Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis

Christine Hennebert, Jessye Dos Santos

Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis

Christine Hennebert, Jessye Dos Santos
CEA-LETI, Minatec Campus, 17 rue des martyrs, 38054 Grenoble, France
christine.hennebert@cea.fr; jessye.dossantos@cea.fr

***Abstract* — With the emergence of the Internet of Things (IoT), many devices organized into network, communicate by themselves on the Internet and send data, or private information on the web. It is essential to secure the transmitted data and the identities that may be disclosed to make these new technologies accepted by the largest number of citizens. However, the security mechanisms widely used on the Internet are too heavy to be integrated on small constrained objects. This paper describes the current protocols and security solutions that can be deployed in constrained resources. It shows the benefits and the limitations of each scheme – the security extension of IEEE 802.15.4e in Time Synchronization Channel Hopping (TSCH) mode , compressed IPsec, DTLS - embedded at different levels of the OSI model into the 6LoWPAN stack. It opens with the challenges one must tackle in the coming years. Several use cases are studied to envisage the security integration in Cyber Physical Systems for host-to-host and host-to-network communications. The privacy issue is also addressed and different ways to hide the device identity are discussed.**

***Index Terms* — 6LoWPAN, Security, Privacy, Protocol Stack, OSI layers, Internet of Things, Cyber Physical Systems, End-to-End Security**

I. Introduction

THE world has changed. Everything went very quickly with the advent of the Internet of Things (IoT) and the emergence in our daily life of heterogeneous objects able to connect to the Internet and communicate on the web. Since the "Nabaztag" launched in 2005 thus being considered as the first communicating object, many other innovative objects have been designed, and today objects as common as a crate of vegetables, a coffee machine or running shoes, exchange data on the Internet [1]. It is estimated that by the end of the decade, the number of communicating objects on the planet will reach 50 billion and will be 7 times greater than the number of humans.

These autonomous objects are provided with memories, a communication channel, a processor and sensors or actuators conferring a form of intelligence. Thus, with the IoT, the object becomes an actor of a process. It contributes at changing individual and collective behavior of people who interact via these objects. In a pervasive environment the communicating objects are able to recognize and to locate by themselves. The intelligence becomes ambient. The systems become ubiquitous [2]. In this context, the challenges of miniaturization and deployment in the environment are significant.

The convergence of the Internet with embedded systems led to the emergence of new systems deployed on a large scale and coupled to their physical environment. It is the concept of the Cyber Physical Systems (CPS) which includes RFID technology, wireless sensor networks (WSN), taking into account the mobility or the use of a smart phone to monitor various data from sensors via the cellular network [3].

The rise of the IoT and the CPS is enabled by the 6LoWPAN technology which means Internet as support infrastructure for the sensor networks [4].

The reference protocol for the Internet is IP (Internet Protocol). By extension, the CPSs are based on IP. But IP is not suitable for the sensor networks composed of resource constrained devices. 6LoWPAN [5] provides an adaptation of the IP world to the constraints of the sensor networks and enables the connection of the sensor networks world with the Internet. However, 6LoWPAN has been designed more to ensure the interoperability of both worlds - the sensor networks and the internet – so they meet the specific constraints related to the lack of resource of the sensor networks [6]. In the OSI abstraction model, 6LoWPAN is an adaptation layer located between the network layer and the link layer (Fig. 1).

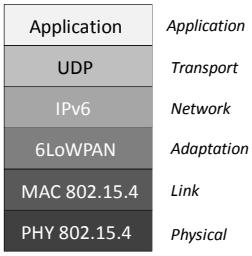

Fig. 1: 6LoWPAN Protocol Stack

6LoWPAN achieves low overhead by applying cross-layer optimization and compression of the headers of the IPv6 protocol stack. This allows making available about 81 bytes to

transmit data on the Internet into an IEEE 802.15.4 frame. However, it remains a key issue: ensuring security of data exchanged via 6LoWPAN [7].

When IPsec and then TLS (Transport Layer security) become mature technologies in the world of the Internet, their adaptation in the LoWPAN world is still a challenge (Fig. 2). These protocols need significant resources and generate a substantial overhead. An attempt to compress IPsec, only in transport mode, is presented by Raza in [8], while the emergence of Datagram TLS to secure the applications raises many questions about the implementation and the deployment in the real world.

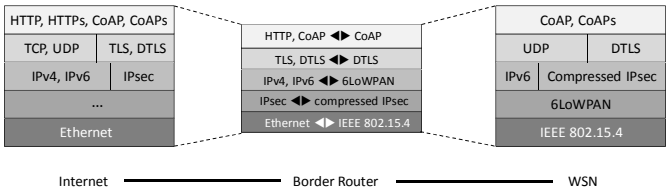

Fig. 2: Protocol Stack of a WSN connected to Internet

This article proposes a review of the security protocols into the 6LoWPAN protocol stack, and it addresses the challenges and the limitations for a pragmatic deployment in a physical environment. The paper is intended for designers and developers of the Internet of Things to provide the technical and decision-making basis in order to integrate security into the system upon its conception.

The paper is organized as follow: Section II summarizes the 6LoWPAN communication protocols standardized and recommended by the IETF (Internet Engineering Task Force). Section III details how the security protocols already adopted in the traditional Internet can be compressed and adapted to 6LoWPAN. It highlights the compromise done, the resulting overhead and the remaining issues. But, by using the security tools which threats do we deal with and how? Section IV provides answers to this question. In section V, the privacy challenge is discussed. Before concluding, section VI presents several network architectures implementing the security protocols for the IoT to ensure the security of end-to-end communications.

II. 6LoWPAN overview

A LoWPAN (Low Power Personal Area Network) is a set of small devices with scarce resources in energy, memory, throughput, power computing, that communicate through a low-power wireless standard. It forms a network of wireless sensors (WSN) with an available throughput up to 250kbits/second. To enable the connection of such networks to Internet, the Internet Protocol (IP) should be adapted to low-power, low-bandwidth and low-cost network communicating over IEEE 802.15.4 standard. The 6LoWPAN adaptation layer, standardized by the IETF [5], achieves the suitability of IPv6 for IEEE 802.15.4 networks. 6LoWPAN is actually embedded into Contiki with µIPv6 and TinyOS with BLIP (Berkeley Low-Power IP stack), two operating systems for WSN motes.

A. Integration of an IPv6 packet into an IEEE 802.15.4 frame

The length of an IEEE 802.15.4 frame at the physical layer is 127 bytes. Including at most the 25 bytes header of the Medium Access Control[1] sub-layer, 102 bytes remain available for the IP payload. The overhead of the IP headers following by optional extensions and the UDP header takes about 48 additional bytes. It remains only 54 bytes for the payload over UDP at application layer. It is obvious that an adaptation must be introduced to support an IPv6 MTU (Maximum Transmission Unit) of 1280 bytes. At this stage, no security mechanism is defined at any layer.

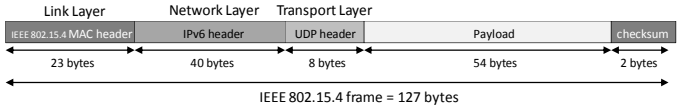

Fig. 3: Integration of an IPv6 packet into an IEEE 802.15.4 frame

The 6LoWPAN adaptation layer is located between the link layer and the network layer and should supply solutions for:

- Fragmentation and reordering of IPv6 packets
- Compression of the protocol stack headers
- Enabling stateless addressing
- Providing a basis for "mesh-under" routing
- Assuring consistency with the upper layers

When routing packets is performed by the network layer, it is called "route-over", and when the routing is implemented at the adaptation layer, it is called "mesh-under".

The traditional compression technique of the IP header consists in optimizing flow traffic while using stateful addresses. But flow-based compression techniques are poorly suited for LoWPAN, for which applications usually consist in singular exchanges instead of long-lived flows. So, the basic concept of 6LoWPAN is to use stateless addresses and shared-context compression between the different layers of the protocol stack. It allows routing protocols to dynamically choose paths without affecting compression efficiency.

6LoWPAN uses header stacking to express its capabilities in self-contained sub-headers: mesh addressing, fragmentation and header compression. The fragmentation header is elided when the datagram fit in a single frame. The mesh header is elided when the frame is delivered over a single hop and when the routing is performed at the network layer.

B. IEEE 802.15.4 frame

The total length of IEEE 802.15.4 frame is 127 bytes long, leaving in the best case a data payload of 102 bytes for the upper layers.

The physical header includes robust mechanisms to synchronize the received packet and decode the data carried

[1] The abbreviation MAC signifies two different notions: the Medium Access Control layer and the Message Authentication Code. In this paper, we use this abbreviation for Message Authentication Code. The Medium Access Control layer is written in plain text or called Link layer.

by the physical payload called PDSU (Physical Data Service Unit) [10]. It comprises 6 bytes that are not included in the 127 bytes of the IEEE 802.15.4 frame.

The data packet is one of the four following structures: Data, Beacon, Acknowledgment or Medium Access Control frames. The data frame handles the "Frame Control" field that specifies the network environment, the "Sequence Number" to verify by acknowledgment that all transmitted packets have been received, and the "Address" fields comprising the source and destination network identifier (0 or 2 bytes) and device identifier (0,2 or 8 bytes). In a given LoWPAN, the source and destination network identifier may be the same. The "Frame Checksum" field is a CRC (Cyclic Redundancy Code) of 16 bits to verify the integrity of the received frame. The acknowledging frame includes the same "Sequence Number" than the corresponding request.

Wireless IEEE 802.15.4 standard enables data transmission at 250kbps at 2.4GHz or 20kbps at 868MHz in Europe or at 40kbps at 915MHz in America. It is a low power and limited range communication standard. As data transmission requires more energy than computation, the compression of the transmitted information to save energy and to avoid message fragmentation is an important issue.

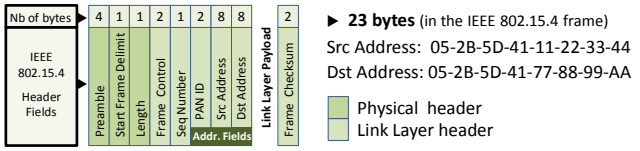

Fig. 4: IEEE 802.15.4 physical and link layer headers

C. IPv6 addressing

An IPv6 address is 128-bits long composed of a network part following by an identifier, and is represented in hexadecimal format. The network can be expressed by the first address of the network following by "::/" and the length of the prefix. The common part of the address is the prefix. There are several types of IPv6 addresses according to the targeted application and the routing scheme used:

- **Unicast address**: A unicast address identifies a unique interface. A packet sent to a unicast address is routed from one host to another host. These addresses are composed of two parts, each of 64 bits. The 64-bits prefix indicates hierarchically the localization into the network or the sub-network. The next 64 bits consists of the Interface Identifier (IID) that identifies the host into the network. The three main addressing types are:
  1) The *Link-Local* addresses are used in a local network and are "stateless" (i.e. automatically generated). Their prefix is "fe80::/10" or "fe80::/64" if the zeros are included [11]. A packet sent to such an address cannot be routed beyond the border router.
  2) The *Site-Local* addresses are identified by the prefix "fec0::/10". A packet sent to such an address cannot be routed beyond the border router. [12] indicates that this type of addresses is deprecated because of its ambiguity and the fuzzy definition of a "site".
  3) The *Global* address is an address unique for all the networks. It can be "stateful" (i.e. fixed manually) and begins by the prefix "001".

  Two particular addresses are also unicast:
  4) The *Unspecified* address 0:0:0:0:0:0:0:0 (or ::) is used by an IPv6 host which has not yet an assigned IPv6 address and launches for example a neighbor discovery.
  5) The *Loopback* address 0:0:0:0:0:0:0:1 (or ::1) is used by a host to send a packet to itself.
- **Anycast address**: it indicates a set of interfaces located at different locations and sharing the same address. A packet sent to an anycast address is delivered only to the first member of the group met.
- **Multicast address**: A multicast address concerns a set of interfaces possibly at multiple locations. The prefix used if "ff". A packet sent to a multicast address is delivered to each member of the group.

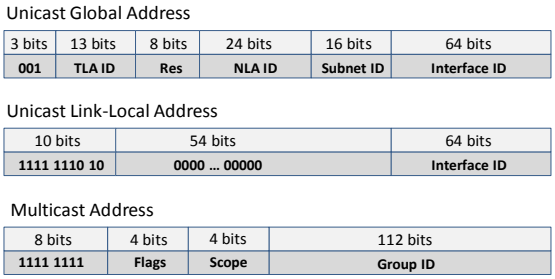

Fig. 5: IPv6 address types

The use of the Medium Access Control address to automatically generate the IPv6 addresses has raised privacy concerns. Indeed, the Medium Access Control address enables the host identification. To overcome this drawback, temporary random addresses or cryptographic addresses could be used. A DHCPv6 server can also provide a service of address assignment (see section IV).

D. Compressed IPv6 over UDP with LOWPAN_IPHC

At the network layer, the IPv6 protocol supports the multicast mode. At the IEEE 802.15.4 link layer, the packets are broadcast on a wireless channel. Hence, IPv6 multicast packets will be carried by link-local broadcast frames into the LoWPAN. To achieve this matching, the network interface identifier (IID) must match the PAN ID of the link layer.

The devices belonging to the same LoWPAN share some characteristics that enable the header compression following several assumptions:

- The version field is always elided and fixed to IPv6;
- The IID part of the IPv6 address is elided when it can be derived from the IEEE 802.15.4 Medium Access Control address;
- The packet length is derived from the "Payload Length" field included in the physical layer header - or the fragment header if the packet is fragmented;
- Both IPv6 fields "Traffic Class" and "Label Flow" can

be elided and fixed to 0;
- The "Hop Limit" is reduced to 8 bits.

Currently, the IETF in document RFC6282 recommends the use of LOWPAN_IPHC for the IPv6 header compression [13]. IPHC provides an efficient compression of both IPv6 addresses link-local, multicast and global. It makes the use of shared-context to elide the prefix of the IPv6 addresses. IPHC enables to code a prefix often used by the LoWPAN on a 4-bits context field, both for the source and the destination. Up to 16 contexts can be defined, also used to communicate with devices located outside the LoWPAN.

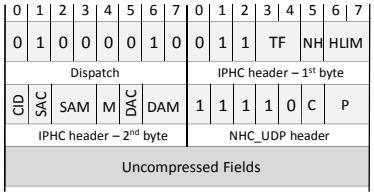

Fig. 6: LOWPAN_IPHC & NHC_UDP headers

The first three bits of the header indicates the use of IPHC, following by TF (Traffic Flow) and NH (Next Header) fields. When NH is fixed to 1, the next header is compressed with NHC. HLIM defines if the "Hop Limit" is carried in line or is elided and fixed to a predefined value. An additional context ID on 8 bits is inserted if CID=1. The SAC and DAC field indicates if the source and destination prefix are compressed using a shared context. Additionally, the field M=1 expresses a multicast destination. SAM and DAM detailed the compression mode both for source and destination addresses.

A NHC_UDP byte is introduced to enable the compression of the UDP header. The first five bits of the NHC_UDP are fixed to "11110". The field C allows the checksum to elide in very specific conditions. When P="11", the source and destination ports can be represented on 4-bits each in the interval 61616 to 61631.

The IPHC encoding enabled with NHC allows the compression of various extension headers. This capability will be useful to support security characteristics in the future.

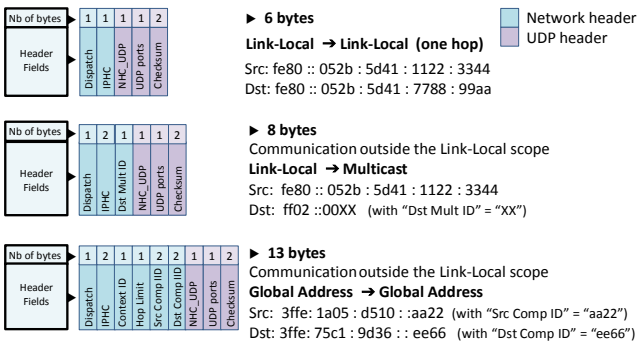

Fig. 7: Compressed UDP over IPv6 headers with LOWPAN_IPHC

In the best case scenario, the IPHC header can compress the IPv6 header down to two bytes, the "Dispatch" and the first byte of IPHC assuming a link-local communication. But the improvement is significant thanks to the use of the shared-context for multicast and global communications. As the "Group ID" of well-known multicast addresses is limited to few bits, the header can be drastically reduced. When routing over multiple hops outside the LoWPAN, the IPHC header grows to 7 bytes because the "Hop Limit" must be decremented at each hop and cannot be compressed, and the destination address cannot be statelessly derived from the link-layer address because it is not comprehensible for the intermediate hops. So, the prefix of both addresses can be compressed thanks to the use of context, and the source and destination IID take their compressed stateful expression.

E. CoAP

Sensor networks will play in the near future as a prominent place in RESTful architecture. They will interact with the web or via the cloud. In this perspective, the resource constrained nodes belonging to a sensor network need a light protocol to communicate. The new CoAP standard proposed by the IETF (Internet Engineer Task Force) meets this expectation [14]. The aim is to extend the web architecture to M2M (Machine to Machine) applications. CoAP is a communication protocol, application, generic and optimized for constrained systems. It provides communication between two "end points" via UDP.

CoAP fits into the "Payload" field of a UDP datagram. It can also be used over DTLS (Datagram Transport Layer Security). It supports IPv6 at the network layer and uses the IEEE 802.15.4 communication protocol both at link and physical layers.

CoAP interaction model is similar to the client/server model used by HTTP. It manages asynchronous messages between the client and server via the UDP datagram.

CoAP is based on the URI commands "coap" or "coaps" - when DTLS is used - to identify the resources and their location: `coap://Host:Port/Path/ ?Query`

The `Host` contains a literal address or an IPv6 address. This field must not be empty, otherwise the URI is considered invalid. The Port is the UDP port where the CoAP server is located.

The use of "coaps" implies secure UDP datagrams with DTLS. Resources available via "coaps" are not shared with "coap" even if their resource identifier indicates the same `Host:Port`.

III. SECURITY PROTOCOLS OVER 6LOWPAN STACK

For many applications and services, the data exchanged over the network need to be cryptographically secured. The aim is to ensure at least the authentication of the sender, the confidentiality of the data, the integrity of the frame and the network availability [15]. Mutual authentication and freshness are also additional security services often ensured.

The security can be handled at the link layer, the network layer and/or the application layer.

A. Security at the Link layer

IEEE 802.15.4 implements security features to achieve data

encryption and authentication. However, the last versions of the standard published in 2011 [9] and 2012 [10] do not specify how the keys have to be managed or what kind of authentication policies should be applied. These issues are addressed in the upper layers.

The following synthesis is based on the version IEEE 802.15.4-2011 [9] and its amendment [10] which introduced Time-Slotted Channel Hopping (TSCH) supporting efficiently multi-hop communications for industrial applications. Three fields in the frames are related to security issues:

- *Frame Control* (located at the Link Layer Header)
- *Auxiliary Security Header* (at the Link Layer Header)
- *Frame Payload* (in the MTU)

When the "Security enabled" bit of the "Frame Control" field is set to 1, an Auxiliary Security Header (ASH) is added. It consists of three new fields (Fig. 8):

- *Security Control* (SC) (1 byte) specifies which kind of protection is used (security mode).
- *Frame Counter* (0/5 bytes) protects the message from replay attacks.
- *Key Identifier* (0/1/5/9 bytes) indicates the key used to secure the communication with a given node.

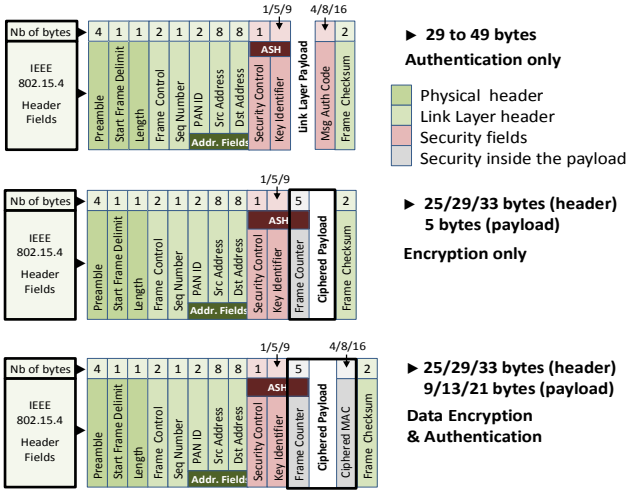

Fig. 8: Security at Link Layer IEEE 802.15.4

Seven security modes are provisioned: The AES-CBC-MAC cipher suite ensures the authentication of the frame including a 32, 64 or 128 bits Message Integrity Code (MIC) behind the payload. The AES-CTR enables encryption with cipher block of 128 bytes length to guarantee confidentiality. The AES-CCM* combines authentication with AES-CBC-MAC following by encryption with AES-CTR. For each mode enabling encryption, a 13-bits AES-CCM* Nonce, composed of the 8-bits "Source Address" concatenated with the 5-bits "Frame Counter" ensures a replay protection. The Frame Counter is incremented for each outgoing frame. When it reaches its maximum, the keying material must be updated. The standard allows moving the "Frame Counter" field into the Payload. The Auxiliary Security Header includes also a "Security Control" field that includes the security parameters and a "Key Identifier" field defining 4 ways to address the macKeyTable where the Key Descriptor is stored. The "Key Identifier" is composed of 1-bit "Key Index" concatenated with a 0/4/8-bits "Key Source" field. When a given emitter holds several keys, the "Key Index" indicates what key to consider, as the "Key Source" defines the source identity. The "Key Source" can be omitted when a key shared between a group of nodes is used. The MAC address of the emitter can be short for a "Key Source" field on 4 bytes or extended when the "Key Source" field is 8 bytes length.

The size of the macKeyTable is not defined and should be adapted to the node capabilities. No indication is given on how to build the initialization vectors. All the types of frames can be cryptographically secured, even the "Acknowledgment" frame that provides security to the protocols.

The "Frame Counter" field can be located either in the header or in the payload leading to different security considerations. When the "Frame Counter" field is located in the header, all the field enabling to build the AES-CCM* Nonce are in "clear" text and could be eavesdropped. An attacker can increment the Frame Counter, forges and injects a fake packet with the next right Nonce. But, when it is located in the encrypted payload, it must be decrypted before deciding if the frame is correct or not. This may waste energy if the frame is finally rejected.

B. *Security at the Network layer*

1) *IPsec overview*

IPsec [16] is a protocol suite for securing Internet by authenticating and encrypting each IP packet of a communication session. IPsec includes the negotiation of cryptographic keys used for encryption. IPsec is used to secure a flow of data between a host couple (host-to-host), a couple of gateways (network-to-network) or between a host and a gateway (host-to-network).

IPsec provides end-to-end security at the network layer. It is implemented in the operating system kernel. It protects exchanges without the application includes security primitives. In the absence of IPsec, TLS / SSL or DTLS should be included in the application to secure communications.

IPsec includes three protocols:

- *Authentication Header* (AH) provides integrity and authenticity of the source of IP datagrams for the whole header. AH also provides protection against replay attacks (Fig. 9).
- *Encapsulating Security Payloads* (ESP) ensures the confidentiality, integrity or source authentication of the data payload and the default header. This protocol also protects against replay attacks (Fig. 10).
- *Security Association* (SA) provides a set of algorithms and data to perform operations AH and / or ESP.

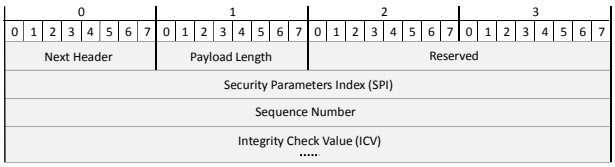

Fig. 9: IPsec AH header

IPsec can be implemented in transport mode or tunnel mode. In transport mode, only the payload of the IP packet is encrypted and / or authenticated. The routing is not affected because the packet header is not encrypted or altered. However, when the AH is used, the IP address cannot be translated because it would invalidate the hashed value (checksum). In tunnel mode, the entire IP packet is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new header. Tunnel mode is used to create a VPN (Virtual Private Network).

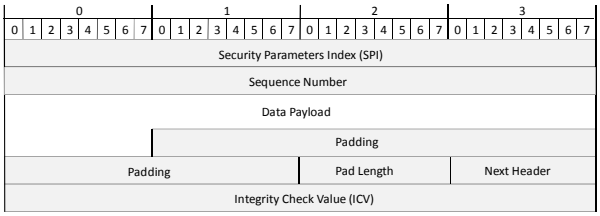

Fig. 10: IPsec ESP header

A key management protocol is associated to IPsec and used from the user interface. IKEv2 [17] is often used as key management scheme.

2) Compressed IPsec for 6LoWPAN

Security capabilities can be added to IP using IPHC header compression and NHC for the next header compression (Fig. 11). The NHC encoding consists in a NHC_EH byte including three bits for Extension Header ID (EID), so eight values. Two free slots ("101" and "110") remains available and will be used to indicate that a next header AH or ESP is to follow. In this case, the "Next Header" field is set to 1.

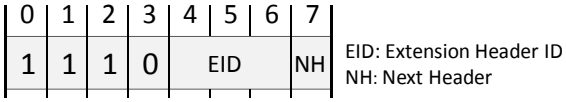

Fig. 11: LOWPAN_NHC header for extension

Written by Shahid Raza, [19] describes how IPsec can be adapted to secure the communication between two IPv6 nodes. It does not address the tunneling mode. AH and ESP protocols are introduced as header extensions of the compressed NHC header.

LoWPAN_NHC for AH: In the IPHC header, NH field indicates the use of a next header. NHC header for AH defines the way to compress IPsec AH header (Fig. 12):

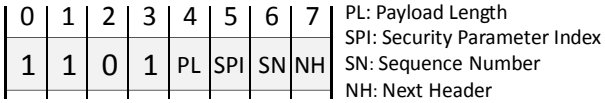

Fig. 12: NHC header for AH

The first four bits are the NHC ID for AH, set to "1101". The SPI and SN fields defines respectively the compression rate of the Security Parameter Index and the Sequence Number in the AH header. The field "Length" can be elided and the ICV size can be derived from the SPI value because the length of the checksum depends on the cryptographic algorithm used. Its smaller size is 12 bytes.

Integrated into the compressed IPv6 header frame, NHC for AH protocol adds a minimum overhead of 16 bytes (Fig. 13).

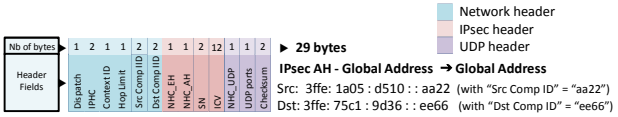

Fig. 13: Compressed UDP over IPv6 headers secured with AH

LoWPAN_NHC for ESP: With NHC for ESP, only SPI and SN fields can be compressed according to the following convention (Fig. 14):

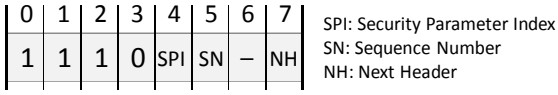

Fig. 14: NHC header for ESP

The first four bits represent the NHC ID for ESP, set to "1110". SPI and SN fields are the same as for AH.

The minimum length of ESP header without authentication is 18 bytes with AES-CBC and perfect alignment of the blocks. After compression, the ESP header can be reduced to 12 bytes. When ESP provides authentication, 12 bytes must be added for the ICV (Fig. 15).

As NHC_ESP performs encryption, the UDP header compression is no longer available. Indeed, the NHC_UDP would be encapsulated inside the encrypted content. And the receiver would imbricate decompression and decryption schemes.

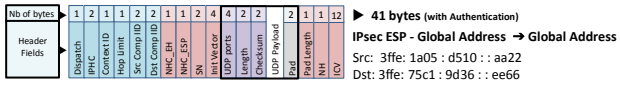

Fig. 15: Compressed UDP over IPv6 headers secured with ESP

The use of IPsec into a LoWPAN is possible but the encryption takes many resources and leads to a significant overhead. Furthermore, the simultaneous use of ESP and AH to perform confidentiality, integrity and authentication is very heavy in a LoWPAN.

3) IPsec Security Association (SA)

The key negotiation scheme IKEv2 [17] is not available for the LoWPAN as it handles too much signaling. The authors of [18] introduce a lightweight IKEv2 scheme for compressed IPsec. A dedicated NHC header, recognized by the ID bits "1101", is defined. SPI=0 indicates that the default defined SA is used, instead than a singular SA with SPI=1.

IKEv2 is a protocol for establishing a session key between two peers. While IKEv2 uses RSA asymmetric cryptography, lightweight IKEv2 is based on Elliptic Curve Cryptography (ECC). The Diffie-Hellmann protocol for key exchange is used in both cases.

C. Security at the Application layer

1) Overview of DTLS

DTLS (Datagram Transport Layer Protocol) is a protocol used to secure network traffic. It is based on TLS and usable with UDP datagram. So, DTLS manages the UDP packet loss, the packet reordering at reception and operates on smaller frames [21].

DTLS is a protocol in two layers: the bottom layer is called "Record Protocol" and can provide a secure symmetric key encryption to ensure the confidentiality and/or the message integrity in the ciphered mode. The upper layer includes four protocols (see Fig. 16):

- *Handshake*: This protocol is used to negotiate security settings and generate a session key for secure communications.
- *ChangeCipherSpec*: This protocol enables the change of the current cipher suite.
- *Alert*: This protocol can be used at any time during the "handshake" to report errors or "warnings".
- *Application Data*: Using this protocol, application data are fragmented, compressed and could be encrypted with the security mode in progress.

The handshake protocol encapsulates 11 types of messages used by the handshake mechanism (Fig. 16).

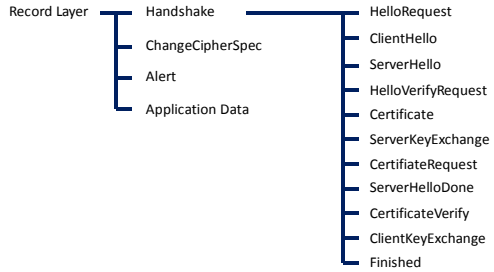

Fig. 16: Structure of the DTLS messages

Fig. 17 details the structure of the ClientHello message launched during the handshake.

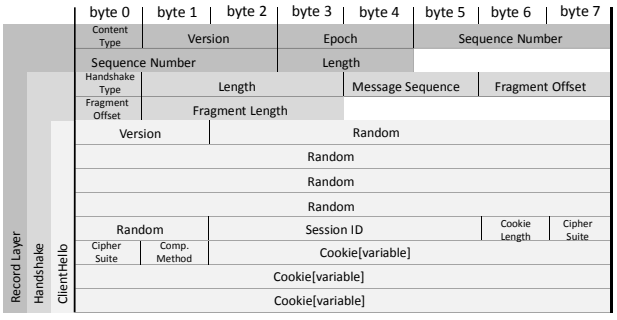

Fig. 17: DTLS ClientHello message
carried into “ClearText” or “Compresed” Record Layer

When key materials negotiation is achieved, the data can be carried securely inside a ciphered Record message (Fig. 18).

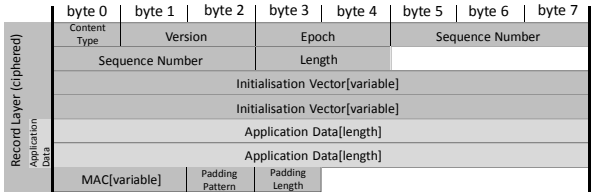

Fig. 18: DTLS ApplicationData message
carried into a “Ciphered” Record Layer

During the deployment stage, a node is provided with secret keys and access control lists according one of these 4 security modes:

- *NoSec*: DTLS is not available.
- *PreSharedKey*: DTLS is used. A list of pre-distributed symmetric keys is established, and for each key the list of nodes with which it can communicate. If more than two nodes share the same key, this key allows authenticating as part of the group. The entropy of the pre-distributed keys should be sufficient to make difficult brute force attacks and dictionary attacks. Communications in clear text on the client identity may compromise privacy.
- *RawPublicKey*: DTLS is used and the node is provided with a pair of asymmetric keys, but without a certificate. The node gets an identity and a list of nodes with which it can communicate. In this mode, the node is provided with an asymmetric key pair generated by the manufacturer and installed on the node before deployment. It must support the cipher suite TLS_ECDHE_WITH_AES_128_CCM_8 (RFC5246), the ECDSA signature scheme and secp256r1 elliptic curve cryptosystem based on prime fields.
- *Certificate*: DTLS is used and the node is provided with an asymmetric key pair and a X.509 certificate known by a certification authority. The node gets also a list of trust anchors that can be used to verify the certificates. It is based on secp256r1 elliptic curve cryptosystem.

In “NoSec” mode, the system sends packets over UDP using the protocol "coap". The other three security modes use DTLS, which is indicated by "coaps".

DTLS has been designed for user end-point (computer, laptop, tablet, smartphone…) and is not optimal for constrained resources. For example, large buffers are needed to manage the loss of messages or to store all the fragments of a message. In addition, the use of X.509 certificates to perform mutual authentication is not suitable for constrained resources because their size can be very large. Multiplying the number of fragments creates a high probability of packet loss. There are still many challenges to overcome to make DTLS an effective technique for securing a constrained network. The protocol must be simplified and a compromise between security and a lightweight implementation must be found.

For constrained resources, all modes of DTLS are not applicable. The initial "handshake" enabling the authentication

of two elements requires a lot of resources (Fig. 19).

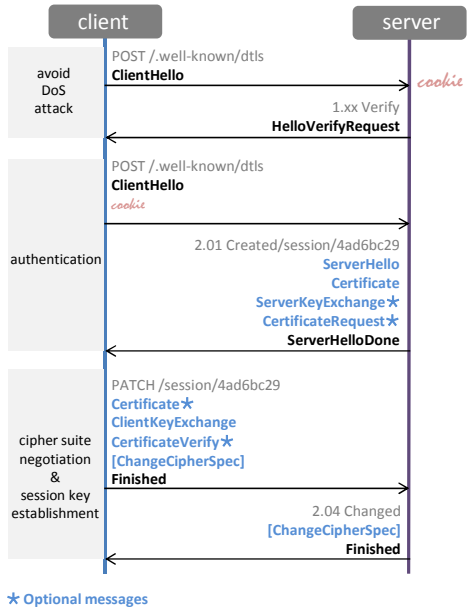

Fig. 19: Handshake Protocol

The complexity of the "handshake" protocol is a big problem for the nodes. Up to 15 messages distributed on 6 flights are needed to establish a secure connection. Compared to TLS, DTLS introduced two new messages containing a cookie to prevent Denial of Service (DoS).

2) Compressed DTLS

Based on the necessity to achieve better energy efficiency by reducing the message size and to avoid as much as possible the message fragmentation, the authors in [22] propose a technique to compress DTLS header in a standard compliant way into a 6LoWPAN network.

DTLS provides a handshake mechanism enabling new nodes to authenticate to the "master" when they reach the network, and to negotiate the cipher suite used for data encryption and signature. For very resources constrained nodes, the handshake is not conceivable and the nodes are deployed with pre-shared keys and a pre-defined cipher suite.

[22] proposes two independent compression schemes:

- For Handshake ClientHello and ServerHello messages;
- For Application Data messages exchanged for the application purpose that can be encrypted and signed using the cryptographic features of the cipher suite.

The DTLS compression leans on the LOWPAN_NHC for UDP transport header. To indicate that compressed DTLS is following – i.e. the UDP payload is compressed as well as the UDP header - , the ID bits are set to "11011" value instead than "11110" used only when the UDP header is compressed.

Compression of Handshake messages: A new LOWPAN_NHC is defined to handle the compression of both Record Layer and Handshake Headers (Fig. 20). As the ID bits "1000" identify this new next header, V is set to 0 when the last DTLS version (currently v1.0) is used and the "Version" field elided into the frame. The "Epoch" can be reduced to 8 bits when EC=0. One bit is assigned to the "Sequence Number" compression, originally 48-bits length. If SN=0, the 16 lower bits are retained. Bit F indicates whereas the message is fragmented (F=1) or not. When the message is carried in a single fragment (F=0), the "Fragment_Offset" and "Fragment_Length" fields are omitted. The "Length" field is always elided as it can be deduced from the lower layers. The "Content_Type" is elided as the presence of this NHC indicates handshake content. At the opposite, "Message_Type" and "Message_Sequence" fields are carried in line.

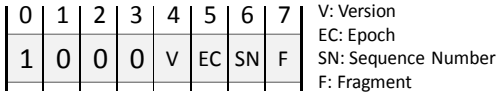

Fig. 20: LOWPAN_NHC for Record and Handshake

The handshake protocol (Fig. 19) encapsulates 11 necessary or optional messages. When the "Message_Type" field indicates that a ClientHello message or a ServerHello is following, the respective NHC byte is inserted into the header (Fig. 21):

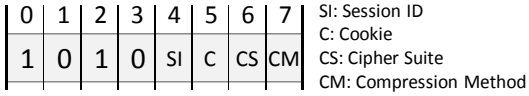

Fig. 21: LOWPAN_NHC for ClientHello

The ID field of the ClientHello NHC header is set to "1010" (see Fig. 21). The "Session_ID" can often be omitted, which is indicated by SI=0. It is the case when no session is available or when new security parameters are negotiated. The ClientHello message is sent twice during the handshake protocol: the first time to initiate the dialogue and request for a cookie, and the second to request for cryptographic features. Bit C=0 indicates that both "Cookie_Length" and "Cookie" fields are elided. The "Cipher_Suite" index can be omitted with CS=0 and set to a default cipher suite, for instance DTLS_ECDHE_ECDSA_WITH_AES_128_CCM_8. No "Compression Method" is used when CM=0. The "Random" field is essential to ensure security and is always carried in line, and the "Version" is the same as in the Record header.

Using this NHC compression, 23 bytes are safe in the ClientHello message (Fig. 22). When a Cookie is carried, one byte and its length must be added to the 43 bytes of the basic compressed ClientHello message.

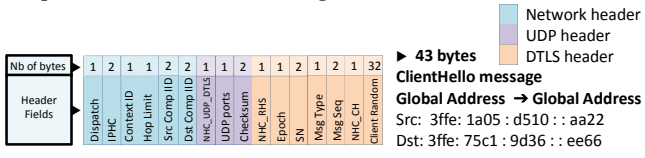

Fig. 22: Compressed DTLS Handshake ClientHello message over compressed UDP / IPv6 headers

The ServerHello NHC header is identified by the "1011" ID value (Fig. 23). The Sever can re-negotiate the DTLS version

used and can achieve this re-negotiation while setting V=1. The status of the others fields are the same than for ClientHello. The ServerHello message always carried a Cookie.

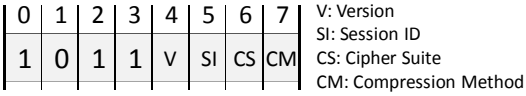

Fig. 23: LOWPAN_NHC for ServerHello

Compression of application Data messages: When the DTLS Record Layer carried an Application Data message, the NHC compresses only the Record header. In this case, the ID bits are set to the value "1001" (Fig. 24). The fields have the same significance than for NHC for Record & Handshake. The SN takes two bits which allow compressing the "Sequence Number" field with a better granularity.

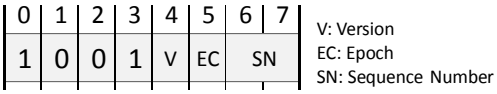

Fig. 24: LOWPAN_NHC for Record only

As key negotiation append before exchange of data into the network, the "Content_Type" field should be elided and the Ciphered Record Layer for Application Data message is used by default. The cipher suite features enables UDP payload encryption and signature. The required security issues are achieved: data confidentiality, frame integrity, emitter authentication, freshness thanks to the sequence number and availability improved with compression.

This compression scheme enables the saving of 9 bytes over 25 of the original Ciphered Record Application Data header (Fig. 25), that corresponds to 36% of the header length.

Fig. 25: Compressed DTLS Ciphered Application Data message over compressed UDP / IPv6 headers

While the others messages of the handshake protocol cannot be compressed, the authors of [24] proposed to simplify the protocol under certain assumptions. In [25], a reduction of the certificate size is envisaged.

IV. Security Concerns

Link layer security ensures the security of the wireless medium whereas upper layer security is designed to achieve end-to-end security between two peers.

It is essential to understand the security requirements and the threats to use against the right counter-measures. In the field of LoWPANs we currently have several tools, including cryptographic, to counter many attacks.

Unfortunately, these tools cannot be deployed at the same time because of the constraints of low-power network and devices. Many challenges need to be solved to deploy and easily manage a secure network at a large-scale. One of the hard points is the deployment and management of cryptographic keys. That's why we devote a paragraph about it. Non-cryptographic counter-measures are briefly exposed and a table summarizes the content of the discussion.

A. Security Requirements

The main security needs for 6LoWPAN networks are:

Data Confidentiality: makes the data content non understandable to unauthorized devices or users,

Data Authentication: verifies the identity of the data source,

Data Integrity: ensures that the received data is correct,

Data Freshness: guarantees that the received data is original and has not been replayed,

Network Availability: ensures that the network services are always available for the legitimate devices or users,

Network Robustness: makes the network usable even when an attack occurs,

Network Resiliency: maintains a given security level over the network even when a node is compromised,

Network Resistance: is the ability to avoid that an attacker takes the control of the network via a compromise node,

Energy Efficiency: prevents battery drain in the network,

Assurance: is the ability to dispatch information over the network to ensure their security,

Device Authorization: checks the legitimacy of a device and enables it to join the network.

B. Threat Analysis for 6LoWPAN

Physical Attacks, such as node destruction, relocation or masking, can make the resource provided by the node inaccessible. Moreover, the cryptographic secrets store inside the node can be extracted allowing replay attacks, packet injection, making a clone or node reprogramming. At the physical layer, Deny-of-Service (DoS) attacks can be launched by jamming or tampering the radio signal.

At the link layer, an attack on network availability can consist in flooding the network with large packets to occupy the entire bandwidth. Packet injection can also lead to battery exhaustion or to packet collision followed by packet loss.

[26] presents two fragmentation attacks on "mesh-under" routing protocol handled by the 6LoWPAN adaptation layer. As the destination address is mentioned only in the first fragment, an attacker can easily flood the network with next fragments duplicated at the time of reception. Another attack consists in maliciously reserving space in the re-assembly buffer with incomplete packets until saturation.

Numerous attacks can be launched at the network layer. Several attacks on routing, such as Selective forwarding, Sinkhole attack, Sybil attack disrupt the network services from a compromise node inside the network. The Wormhole attack is more dangerous as it does not need to compromise a node: The attacker eavesdrops a packet and tunnels it to another node of the network. This attack can be launched at the start during the neighbor discovery phase.

At transport layer, a compromise node can inject message over the network to force the end-point to request retransmissions.

Application data may be peeked by an illegitimate user or impersonated. Attacks can also be launched to disrupt the data aggregation.

C. *Key Management*

The secret key is the support of cryptography security. It must be remained secret during the whole lifespan of the network, from deployment to revocation. So, the key management is an important issue of the security. The key management implies the concept of authorization because the security credentials and keys are given only to devices able to prove their legitimacy.

Link Layer: All the IEEE 802.15.4 frames should be cryptographically protected to ensure the frame integrity, authentication, freshness and optionally confidentiality. But the standard does not explain how to deploy a secure IEEE 802.15.4 network, to securely add a new node to the network or to manage the cryptographic keys over time. The 6TiSCH working group, whose goal is enabling IPv6 over TSCH mode of IEEE 802.15.4e standard, introduces in [27] a security framework in order to provide security services at the link layer. Three kinds of keys are defined: (1) the master key pre-distributed initially in all the nodes of the network, (2) the network key shared by the legitimate nodes after authorization and authentication services provided by the upper layers, and (3) the link key established between neighbor legitimate nodes. At the start, a Setting-up phase consists in storing in software or hardware secured memory of the node, the master key and potentially any initial secrets. An out-of-band channel may be set for this operation. The bootstrapping phase initiates a secure communication, thanks to the shared master key, between the node and the network coordinator to configure the security attributes and the security level of the remote node. Then, upper layers can provide authorization and authentication services to provide security credentials, as token, to the node and disclose the symmetric network key, shared over the IEEE 802.15.4 network. A last phase called key negotiation may consists of establishing pairwise link key between neighbor nodes of the LoWPAN. The security level of the local network depends on the capability of its nodes to perform or not these four phases.

As requirements, the master key must be physically secured to avoid node tampering. An attacker who is able to get this key can take the control of the whole IEEE 802.15.4 network. The lifespan of the master key expires when the “Frame Counter” reaches its maximum value. Its upgrade is not defined. The network key disclosure implies key management at the upper layers. The link key establishment involves protocols based on asymmetric cryptography, the owning of a certificate and a couple of public/private key. The nodes able to execute such protocols are not so constrained. Lightweight mechanisms for each deployment phase must be designed in order to offer to the most constrained nodes the higher security level possible.

Network Layer:

An efficient key establishment into the LoWPAN remains an open issue. Lightweight IKEv2 is based on prime fields ECC family, more secure but wider in memory than ECC based on binary fields. Lightweight IKEv2 is designed to establish a session key that will secure a significant data flow exchanged between two peers. In the area of IoT, the data exchanged are more usually measures from sensors than long data streams. Moreover, the IPsec protocol does not provide any acknowledgement mechanism.

Application Layer:

The full handshake can flexibly negotiate a session key between two peers, without pre-distribution. However, the signaling and the size of the messages exchanged are high. The compression can only be applied for few message types. However, the certificate issue is addressed and highlights the need of development of a shorter certificate for the constrained nodes.

Both IPsec and compressed DTLS support manual pre-shared key and automatic key exchange based on asymmetric cryptography. Manual technique is tedious for high density network and is not scalable. Automatic protocols are heavy for constrained LoWPAN, but are flexible and scalable. In the future, 6LoWPAN needs to define its own keying management methods that require low overhead in packet size and few signaling protocols.

D. *Non Cryptographic security tools*

The cryptographic security can be really efficient if the cryptographic features (keys, seeds…) are physically protected against stealing or disclosure. The constrained devices should embed physical protection as secure element or secure firmware to avoid side channel attack and cloning.

The version of standard IEEE 802.15.4e-2012 includes security features for acknowledgement frames to avoid many well-known attacks. The technology CSMA-CA (Carrier-Sense Multiple Access – Collision Detection) is to ensure that the radio channel is available before transmitting. It enables channel hopping and prevents from physical DoS attacks on the radio channel.

IDS is a security approach that monitors the network activity to detect signs of intrusion or anomalies. In addition to cryptography, the implementation of an IDS in a 6LoWPAN network should be useful to ensure the network services.

IPsec uses SeND protocol (RFC 3971) ("Secure Neighbor Discovery protocol") to discover its neighbors. An extension of this protocol, called LSeND ("Lightweight Secure Neighbor Discovery protocol"), has been designed for 6LoWPAN networks and is described in the patent [20]. In the future, 6LoWPAN needs to define open source solutions for the discovery service.

Table 1: Security Elements

Layer	Security mechanism	Header Overhead	Requirement achieved	Foiled Attack
Physique	CSMA-CA	None	Availability	Jamming / Collision / Flooding
	Secure firmware	None		Node Tampering
	Secure element	None		Cloning
Link	MIC	6 to 26 bytes	Authentication & Integrity	Packet Injection
	AES encryption only	7 to 15 bytes	Confidentiality	Eavesdropping
	AES-CCM* Nonce	11 to 29 bytes	Authentication, Integrity, Confidentiality & Freshness	Replay Attack
	Address Filtering	None	Energy Efficient	DoS / Battery Exhaustion
Adaptation	Hash Chain	8 bytes	Integrity	Fragmentation Attack
	Split Buffer	None	Availability	DoS / Buffer saturation
Network	IPsec AH	16 bytes	Authentication of the emitter & Integrity Network Resiliency, Robustness, Resistance	Packet Injection Replay Attack
	IPsec ESP	28 bytes	Confidentiality between two peers	Eavesdropping Replay Attack
	Secure Routing	/	Availability	Routing Attacks
	Secure Neighbor Discovery	/	Protect Network Services	Intrusion
Application	Compressed DTLS Ciphered Layer	16 bytes	Authorization through a token &Authentication of the emitter & Integrity & Confidentiality between two peers using a given application Network Resiliency, Robustness, Resistance	Aggregation Data Peeking Packet Injection
	IDS	/	Network Services	Every Intrusions

V. Privacy Concerns

Among the security services, the encryption ensures the confidentiality of the data exchanged over the network. The integrity and the authentication of the whole frame can be guaranteed, but the confidentiality of information included in the header remains unprotected. This causes a problem for privacy. The header carried information called “metadata” and is used for “data mining”. They may enable tracking, as well geo-localization, identification as social links inference or activity recognition.

In this section, we envision how the private information included in the header may be protected.

A. Temporary Stateless Addresses Auto-configuration

The use of a constant part in the address field is fundamental to route the packet over the network. This information cannot be easily hidden. Even when the payload is ciphered, the addresses included in the header are sent in clear text and can be eavesdropped. The private information carried in the packet header should be hidden to avoid tracking and data mining.

IPv6 addresses are divided into two distinct parts: the interface identifier (IID) and the topology. The topology changes for mobile devices and carries localization information. The IID remain constant as it identifies a given device. “Data mining” techniques that correlate the activity with the address are based on the IID tracking.

A compatible approach with the auto-configuration of stateless addresses consists in modifying the IID over time. Thus, it becomes more difficult to associate an activity with a device (or a person) even if the routing prefix doesn’t change.

The document RFC2462 [28] details a methodology for generating a temporary link-local address of a given IEEE 802.15.4 interface without the need of a DHCP server. It also tackles the extension of a temporary random stateless address to global scope addressing for outgoing message. Pseudo-random sequence of interface identifiers (IID) is generated with a MD5 hash function from a random component and the IEEE 802.15.4 identifier. A dedicated algorithm verifies that the generated IID has not already been used. The concatenation of the 64-bits random IID with the 64-bits prefix forms a temporary IPv6 address. When a new address is created, the old one is deprecated to avoid its further use.

Each application should have the choice to prefer the use of public IPv6 address or the use of temporary address to

communicate with a given node (RFC3041) [29]. A UDP-based application could be unique to get the knowledge about the addresses currently in use. In this case, a heuristic could be useful to decide when the addresses expire. The APIs should be developed in order to enable applications to indicate their "privacy" needs with an adequate granularity.

Auto-configuration through stateless addressing allows a host connecting to a network, configuring its address, and establishing a communication with the other nodes without having registered nor authenticated into a local sub network. Thanks to this technique, non-authorized users can connect to the network and use it. Many Denial of Service (DoS) attacks can be launched thanks to the use of stateless addresses generated by auto-configuration (RFC4862) [30].

The final user must be able to voluntarily activate the use of temporary addresses that protect its private profile while avoiding the access to some services or application (RFC4941) [31]. The network administrator must also be able to deactivate the use of temporary addresses, for instance in order to debug easily or for a chosen prefix. The use of temporary addresses can perturb some applications that use private information. Some servers deny communications coming from clients whose IP address doesn't match with the DNS name. If an address expires before the application has ended, it can also create bugs and stop the application. Furthermore, if an application opens several sessions, it can expect the client to have the same address for all sessions. This requirement cannot be fulfilled with the use of temporary addresses.

If a node uses the same prefix over a long period, changing the IID will not be enough to protect its privacy. To get an efficient temporary addressing, the prefix must not be static or the same for a large number of nodes.

Moreover, the addresses may be spoofed. On a high density network where temporary addresses are frequently created, it can be difficult to distinguish between a legitimate address and a spoofed address composed of a correct prefix and a nonexistent IID. However, even when the address is spoofed, the identity of the owner remains protected.

B. Cryptographically Generated Address (CGA)

CGA (Cryptographically Generated Address) aim is to prevent against stolen or spoofed IPv6 addresses (RFC3972) [32]. It is based on the use of asymmetric cryptography relying on couple of public key/secret key. It consists on binding the IID of the address - generated with a cryptographic one-way hash function - with the public key of the node. This scheme can be applied without certificate or security infrastructure.

The public key of the device is cryptographically linked to its identity carried by its address. The address owner uses its secret key to sign the message and prove its identity to assure the authentication from its address.

Following this scheme, a attacker can create a new address based on an arbitrary prefix and its own public key making profit that the CGA is not certified. But, the attacker cannot steal the identity of a legitimate node.

There is another limitation to the use of CGA: No mechanism is available to prove whether an address comes from a CGA or not. A attacker can intercept a CGA address and use it as a non-cryptographically signed address. Nevertheless, he will have difficulties to make profit of this hack because nodes give priority to signed addresses.

Hence, CGA brings the same level for pseudo-naming as temporary random addresses described in [29].

Two other minor limitations of the use of this CGA for "privacy" can also be highlighted:

- The generation of a new address requires a high computing power and consumes significant energy. This is orthogonal with the need to frequently renew the addresses.
- The public key is disclosed in a "SeND" message. If the transceiver wants to stay anonymous through the nodes used (multi-hop), they have to generate not only a new address but also a new public key. However, the address is the unique identifier of the node at the link layer. So the node may keep the same public key as long as the address does not change.

The CGA scheme described in [32] is based on a RSA cryptosystem. The RSA cryptography is heavy and not adapted for a use into LoWPANs. To use such a privacy protection over LoWPAN networks, a cryptographic address generation scheme based on Elliptic Curve Cryptography (ECC) must be developed.

VI. ARCHITECTURE FOR END-TO-END SECURITY

The main objective of these recent years has been to ensure the interoperability of communication protocols between the world of LoWPAN and the World Wide Web (WWW). Nowadays, this work has reached a certain maturity and is used to route end-to-end communications in a Cyber Physical System. But what about the data security or the user privacy? Security protocols such as IPsec or TLS deployed in the domain of traditional Internet have won the trust of the society and are now widely used by citizens. The standardization effort done on 6LoWPAN aims to ensure the interoperability of these security protocols with the LoWPAN world. This leads to the introduction of resources at "shoe-horn" and bandwidth consumer protocols into constrained, even very constrained systems. Many compromises are considered questioning the notion of end-to-end security.

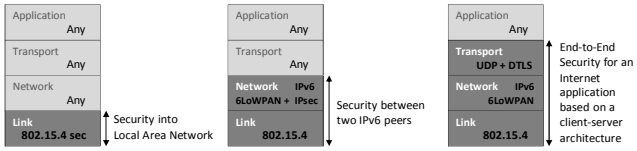

Fig. 26: Security of the communication at different level of the protocol stack

At the link layer, the frame can be protected (Fig. 26) ensuring some security requirements for the LoWPAN. Implemented at the network or transport layers, it could

provide hop-by-hop security or end-to-end security with many restrictions.

In this section, we propose to analyze the security issues in three configurations:

- Internally to a LoWPAN;
- Between two hosts, the first one belonging to the WWW and the second one to the LoWPAN world;
- Between a host and a LoWPAN network.

Each of these configurations supports different types of applications or services.

A. End-to-end security inside a LoWPAN

Inside the LoWPAN (Fig. 27), the security can be envisaged either at the link layer with the security extension of IEEE 802.15.4e in TSCH mode, either at the network layer with compressed IPsec or between transport and application layers with DTLS. The stateless addressing allows both to reduce the address length and to pseudo-name devices with the use of temporary identities generated cryptographically or randomly. By this way, privacy can be ensured and maintained for the packets routed inside the LoWPAN.

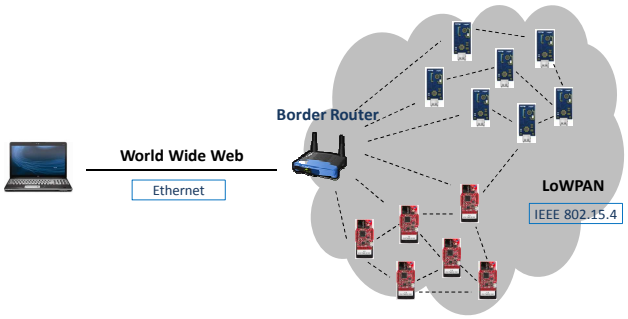

Fig. 27: Communications internal to the LoWPAN

1) Security at link layer

At the link layer, the security should be used for all types of frames. Each unsecured frame is a flaw that can be used by an attacker. The question is “what level of security should be chosen?”. The answer will depend on the criticality of the application supported. The node capabilities will be chosen in function of the application needs and the cost.

The lowest security level consists in using a symmetric shared master key for the whole local network. This implies that each node is able to physically secure this key because if a node is compromise, the whole local network is broken. This key must also be updated over time when the frame counter has reached its maximum value. The confidentiality can be ensured against the outside but not inside the network. All the nodes have access to the information exchanged over the local network.

The use of the network key supposes the presence of an authorization server. The following question is “how does each node prove its legitimacy to the server and vice versa?”. The network key is a session key which is easier to manage over time than the master key as the authorization server can perform this task. Its generation involves the upper layers.

Generation of link keys between a pair of nodes needs more resource notably embedded asymmetric cryptography and the capability for the nodes to launch a key exchange protocol.

The use of cryptography to secure the frame masks the frame content. The routing algorithm should be “mesh-under” and based on the MAC address located in the frame header. When link keys are used, a “route-over” algorithm based on the IP address located in the network header can be employed because the frame content is decrypted and re-encrypted at each hop.

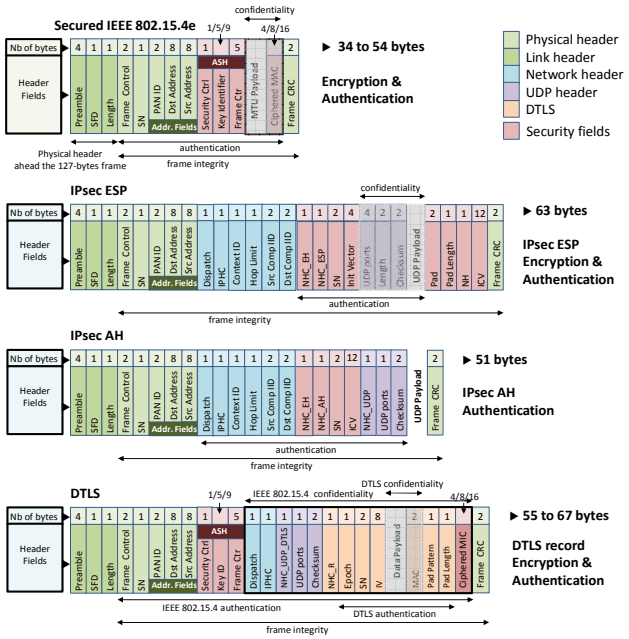

Fig. 28: Header of a secured packet over a LoWPAN

2) Security at network layer

IPsec offers several security modes integrated by the operating system (OS) of the node in the kernel. IPsec is based on previously negotiated session keys and a cipher suite using symmetric cryptographic functions. A default SA (Security Association) common for all nodes of the LoWPAN can be pre-defined.

IPv6 addressing is very powerful and offers many possibilities to carry a packet over a LoWPAN. First, it allows hiding the node identity by auto-configuration of the addresses randomly or cryptographically and by frequent renewal of the pseudonyms used. This functionality is a valuable tool to protect privacy at a significant cost of energy for a constrained device. Secondly, the link-local and multicast addressing requires an adequate key management scheme performed at the application level.

IPsec ESP is available with or without authentication. Several security faults were highlighted when the messages are not authenticated. The CRC is not cryptographically built and an attacker can forge a CRC to make the receiver accepted the packet at the link layer. Moreover, the use of IPsec ESP requires to decrypt and to re-encrypt the UPD header at each hop. This is costly for the node in computing operation.

With IPsec AH, IPv6 global addressing must be used enabling the use of “route-over” routing protocols at a minimal cost of 6 bytes on the network header (Fig. 28).

The IPsec protocol ensures the freshness of the packet but does not provide support for acknowledgment. Compressed

IPsec ESP and AH do not offer enough flexibility to be used over IEEE 802.15.4e. Research must focus on the development of a new version of lightweight IPsec dedicated to the LoWPAN.

3) Security at transport layer

At the application layer, DTLS attempts to provide solutions to establish and manage the session keys with a handshake mechanism and various compromises. The full handshake is very expensive for a LoWPAN and various studies focus on its simplification [24]. DTLS is also compatible with a RESTful interface and fully implementable into a WWW environment.

It provides confidentiality, authentication, integrity, freshness and acknowledgment over UDP frames with an overhead about 22 bytes. At the application layer, the security is easier to manage but the application developer needs to be aware about the security features and protocols to avoid pitfall at the development stage. The security provided by DTLS does not protect the headers of the lower layers and should be used in concordance with link layer security.

B. End-to-end security beyond the Border Router

The global addresses enable a hop-by-hop communication between two hosts of a Cyber Physical System. When one end point belongs to the WWW and the other to the LoWPAN world, a Border Router (BR) is in charge to translate the communication protocols to achieve the communication. But the security protocols translation implies the packet decryption and the re-encryption into a 127-bytes fragment, hence the term “hop-by-hop” instead of “end-to-end” security. In the future, a real end-to-end security should be achieved to guarantee the data security between two end-points and to facilitate the key management as the keys and credentials should be disclosed by the authorization server only to these two end-points. In this context, we will consider several configurations.

1) End-to-end security achieved at network layer

IPsec has been designed to secure a data flow between two hosts (Fig. 29) or a host and a network (Fig. 30).

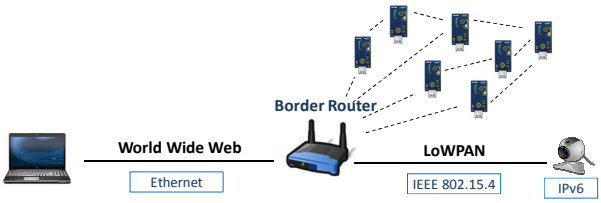

Fig. 29: Communication between two peers

The configuration shown on Fig. 29 can be envisaged if the LoWPAN end point is an IPv6 device not too constrained. The BR translates the global address of the source and destination expressed on 16 bytes in the WWW domain into a known context in the LoWPAN domain. The 6LoWPAN adaptation layer will fragment the incoming packet of at least 1280-bytes long to form fragments of 127-bytes long. This requires checking the MAC of the incoming IP packet and decrypting its content with the SA negotiated between the laptop and the BR. The IEEE 802.15.4 header is added to the fragmented 6LoWPAN packets holding a compressed IP header. The MAC is re-computed on each fragment and the content may be ciphered according to the SA negotiated between the BR and the LoWPAN device. The Fig. 28 details the header of a fragment in the LoWPAN world for a security achieved either with IPsec ESP or IPsec AH.

The fragmentation performed by the 6LoWPAN adaptation layer interrupts the end-to-end security. As consequence, the BR must be a trusted element. Compressed to maximum, the header takes 51 bytes for IPsec AH, letting 76 bytes per packet for the application protocol and data flow content. The confidentiality of the data can be ensured with a 12-bytes additional cost. This configuration enables a secure transportation of the data thanks to a trusted BR, but does not ensure the privacy.

The tunneling mode is not available in the LoWPAN domain, but it can be employed in the WWW domain. By this way (Fig. 30), the privacy is guaranteed between the laptop and the BR. In the LoWPAN side, stateless addressing may be employed to hide the device identity. As in the previous case (Fig. 29), the BR must be trusted as it performs the security protocols translation.

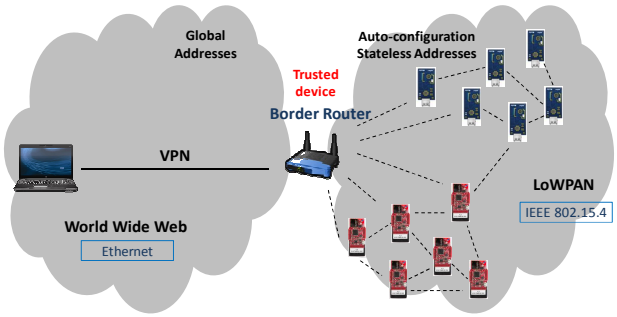

Fig. 30: Private communication between a host and a LoWPAN

2) End-to-end security achieved at application layer

DTLS is generally used with CoAP in a RESTful environment based on URI addressing. DTLS is not intended to carry large amount of data over the network, but to secure measures collected by the nodes and delivered to the BR, either spontaneously or in response to a query. DTLS is typically suitable for client-server architecture (Fig. 31).

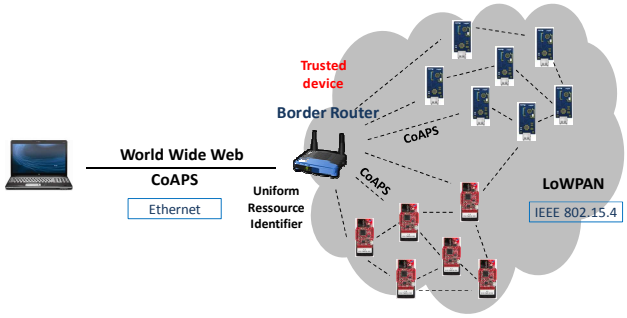

Fig. 31: Communication between a host and a LoWPAN via CoAP

Controlled at the application level, the laptop accesses the BR by a global addressing like URI that may also contain a request for a given node. The BR transmits the request from the WWW domain to the LoWPAN domain and delivers the response. While the security keys are established independently in each domain between the BR and the laptop and between the BR and the LoWPAN devices, the BR must remain a trusted element.

Many improvements may be envisaged to achieve end-to-end security from the laptop to the LoWPAN with any BR (untrusted). First, the session key must be negotiated between the two end-points, with the important issue questioning the node authorization in the LoWPAN. Secondly, the packets must be formed at the application level in the WWW as they will be inserted into a 6LoWPAN fragment using the application protocol used in the LoWPAN. A tentative to implement CoAP into a Web browser has been developed with Cooper as a Firefox Add-on. DTLS could be implemented into the Web browser for end-to-end security issues. Thirdly, compressed DTLS provides a lightweight version of DTLS reducing the packet size. The occupation of the bandwidth must also be reduced and a lightweight full handshake should be developed for constrained nodes. And, fourthly, the privacy problem will remain as DTLS does not cover this aspect.

VII. Conclusion

This paper presents a synthesis of the different ways to achieve the security of the communications in the IoT. It details notably the security for IEEE 802.15.4e in Time Synchronization Channel Hopping mode. As, the last version of IEEE 802.15.4 standard becomes mature, the working group 6TiSCH has been created to study the secure connection of an IEEE 802.15.4 meshed network over IPv6. The IPsec protocol suites, widely used to secure the traditional Internet, has been compressed and adapted to the LoWPAN. While the key establishment and cipher suite negotiation remain an issue, compressed IPsec provides features to ensure the source authentication and the data confidentiality with an additional cost for the message overhead. Datagram TLS emerges in the LoWPAN world and offers security tools at the application level. DTLS is actually heavy for constrained nodes and trade-off solutions are proposed in the literature. DTLS is scalable, compatible with a RESTful environment but authenticates only a few part of the message and does not protect the privacy. At the end of the paper, several end-to-end security use cases are studied to highlight how these security schemes could be used in a real system and what are the challenges to be addressed in the future work.

Acknowledgements

This research work was supported by the FP7 European projects BUTLER under contract no. 287901 and SocIoTal under contract no. 609112.

References

[1] Ovidiu Vermesan, Peter Friess, Gunter Woysch, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Markus Eisenhauer, Klaus Moessner, “Europe’s IoT Strategic Research Agenda 2012”, Chapter 2, CASAGRAS2, IERC 2012.

[2] Charith Perera, Arkady Zaslavsky, Peter Christen, Dimitrios Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey”, IEEE Communications Surveys & Tutorials, 2013.

[3] Ragunathan Rajkumar, Insup Lee, Lui Sha, Jonh Stankovik, “Cyber-Physical Systems: The Next Computing Revolution”, 47th ACM/IEEE Design Automation Conference, pp.731-736, 2010.

[4] Prasant Misra, Luca Mottola, Shahid Raza, Simon Duquennoy, Thiemo Voigt, “Supporting Cyber Physical Systems with Wireless Sensor Networks: An outlook of software and Services” Journal of the Indian Institute of Science, vol. 93, no. 3, 2013.

[5] “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, RFC4944, September 2007.

[6] Nurul Halimatul, Asmak Ismail, Rosilah Hassan, Khadijah W. M. Ghazali, “A study of Protocol tack in 6LoWPAN model”, Journal of Theoretical and Applied Information Technology, vol. 41, no.2, July 2012.

[7] Tobias Heer, Oscar Garcia-Morchon, René Hummen, Sye Loong Keoh, Sandeep S. Kumar, Klaus Wehrle, “Security Challenges in the IP-based Internet of Things”, Journal of Wireless Personal Communications, vol. 61, no. 3, pp. 527-542,December 2011.

[8] Shahid Raza, Simon Duquennoy, Tony Chung, Dogan Yazar, Thiemo Voigt, Utz Roedig, “Securing Communication in 6LoWPAN with Compressed IPsec,” Proceedings of the 7th IEEE International Conference on Distributing Computing in Sensor Systems, IEEE DCOSS, June 2011, Barcelona, Spain.

[9] IEEE802.15.4-2011, IEEE Standard for local and metropolitan area networks – 802.15.4: Low Rate Wireless Personal area Networks (LR-WPANs), September 2011.

[10] IEEE802.15.4e-2012, Amendment to IEEE 802.15.4-2011, IEEE Standard for local and metropolitan area networks – 802.15.4: Low Rate Wireless Personal area Networks, MAC sub-layer, April 2012.

[11] “IP Version 6 Addressing Architecture”, RFC4291, February 2006.

[12] “Deprecated Site Local Addresses”, RFC3879, September 2004.

[13] “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks”, RFC6282, September 2011.

[14] Z. Shelby, K. Hartke, C. Bormann, “Constrained application Protocol (CoAP)”, draft-ietf-core-coap-18, june 2013.

[15] “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals”, RFC4919, August 2007.

[16] “Security architecture for the Internet protocol”, RFC 2401, November 1998.

[17] “Internet Key Exchange Protocol Version 2 (IKEv2)”, RFC5996, September 2010.

[18] Shahid Raza, Thiemo Voigt, Vihelm Jutvik, “Lightweight IKEv2: A key management solution for both the compressed IPsec and the IEEE 802.15.4 security”, Workshop of SmartObject Security, Paris, France, 23 March 2012.

[19] Shahid Raza, Tony Chung, Simon Duquennoy, Dogan Yazar, Thiemo Voigt, Utz Roedig, “Securing Internet of Things with lightweight IPsec,” SICS technical Report, T2010:08, ISSN: 1100-3154, February 2011.

[20] Sarikaya Behcet, Xia Yangsong , “Lightweight secure neighbor discovery protocol for low-power and lossy networks”, Patent WO 2012044995 A1

[21] “Datagram Transport Layer Security Version 1.2”, RFC6347, September 2012.

[22] Shahid Raza, Hosein Shafagh, Kasun Hewage, René Hummen, Thiemo Voigt, “Lithe: Leightweight Secure CoAP for the Internet of Things,” IEEE Sensor Journal, vol. 13, no. 10, pp. 3711–3720, 2013, DIO: 10.1109/JSEN.2013.2277656, ISSN: 1530-437X.

[23] Shahid Raza, D. Trabalza, Thiemo Voigt, “6LoWPAN Compressed DTLS for CoAP”, IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 287-289, 2012.

[24] René Hummen, Jan H. Ziegeldorf, Hossein Shafagh, Shahid raza, Klaus Wehrle, "Towards Viable Certificate-based Authentication for the Internet of Things", 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec'2013, Budapest, Hungary, May 2013.

[25] D. McGrew, M. Pritikin, "The compressed X.509 Certificate Format", draft-pritikin-comp-x509-00, May 2010.
[26] René Hummen, Jens Hiller, Hanno Wirtz, Martin Henze, Hossein Shafagh, Klaus Wehrle, "6LoWPAN Fragmentation Attacks and Mitigation Mechanisms", ACM WiSec'2013, Budapest, Hungary.
[27] "Layer-2 security aspects for the IEEE 802.15.4e MAC", draft-piro-6tisch-security-issues-02, June 2014.
[28] S. Thomson and T. Narten,"IPv6 Address Autoconfiguration", RFC2462, December 1998.
[29] "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC3041, January 2001.
[30] "IPv6 Stateless Address Autoconfiguration", RFC4862, September 2007.
[31] "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC4941, September 2007.
[32] "Cryptographically Generated Addresses (CGA) ", RFC3972, March 2005.
[33] Naveen Sastry, David Wagner, "Security Considerations for IEEE 802.15.4 networks", Proceedings of the 3rd ACM Workshop on Wireless Security, pp. 32-34, DOI: 10.1145/1023646.1023654, ISBN:1-58113-925-X.
[34] Thomas Kothmayr, Corinna Schmitt, Wen Hu, Michael Brunig, Georg Carle, "A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication", IEEE 37th Conference on Local Computer Networks, pp. 956–963, 2012, DOI: 10.1109/ LCNW.2012.6424088.
[35] Matthias Kovatsch, Cooper (Cu), https://addons.mozilla.org/fr/firefox/addon/copper-270430/

Christine Hennebert is graduated from the engineering school ESIGELEC, France, in 1992. She obtained a PhD degree in 1996 from the Institut National Polytechnique de Grenoble, France. She was hired at the CEA Grenoble in 1997 as research engineer. She has worked on several scientific fields from signal and image processing to the implementation of wireless communication applications on innovative and heterogeneous hardware platforms. She is currently part of the team of the laboratory of the security of the connected objects and systems. Her main research interests are in the security and the privacy of the Internet of Things (IoT), the security of the wireless communications and the realization of IoT demonstrators embedding heterogeneous, wireless and autonomous devices with constrained resources. She has participated to the security deployment of the IEEE 802.15.4 network of the SmartCity of Santander in Spain.

Jessye Dos Santos is currently a PhD student in the laboratory of the security of connected objects and systems at CEA Grenoble, France. She was graduated from the engineering school Polytech'Grenoble in 2013 in Computer Sciences and Electronics. Her research interests are the security and the privacy of information and communication technologies.