in the ietf 6tisch communication stack," in *Proc. of IEEE International Symposium on Industrial Electronics (ISIE)*, Jun. 2014.

[9] Accettura, N. and Grieco, L.A. and Boggia, G. and Camarda, P., "Performance Analysis of the RPL Routing Protocol," in *IEEE International Conference on Mechatronics*, Istanbul, Turkey, Apr. 2011.

[10] G. Boggia, P. Camarda, and V. G. Squeo, "ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems," in *Proc. of IEEE Int. Conf. on Communications (ICC)*, New York, USA, Apr.-May 2002.

[11] G. Piro, G. Boggia, and L. A. Grieco, "A standard compliant security framework for ieee 802.15.4 networks," in *Proc. of IEEE World Forum on Internet of Things (WF-IoT)*, Seoul, South Korea, Mar. 2014.

[12] Piro, G. and Boggia, G. and Grieco, L.A., *A standard compliant security framework for Low-power and Lossy Networks draft-piro-6tisch-security-issues-00 (work in progress)*, IETF 6TiSCH WG, October 2013.

[13] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wank, S. Glaser, and K. Pister, "OpenWSN: a standards-based low-power wireless development environment," *Tansactions an Emerging Telecomunications Technologies IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, p. 480493, 2012.

[14] Telosb datasheet. [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf

[15] Zigbee ip specification overview. [Online]. Available: http://www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx

[16] S. Park, K. Kim, S. Chakrabarti, and J. Laganier, "IPv6 over Low Power WPAN Security Analysis draft-daniel-6lowpan-security-analysis-05," IETF, Internet Draft, March, 15 2011.

[17] O. Garcia-Morchon, S. Kumar, S. Keoh, R. Hummen, and R. Struik, "Security Considerations in the IP-based Internet of Things draft-garcia-core-security-06," IETF, Internet Draft, Sep. 2013.

[18] L. Seitz and G. Selander, "Additional Security Modes for CoAP draft-seitz-core-security-modes-00," IETF, Internet Draft, Oct. 2013.

[19] A. Ukil, S. Bandyopadhyay, A. Bhattacharyya, A. Pal, and T. Bose, "Auth-Lite: Lightweight M2M Authentication reinforcing DTLS for CoAP," in *IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PERCOM)*, Mar. 2014, pp. 215–219.

[20] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, "A Security Threat Analysis for Routing over Low-Power and Lossy Networks draft-ietf-roll-security-threats-05," IETF, Internet Draft, Oct. 2013.

[21] M. Richardson, "security architecture for 6top: requirements and structure draft-richardson-6tisch-security-architecture-00," IETF, Internet Draft, Dec. 2013.

[22] S. Chasko, S. Das, R. Marin-Lopez, Y. Ohba, P. Thubert, and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TiSCH draft-ohba-6tisch-security-00," IETF, Internet Draft, Oct. 2013.

[23] Rescorla, E. and Modalugu, N. , *Datagram Transport Layer Security Version 1.2 (RFC 6347)*, IETF, January 2012.

[24] M. Brachmann, S. L. Keoh, O. Morchon, and S. Kumar, "End-to-End Transport Security in the IP-Based Internet of Things," in *Proc. of Int. IEEE Conf. on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–5.

[25] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3711–3720, 2013.

[26] P. Varadarajan and G. Crosby, "Implementing IPsec in Wireless Sensor Networks," in *Proc. of IEEE Int. Conf. of New Technologies, Mobility and Security (NTMS)*, Mar. 2014, pp. 1–5.

[27] T. Shon, B. Koo, H. Choi, and Y. Park, "Security Architecture for IEEE 802.15.4-based Wireless Sensor Network," in *Proc. of IEEE Int. Symposium on Wireless Pervasive Computing (ISWPC)*, 2009, pp. 1–5.

[28] E. Hagras, D. El-Saied, and H. Aly, "Energy efficient key management scheme based on elliptic curve signcryption for wireless sensor networks," in *Proc. of IEEE National Radio Science Conference (NRSC)*, Apr. 2011, pp. 1–9.

[29] L. Veltri, S. Cirani, G. Ferrari, and S. Busanelli, "Batch-based group key management with shared key derivation in the internet of things," in *Proc. of IEEE Int. Conf. on Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul. 2013, pp. 1688–1693.

[30] B. Tian, S. Han, S. Parvin, and T. S. Dillon, "A key management protocol for multiphase hierarchical wireless sensor networks," in *Proc. of IEEE Int. Conf. Embedded and Ubiquitous Computing (EUC)*, Dec. 2010, pp. 617–623.

[31] M. Wilhelm, I. Martinovic, and J. Schmitt, "Secure key generation in sensor networks based on frequency-selective channels," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 31, no. 9, pp. 1779–1790, Sep. 2013.

[32] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theor.*, vol. 22, no. 6, pp. 644–654, Sep 2006.

[33] R. Riaz, K.-H. Kim, and H. Ahmed, "Security analysis survey and framework design for IP connected LoWPANs, year=2009, month=Mar., pages=1-6,," in *Proc. of IEEE Int. Symposium Autonomous Decentralized Systems (ISADS)*.

[34] D. Altolini, V. Lakkundi, N. Bui, C. Tapparello, and M. Rossi, "Low power link layer security for IoT: Implementation and performance analysis," in *Proc. of IEEE Int. Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013, pp. 919–925.

[35] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Des. Codes Cryptography*, vol. 2, no. 2, pp. 107–125, Jun 1992.

[36] S. Sciancalepore, G. Piro, G. Boggia, and L. A. Grieco, "Application of IEEE 802.15.4 security procedures in OpenWSN protocol stack," *IEEE Standards Education e-Magazine*, vol. 4, no. 2, 2014.