Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues

Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva

***Abstract*—The Internet of Things (IoT) introduces a vision of a future Internet where users, computing systems, and everyday objects possessing sensing and actuating capabilities cooperate with unprecedented convenience and economical benefits. As with the current Internet architecture, IP-based communication protocols will play a key role in enabling the ubiquitous connectivity of devices in the context of IoT applications. Such communication technologies are being developed in line with the constraints of the sensing platforms likely to be employed by IoT applications, forming a communications stack able to provide the required power—efficiency, reliability, and Internet connectivity. As security will be a fundamental enabling factor of most IoT applications, mechanisms must also be designed to protect communications enabled by such technologies. This survey analyzes existing protocols and mechanisms to secure communications in the IoT, as well as open research issues. We analyze how existing approaches ensure fundamental security requirements and protect communications on the IoT, together with the open challenges and strategies for future research work in the area. This is, as far as our knowledge goes, the first survey with such goals.**

***Index Terms*—6LoWPAN, CoAP, DTLS, end-to-end security, IEEE 802.15.4, Internet of things, RPL, security.**

I. Introduction

The Internet of Things (IoT) is a widely used expression, although still a fuzzy one, mostly due to the large amount of concepts it encompasses. Connotations currently relating to the IoT include concepts such as Wireless Sensor Networks (WSN), Machine-to-Machine (M2M) communications and Low power Wireless Personal Area Networks (LoWPAN), or technologies such as Radio-Frequency Identification (RFID). The IoT materializes a vision of a future Internet where any object possessing computing and sensorial capabilities is able to communicate with other devices using Internet communication protocols, in the context of sensing applications. Many of such applications are expected to employ a large amount of sensing and actuating devices, and in consequence its cost will be an important factor. On the other hand, cost restrictions dictate constraints in terms of the resources available in sensing platforms, such as memory and computational power, while the unattended employment of many devices will also require the usage of batteries for energy storage. Overall, such factors motivate the design and adoption of communications and security mechanisms optimized for constrained sensing platforms, capable of providing its functionalities efficiently and reliably.

As the Internet communications infrastructure evolves to encompass sensing objects, appropriate mechanisms will be required to secure communications with such devices, in the context of future IoT applications, in areas as diverse as health-care (e.g. remote patient monitoring or monitoring of elderly people), smart grid, home automation (e.g. security, heating and lightning control) and smart cities (e.g. distributed pollution monitoring, smart lightning systems), among many others. After numerous research contributions in the recent past targeting low-energy wireless sensing applications and communication isolated from the outside world, a shift towards its integration with the Internet is taking place. This trend is also reflected in the efforts conducted by standardization bodies such as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF), towards the design of communication and security technologies for the IoT. Such technologies currently form a much necessary wireless communications protocol stack for the IoT that, together with the various communication technologies, is analyzed in detail in [1] and discussed later in the article. This stack is enabled by the technologies the industry believes to meet the important criteria of reliability, power-efficiency and Internet connectivity, and which may support Internet communications between constrained sensing devices or end-to-end communications with Internet devices outside of a local sensor network, thus laying the ground for the creation and deployment of new services and distributed applications encompassing both Internet and constrained sensing devices.

Throughout this survey we focus on security for communications on the IoT, analyzing both the solutions available in the context of the various IoT communication technologies, as well as those proposed in the literature. We also identify and discuss the open challenges and possible strategies for future research work in the area. As our focus is on standardized communication protocols for the IoT, our discussion is guided by the protocol stack enabled by the various IoT communication protocols available or currently being designed, and we also discuss cross-layer mechanisms and approaches whenever applicable. In our discussion we include works available both in published research proposals and in the form of currently active (at the time of writing of the article) Internet-Draft (I-D) documents submitted for discussion in relevant working groups. The security requirements targeted by the analyzed security protocols are identified in Table II, side-by-side with the provided functionalities.

Manuscript received July 22, 2013; revised February 21, 2014, June 5, 2014, and November 11, 2014; accepted December 28, 2014.

The authors are with University of Coimbra, 3000-370 Coimbra, Portugal (e-mail: jgranjal@dei.uc.pt; edmundo@dei.uc.pt; sasilva@dei.uc.pt).

Digital Object Identifier 10.1109/COMST.2015.2388550

This article analyzes the literature from 2003 to the present and is, as far as our knowledge goes, the first survey focusing on security for communications in the IoT. Other surveys do exist that, rather than analyzing the technologies currently being designed to enable Internet communications with sensing and actuating devices, focus on the identification of security requirements and on the discussion of approaches to the design of new security mechanisms [2], [3], or on the other end discuss the legal aspects surrounding the impact of the IoT on the security and privacy of its users [4].

Our discussion proceeds as follows. In Section II we identify the IoT communication protocols that are the focus of our discussion, together with the security requirements to consider for its employment. In Section III we discuss IoT communications and security at the physical and MAC layers, and in the following Sections the paper focuses on the technologies enabling end-to-end Internet communications involving sensing devices: 6LoWPAN at the network layer in Section IV, RPL routing in Section V and CoAP in Section VI. In Section VII we discuss research proposals on security mechanisms addressing open issues, as well as research challenges and opportunities for future work. Finally, in Section VIII we conclude the survey.

II. Communications and Security on the IoT

We proceed by identifying the protocols designed to support Internet communications with sensing devices in the IoT, which are the main focus of our analysis throughout the survey. In our following discussion we also discuss the security requirements that must be targeted by mechanisms designed to secure communications using such protocols.

A. *A Protocol Stack for the IoT*

Considering that the constraints of sensing platforms and the scale factors of the IoT typically make most of the communications and security solutions employed in the Internet ill suited for the IoT, working groups formed at standardization bodies as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF) are designing new communications and security protocols that will play a fundamental role in enabling future IoT applications. Such technological solutions are being designed in line with the constraints and characteristics of low-energy sensing devices and low—rate wireless communications. Although such characteristics have also influenced previous designs of applications employing Wireless Sensor Networks (WSN) isolated from the Internet and numerous research proposals on security mechanisms [5], the new standardized solutions are being designed to guarantee interoperability with existing Internet standards and guarantee that sensing devices are able to communicate with other Internet entities in the context of future IoT distributed applications.

The communication protocols available or being designed at the IEEE and IETF currently enable a standardized protocol stack discussed in [1] and illustrated in Fig. 1. The mechanisms forming this stack must thus enable Internet communications involving constrained sensing devices, while copying with the requirements of low-energy communications environments and the goals and the lifetime of IoT applications. From a bottom-up approach, the following are the main characteristics of the various protocols in this stack:

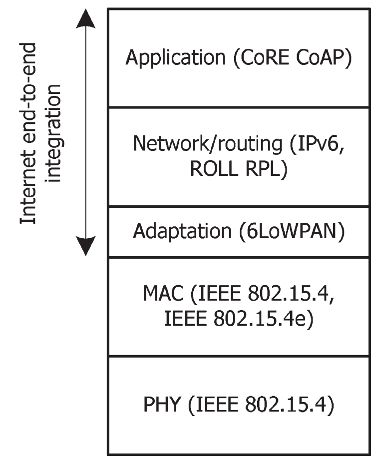

Fig. 1. Communication protocols in the IoT.

1) Low-energy communications at the physical (PHY) and Medium Access Control (MAC) layers are supported by IEEE 802.15.4 [6], [7]. IEEE 802.15.4 therefore sets the rules for communications at the lower layers of the stack and lays the ground for IoT communication protocols at higher layers.
2) Low-energy communication environments using IEEE 802.15.4 spare at most 102 bytes for the transmission of data at higher layers of the stack, a value much less than the maximum transmission unit (MTU) of 1280 bytes required for IPv6. The 6LoWPAN [8]–[10] adaptation layer addresses this aspect by enabling the transmission of IPv6 packets over IEEE 802.15.4. 6loWPAN also implements mechanisms for packet fragmentation and reassembly, among other functionalities.
3) Routing over 6LoWPAN environments is supported by the Routing Protocol for Low-power and Lossy Networks (RPL) [11]. Rather than being a routing protocol, RPL provides a framework that is adaptable to the requirements of particular IoT application domains. Application—specific profiles are already defined to identify the corresponding routing requirements and optimization goals.
4) The Constrained Application Protocol (CoAP) [12] supports communications at the application layer. This Protocol is currently being designed at the IETF to provide interoperability in conformance with the representational state transfer architecture of the web.

In this survey we identify and analyze the security protocols and mechanisms available to secure communications using the IoT technologies forming the stack illustrated in Fig. 1, together with the research proposals addressing open issues and opportunities for future work in the area. Given that the analyzed security solutions are designed in the context of the various IoT communications protocols, we also address its internal operation.

B. Security Requirements

The security mechanisms designed to protect communications with the previously discussed protocols must provide appropriate assurances in terms of *confidentiality*, *integrity*, *authentication* and *non-repudiation* of the information flows. Security of IoT communications may be addressed in the context of the communication protocol itself, or on the other end by external mechanisms, as we analyze throughout the article.

Other security requirements must also be considered for the IoT and in particular regarding communications with sensing devices. For example, WSN environments may be exposed to Internet-originated attacks such as Denial of Service (DoS), and in this context *availability* and *resilience* are important requirements. Mechanisms will also be required to implement protection against threats to the normal functioning of IoT communication protocols, an example of which may be fragmentation attacks at the 6LoWPAN adaptation layer. Other relevant security requirements are *privacy*, *anonymity*, *liability* and *trust*, which will be fundamental for the social acceptance of most of the future IoT applications employing Internet-integrated sensing devices. In the analysis throughout the article we identify how the various security requirements are verified by each security protocol and mechanism analyzed.

III. Security for IoT PHY and MAC Layer Communications

The IEEE produces standards to facilitate a common platform of rules for new technological developments. This is also the goal of the IEEE 802.15.4 standard [6], designed to support a healthy trade-off between energy-efficiency, range and data rate of communications. As illustrated in Fig. 1, the communications protocol stack for the IoT employs IEEE 802.15.4 with the goal of supporting low—energy communications at the physical (PHY) and Medium Access Control (MAC) layers.

IEEE 802.15.4 supports communications at 250 Kbit/s in a short-range of roundly 10 meters. The original IEEE 802.15.4 standard from 2006 was recently updated in 2011, mainly to include a discussion on the market applicability and practical deployments of the standard. Other amendments were introduced for the standard, namely IEEE 802.15.4a [13] specifying additional PHY layers, IEEE 802.15.4c [14] to support recently opened frequency bands in China and IEEE 802.15.4d [15] with a similar goal for Japan. Of particular interest for our discussion is IEEE 802.15.4e [7], an addendum defining modifications to the MAC layer with the goal of supporting time—synchronized multi-hop communications. Next we discuss how communications using IEEE 802.15.4 and IEEE 802.15.4e operate, and also the security services provided by the standard.

A. PHY Communications With IEEE 802.15.4

Due to its suitability to low-energy wireless communication environments, IEEE 802.15.4 lays the ground for the design of standardized technologies such as 6LoWPAN or CoAP at higher layers. IEEE 802.15.4 was also adopted in the recent past as the foundation of industrial WSN standards such as ZigBee-2006 [16], ZigBee PRO (2007) [17], ISA 100.11a [18] and WirelessHART [19]. Although such technologies provide proven industry solutions, they were not designed to support Internet communications with sensing devices. ZigBee defines application profiles targeting market areas such as home automation and smart energy, while WirelessHART and ISA (Wireless Systems for Automation) 100.11a target the industrial automation and control market. The IEEE 802.15.4e addendum to the standard was introduced in 2012 to enable support for the critical industrial applications supported by such industry standards, consequently opening the door for Internet communication protocols in the context of industrial applications in the future.

The IEEE 802.15.4 PHY manages the physical Radio Frequency (RF) transceiver of the sensing device, and also channel selection and energy and signal management. The standard supports 16 channels in the 2.4 GHz Industrial, Scientific and Medical (ISM) radio band. Reliability is introduced at the PHY by employing the Direct Sequence Spread Spectrum (DSS), Direct Sequence Ultra-Wideband (UWB) and Chirp Spread Spectrum (CSS) modulation techniques. DSSS was introduced in the original 2006 version of the standard, while UWB and CSS were added later in 2007 in the IEEE 802.15.4a addendum. The main goal of these modulation techniques is to achieve reliability by transforming the information being transmitted, so that it occupies more bandwidth at a lower spectral power density in order to achieve less interference along the frequency bands, together with an improved Signal to Noise (SNR) ratio at the receiver. PHY data frames occupy at most 128 bytes, and such packets are small in order to minimize the probability of errors taking place in low-energy wireless communication environments. In IEEE 802.15.4 security is available only at the MAC layer, as discussed next.

B. MAC Layer Communications With IEEE 802.15.4

The MAC layer manages, besides the data service, other operations, namely accesses to the physical channel, network beaconing, validation of frames, guaranteed time slots, node association and security. The standard distinguishes sensing devices by its capabilities and roles in the network. A full-function device (FFD) is able to coordinate a network of devices, while a Reduced-function device (RFD) is only able to communicate with other devices (of RFD or FFD types). By using RFD and FFD devices, IEEE 802.15.4 can support network topologies such as peer-to-peer, star and cluster networks. IEEE 802.15.4 devices may be identified using either a 16-bit short identifier or a 64-bit IEEE EUI-64 [20] identifier. Short identifiers are usually employed in restricted environments, while the 64-bit identifier is the IEEE EUI-64 identifier of the device. The 6LoWPAN adaptation layer analyzed later in the survey provides mechanisms to map standard Internet IPv6 addresses to 16-bit and 64-bit identifiers.

Regarding the formatting of data to be transmitted, the IEEE 802.15.4 standard defines four types of frames: data frames, acknowledgment frames, beacon frames and MAC command frames. Collisions during data communications are managed in the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access method or, in alternative, the coordinator may establish a super frame in the context of which applications with predefined bandwidth requirements may reserve and use one or more exclusive time slots. In this situation, beacon frames act as the limits of the super frame and provide synchronization to other devices, as well as configuration information.

C. Time-Synchronized Channel-Hopping MAC Layer Communications

Single-channel communications as enabled by the current version of the IEEE 802.15.4 standard may be unpredictable in terms of reliability, particularly in multi-hop usage scenarios, thus not being well suited to applications with restricted time constraints. As previously discussed, this is the case of applications in industrial environments currently supported by closed specifications such as WirelessHART and ISA 100.11a. With the goal of approaching this limitation, the recent IEEE 802.15.4e [7] addendum to the standard supports multi-hop communications using a technique originally proposed in the form of the Time Synchronized Mesh Protocol (TMSP) [21]. The TMSP protocol employs time synchronized frequency channel hopping to combat multipath fading and external interference, and is also the foundation of WirelessHART [19].

The mechanisms defined in IEEE 802.15.4e will be part of the next revision of the IEEE 802.15.4 standard, and as such opens the door for the usage of Internet communication technologies in the context of time—critical (e.g. industrial) applications. In IEEE 802.15.4e devices synchronize to a slot frame structure, a group of slots repeating over time. For each active slot, a schedule indicates with which neighbor a given device communicates with, and on which channel offset. Although IEEE 802.15.4e enables the definition of how the MAC layer executes a given schedule, it does not define how such a schedule is built.

IEEE 802.15.4e channel hopping also requires synchronization between devices, which may be acknowledgment-based or frame-based. In the former, the receiver calculates the difference between the expected time of arrival of the frame and its actual arrival, and provides this information to the sender in the corresponding acknowledgment, thus enabling the sender to synchronize its clock to the clock of the receiver. In the latter, the receiver adjusts its own clock by the same difference, thus synchronizing to the clock of the sender. IEEE 802.15.4e also introduces a few modifications to the security services provided at the MAC layer, as we discuss later.

D. Security in IEEE 802.15.4

The IEEE 802.15.4-2011 standard provides security services at the MAC layer that, despite being designed to secure communications at the link layer, are valuable in supporting security mechanisms designed at higher layers of the protocol stack illustrated in Fig. 1. This is motivated by the support of efficient symmetric cryptography at the hardware in IEEE 802.15.4 sensing platforms. For example, current sensing platforms employing the *cc2420* single-chip [22] RF transceiver from Texas Instruments, as the TelosB [23] mote from Crossbow, support IEEE 802.15.4 security and symmetric cryptography at the hardware using the Advanced Encryption Standard (AES) [24].

TABLE I
SECURITY MODES IN THE IEEE 802.15.4 STANDARD

Security mode	Security provided
No Security	Data is not encrypted Data authenticity is not validated
AES-CBC-MAC-32	Data is not encrypted Data authenticity using a 32-bit MIC
AES-CBC-MAC-64	Data is not encrypted Data authenticity using a 64-bit MIC
AES-CBC-MAC-128	Data is not encrypted Data authenticity using a 128-bit MIC
AES-CTR	Data is encrypted Data authenticity is not validated
AES-CCM-32	Data is encrypted Data authenticity using a 32-bit MIC
AES-CCM-64	Data is encrypted Data authenticity using a 64-bit MIC
AES-CCM-128	Data is encrypted Data authenticity using a 128-bit MIC

Security Modes: The IEEE 802.15.4 standard support various security modes at the MAC layer, which are described in Table I. The available security modes are distinguished by the security guarantees provided and by the size of the integrity data employed. Fig. 2 illustrates the application of security to an IEEE 802.15.4 link-layer data frame. A protected frame is identified by the *Security Enabled Bit* field of the *Frame Control* field being set at the beginning of the header. The *Auxiliary Security Header* is employed only when security is used, and identifies how security is applied to the frame. In the *Auxiliary Security Header*, the *Security Control* field identifies the *Security Level* mode from the modes identified in Table I, and how the cryptographic key required to process security for the link-layer frame is to be determined by the sender and receiver. The standard employs 128-bit keys that may be known implicitly by the two communication parties, or on the other end determined from information transported in the *Key Source* and *Key Index* subfields of the *Key Identifier* field. The *Key Source* subfield specifies the group key originator, and the *Key Index* subfield identifies a key from a specific source.

The various security modes require the transportation of security-related information in different configurations, as in Fig. 3. In our following discussion we identify how fundamental security requirements are assured by security at the MAC.

Confidentiality: Security as currently defined by IEEE 802.15.4 is optional, given that an application may opt for no security or for security at others layers of the protocol stack. For applications requiring only confidentiality of link-layer communications, the transmitted data may be encrypted using AES in the Counter (CTR) mode, using the AES-CTR security mode. As with all the security modes available at the IEEE 802.15.4 MAC layer, 128-bit keys are used to support this requirement.

Data Authenticity and Integrity: Applications requiring authenticity and integrity of link-layer communications may use one of the security modes employing AES in the Cypher Block Chaining (CBC) mode, which produces a Message Integrity Code (MIC) or Message Authentication Code (MAC) appended to the transmitted data. The security modes supporting this are AES-CBC-MAC-32, AES-CBC-MAC-64 and

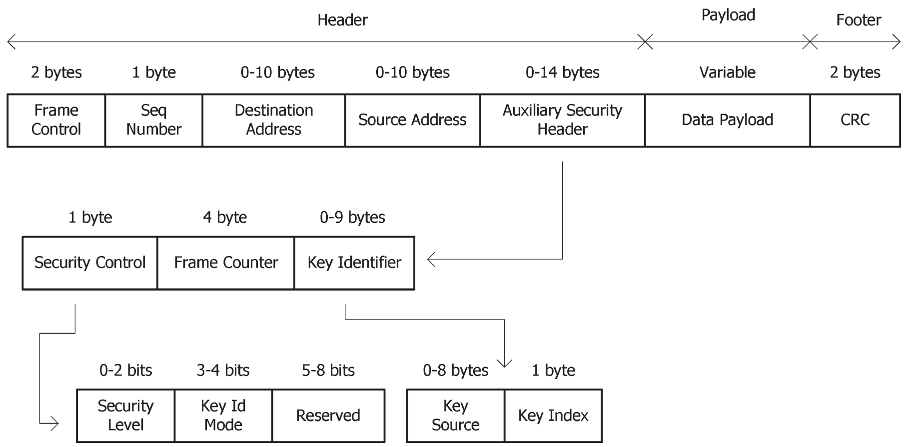

Fig. 2. Security data and control fields in IEEE 802.15.4.

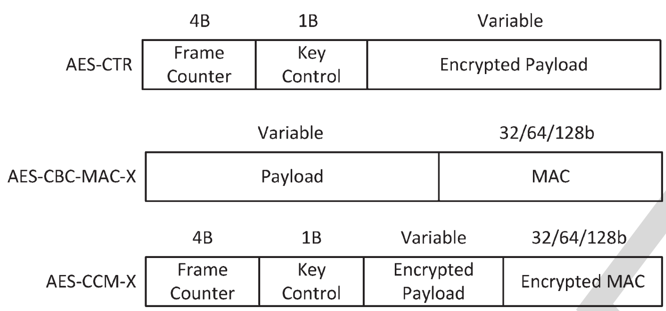

Fig. 3. Payload data formats with IEEE 802.15.4 security.

AES-CBC-MAC-128, which differ on the size of the integrity code produced. This code is created with information from the 802.15.4 MAC header plus the payload data, and in such security modes the payload is transmitted unencrypted.

Confidentiality, Data Authenticity and Integrity: The CTR and CBC modes may be jointly employed using the combined Counter with CBC-MAC AES/CCM encryption mode, which in IEEE 802.15.4 is used to support confidentiality as well as data authenticity and integrity for link-layer communications. This mode is supported in sensing platforms such as the TelosB in the CCM$*$ variant, which also offers provides for integrity-only and encryption-only security. This usage mode of AES provides confidentiality, message integrity and authenticity for data communications. The security modes are AES-CCM-32, AES-CCM-64 and AES-CCM-128, which again differ on the size of the MIC code following each message. AES-CCM modes require the transportation of all the security-related fields after the encrypted payload, as is illustrated in Fig. 3.

Semantic Security and Protection Against Message Replay Attacks: The *Frame Counter* and *Key Control* fields of the IEEE 802.15.4 Auxiliary Security Header may be set by the sender and provide support for semantic security and message replay protection in all the IEEE 802.15.4 security modes. The *Frame Counter* sets the unique message ID and the key counter (*Key Control* field) is under the control of the application, which may increment it if the maximum value for the *Frame Counter* is reached. The sender breaks the original packet into 16-byte blocks, with each block identified by its own block counter. In order to support semantic security and replay protection, each block is encrypted using a different nonce or Initialization Vector (IV).

As illustrated in Fig. 4, the Frame Counter and Key Counter fields, together with a static 1-byte Flags field, the sender's address and a 2-byte Block Counter field, constitute the IV. The Block Counter is not transmitted with the message, rather inferred by the receiver for each block. The IV is also employed for encryption using the security modes based on AES/CCM previously described.

Access Control Mechanisms: The IEEE 802.15.4 standard also provides access control functionalities, enabling a sensing device to use the source and destination addresses of the frame to search for information on the security mode and security-related information required to process security for the message. The 802.15.4 radio chips of the device stores an access control lists (ACL) with a maximum of 255 entries, each containing the information required for the processing of security for communications with a particular destination device. A default ACL entry may also be employed, defining how security is applied for packets not belonging to a more specific ACL entry. Fig. 5 illustrates the format of an ACL entry as defined in IEEE 802.15.4.

The ACL entry stores an IEEE 802.15.4 address, a *Security Suite* identifier field and the security material required to process security for communications with the device identified in the *Address* field. This security material consists of the cryptographic *Key* and, for suites supporting encryption, the *Nonce* (IV) that must be preserved across different packet encryption invocations. When replay protection is active, the ACL also stores a high water mark of the most recently received packet's identifier in the *Replay Counter* field.

Security With Time-Synchronized Communications: As previously discussed, the IEEE 802.15.4e [7] addendum introduces time-synchronized channel-hopping communications, and also adapts security accordingly. IEEE 802.15.4e adapts replay protection and semantic security to time-synchronized network communications, as supported by the addendum. The addendum defines the possibility of using null or 5-byte *Frame Counter* values, which in the latter case shall be set to the global Absolute Slot Number (ASN) of the network. The ASN stores

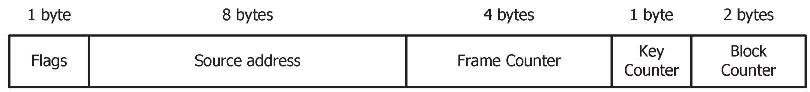

Fig. 4. Format of the Initialization Vector for AES-CRT and AES-CCM security in IEEE 802.15.4.

Fig. 5. Format of an ACL entry in IEEE 802.15.4.

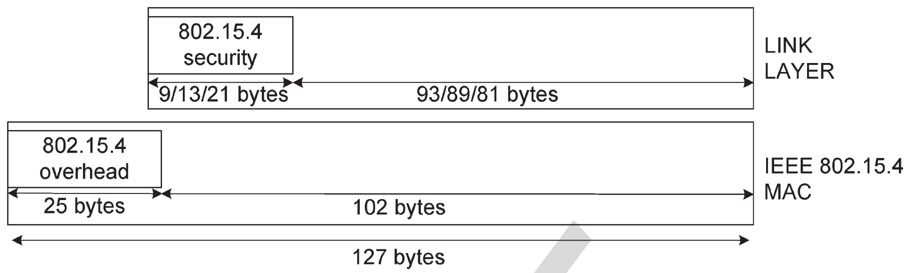

Fig. 6. Payload space availability with IEEE 802.15.4.

the total number of timeslots that have elapsed since the start of the network and is beaconed by devices already in the network, allowing new devices to synchronize.

The usage of the ASN as a global frame counter value enables time-dependent security, replay protection and semantic security. To enable the usage of a 5-byte *Frame Counter* value, IEEE 802.15.4e introduces modifications to the *Security Control* field illustrated in Fig. 2 which, in addition to the *Security Level* and the *Key Identifier Mode* fields, now employs two bits from the reserved space: bit 5 to enable suppression of the *Frame Counter* field and bit 6 to distinguish between a *Frame Counter* field occupying 4 or 5 bytes. In consequence, the *Auxiliary Security Header* illustrated in Fig. 2 may now transport a null, a 4-byte or a 5-byte *Frame Counter* field. The CCM∗ IV for AES encryption may now contain a 5-byte *Frame Counter*, instead of a 4-byte *Frame Counter* followed by a 1-byte *Key Control* as illustrated in Fig. 4. Other than the previously described modifications, the remaining security services provided by the IEEE 802.15.4 base specification still apply to applications employing IEEE 802.15.4e. Later in Section VII we address the limitations of the security mechanisms previously described in providing effective protection of communications in the IoT, and we also identify how such limitations can be addressed either with new research proposals or in future versions on the standard.

IV. Security for IoT Network-Layer Communications

One fundamental characteristic of the Internet architecture is that it enables packets to traverse interconnected networks using heterogeneous link-layer technologies, and the mechanisms and adaptations required to transport IP packets over particular link-layer technologies are defined in appropriate specifications. With a similar goal, the IETF IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) working group was formed in 2007 to produce a specification enabling the transportation of IPv6 packets over low-energy IEEE 802.15.4 and similar wireless communication environments.

6LoWPAN is currently a key technology to support Internet communications in the IoT, and one that has changed a previous perception of IPv6 as being impractical for constrained low-energy wireless communication environments. The 6LoWPAN adaptation layer materializes a good example of how cross-layer mechanisms and optimizations may enable standardized communication protocols for the IoT, and enables IPv6 end-to-end communications between constrained IoT sensing devices and other similar or more powerful Internet entities, thus providing the required support for the building of future IPv6-based distributed sensing applications on the IoT. The 6LoWPAN adaptation layer maps the services required by the IP layer on the services provided by the IEEE 802.15.4 MAC layer. The characteristics of IEEE 802.15.4 previously discussed strongly determine the usage of very-optimized adaptation mechanisms at the adaptation layer, as we proceed to discuss.

A. 6LoWPAN Frame Format and Header Compression

As illustrated in Fig. 1 and previously discussed, IEEE 802.15.4 supports PHY and MAC layer communications, which enable the transportation of data from communication protocols at higher layers of the stack. In the absence of link-layer security, the data payload for protocols at higher layers of the stack is limited to 102 bytes, as illustrated in Fig. 6.

The 6LoWPAN adaptation layer optimizes the usage of this limited payload space through packet header compression, while also defining mechanisms for the support of operations required in IPv6, in particular neighbor discovery and address auto-configuration. The adaptation layer is defined in various RFC (Request for Comments) documents, as we proceed to discuss. RFC 4919 [8] discusses the general goals and assumptions of the work performed in the IETF 6LoWPAN working group. RFC 4944 [9] defines the mechanisms for the transmission of IPv6 packets over IEEE 802.15.4 networks, with header compression being defined in RFC 6282 [10]. Header compression is performed with information from the link and adaptation layers, which is used to jointly compress network and transport protocol headers. RFC 6282 [10] specifies how User Datagram Protocol (UDP) headers may be compressed in the context of the 6LoWPAN adaptation layer. Other relevant documents are RFC 6568 [25] discussing design and application spaces for 6LoWPAN, RFC 6606 [26] discussing the main requirements for 6LoWPAN routing, and RFC 6775 [27] defining optimizations for Neighbor Discovery.

All 6LoWPAN encapsulated datagrams transported over IEEE 802.15.4 MAC frames are prefixed by a stack of 6LoWPAN headers. A *type* field occupying the first two bits of the header identifies each 6LoWPAN header, and the standard currently defines the following four header types:

- *No 6LoWPAN*: indicates that a given packet is not for 6LoWPAN processing, thus enabling the coexistence with devices not supporting 6LoWPAN.
- *Dispatch*: supports IPv6 header compression and link-layer multicast and broadcast communications.
- *Mesh addressing*: supports forwarding of IEEE 802.15.4 frames at the link-layer, as required for the formation of multi-hop networks.
- *Fragmentation*: supports fragmentation and reassembly mechanisms required to transmit IPv6 datagrams over IEEE 802.15.4 networks.

The presence of each 6LoWPAN header is optional, and headers must appear in a particular order, starting from the *mesh addressing*, and next the *broadcast*, *fragmentation* and *dispatch* headers. The *dispatch* header identifies the compression method applied to a given packet:

- LOWPAN_HC1 was the original compression scheme defined in RFC 4944 [9], supporting compression of link-local IPv6 addresses only. This scheme doesn't support compression of global IPv6 addresses, thus being suboptimal for IoT applications.
- LOWPAN_HC1g and LOWPAN_HC2 [28] provided an initial approach to compress global IPv6 addresses and UDP headers, respectively. LOWPAN_HC1g assumes that a given network of IoT devices is assigned a compressible 64-bit global IPv6 prefix.
- LOWPAN_IPHC is defined in RFC 6282 [10] and replaces the previous methods with compression based on shared states. This scheme may compress link-local addresses and also global and multicast IPv6 addresses. RFC 6282 also defines the LOWPAN_NHC scheme to compress IPv6 next headers and how UDP header compression may be accomplished. For compatibility with the previous implementations, networking stacks supporting 6LoWPAN must also process packet decompression using the previous LOWPAN_HC1 scheme.

We may observe the importance of 6LoWPAN as a convergence technology supporting an increasingly growing ecosystem of PHY/MAC communications technologies optimized for particular communication environments and applications. Proposals have been submitted for the support in 6LoWPAN of communications using Bluetooth Low Energy (BLE) [29], Digital Enhanced Cordless Telecommunications Ultra Low Energy (DECT-ULE) [30], ITU-T G. 9959 [31] and Near Field Communications (NFC) [32]. Very constrained devices such as RFID may currently employ different communication and security approaches [33], but can also evolve to support Internet communications in the future.

B. Security in 6LoWPAN

No security mechanisms are currently defined in the context of the 6LoWPAN adaptation layer, but the relevant documents include discussions on the security vulnerabilities, requirements and approaches to consider for the usage of network-layer security, as we proceed to discuss. Later in Section VII we analyze research proposals on approaches to 6LoWPAN security, as well as the open research challenges and opportunities.

Identification of Security Vulnerabilities: The discussion regarding security on RFC 4944 [9] is related to the possibility of forging or accidentally duplicating EUI-64 interface addresses, which may consequently compromise the global uniqueness of 6LoWPAN interface identifiers. This document also discusses that Neighbor Discovery and mesh routing mechanisms on IEEE 802.15.4 environments may be susceptible to security threats, and that AES security at the link-layer may provide a basis for the development of mechanisms protecting against such threats, particularly for very constrained devices. Other interesting discussion is on the possibility of employing more powerful 6LoWPAN devices in order to support heavy security-related operations, also because such devices may support existing Internet security protocols, as such representing strategic places for the enforcement of security control mechanisms.

The discussion concerning security on RFC 6282 [10] focuses on the security issues posed by the usage of a mechanism inherited from RFC 4944, which enables the compression of a particular range of 16 UDP port numbers down to 4 bits. This document discusses that the overload of ports in this range, if employed with applications not honoring the reserved set for port compression, may increase the risk of an application getting the wrong type of payload or of an application misinterpreting the content of a message. As a result, RFC 6282 recommends that the usage of such ports be associated with a security mechanism employing MIC codes.

Identification of Security Requirements and Strategies: The informational RFC 4919 [8] discusses the addressing of security at various complementary protocol layers of the stack illustrated in Fig. 1, considering that the most appropriate approach may depend on the application requirements and on the constraints of particular sensing devices. This document also identifies the possibility of employing security at the network-layer using IPSec, together with the interest in investigating its applicability in the transport and tunnel usage modes.

The discussion on security in RFC 6568 [25] focuses on the possible approaches to adopt security in the light of the characteristics and constraints of wireless sensing devices. This document discusses threats due to the physical exposure of such devices, which may pose serious demands for its resiliency and survivability. It also discusses how IEEE 802.15.4 communications may facilitate attacks against the confidentiality, integrity, authenticity and availability of 6LoWPAN devices and communications.

Rather than providing a specific approach to routing in 6LoWPAN environments, RFC 6606 [26] provides guidelines that are useful in designing specific routing approaches. As with the previous standard documents, RFC 6606 identifies the importance of addressing security and the usefulness of AES/CCM available at the hardware of IEEE 802.15.4 sensing platforms. This document also discusses the importance of designing security mechanisms that are able to adapt to changes in the network topology and devices, rather than employing a static security configuration, given that many 6LoWPAN applications may employ networks that are dynamic in such respects. This document also discusses the importance of time synchronization, self-organization and security localization in providing security for data and multi-hop routing control packets. Other important security requirements identified are the support of authenticated broadcasts and multicasts, and the verification of bidirectional links.

RFC 6775 [27] focuses on optimizations to enable Neighbor Discovery (ND) operations in 6LoWPAN environments, and also on the application of the threat model for ND operations defined in RFC 4861 [34] to 6LoWPAN environments. Other possibilities discussed in this document consists in the adaptation of the SEcure Neighbor Discovery (SEND) [35] and cryptographically generated addresses [36] mechanisms to 6LoWPAN environments.

V. Security for Routing in the IoT

The Routing Over Low-power and Lossy Networks (ROLL) working group of the IETF was formed with the goal of designing routing solutions for IoT applications. The current approach to routing in 6LoWPAN environments is materialized in the Routing Protocol for Low power and Lossy Networks (RPL) [11] Protocol. Rather than providing a generic approach to routing, RPL provides in reality a framework that is adaptable to the requirements of particular classes of applications. In the following discussion we analyze the internal operation of RPL, and later the security mechanisms designed to protect communications in the context of routing operations.

A. Routing With RPL

The adoption of appropriate routing strategies in 6LoWPAN environments is a very challenging task, mostly due to the inherent specificities of each application and of the constraints of the sensing devices employed. In consequence, RPL assumes that routing must adapt to the requirements of particular application areas and, for each application area, an appropriate RFC documents an objective function that maps the optimization requirements of the target scenario. Requirements for application areas are currently defined in RFC 5548 [37] for urban low-power applications, in RFC 5673 [38] for industrial applications, in RFC 5826 [39] for home automation applications and in RFC 5867 [40] for building automation applications. RPL also employs metrics that are appropriate to 6LoWPAN environments, such as those defined in RFC 6551 [41].

Considering that in the most typical setting various LoWPAN nodes are connected through multi-hop paths to a small set of root devices responsible for data collection and coordination, RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) identified by a DODAGID for each root device, by accounting for link costs, node attributes, note status information, and its respective objective function. The topology is set up based on a rank metric, which encodes the distance of each node with respect to its reference root, as specified by the objective function. According to the gradient-based approach, the rank should monotonically decrease along the DODAG and towards the destination node.

The simplest RPL routing topology is constituted by a single DODAG containing just one root, although more complex scenarios are possible. Multiple instances of RPL may run concurrently on the network, each with different optimization objectives, as traduced by the correspondent objective function. RPL is designed to support three fundamental traffic topologies: Multipoint-to-Point (MP2P), Point-to-Multipoint (P2MP) and Point-to-Point (P2P). MP2P traffic is routed towards nodes supporting the DODAG root role and possibly gateway functions with the Internet or other external IP networks. P2MP can be used for networks requiring the usage of actuating devices, in addition to sensors. P2P employs a packet flowing from the source towards the common ancestor of the two communicating devices and then downward to the destination device. These three topologies require RPL to discover both upward routes to support MP2P and P2P traffic, and downward routes to support P2P and P2MP traffic. Tree-based topologies also map well with time-synchronized schedule-based MAC communications using IEEE 802.15.4e.

The RPL protocol supports various types of control messages, particularly DIO (DODAG Information Object), DIS (DODAG Information Solicitation), DAO (Destination Advertisement Object), DAO-ACK (DAO acknowledgment) and CC (Consistency Check) messages. A node transmits DIO messages containing information required for other nodes to compute their own rank, to join an existing DODAG and to select a set of parents and the preferred parent in that DODAG among all possible neighbors. DIO messages may be requested by sending a message of type DIS (DODAG Information Solicitation). DIO and DIS messages are employed for the establishment of routes upward in the RPL routing tree, while downward paths are established by having DAO messages to back-propagate routing information from leaf nodes to the roots. A DAO message is triggered by the reception of a DIO message, and its recipient may send a DAO-ACK message to a DAO parent or to the DODAG root. CC messages are used for synchronization of counter values among communicating nodes and provide a basis for the protection against packet replay attacks. All RPL control messages are encapsulated in ICMPv6 packets [42] and are identified by an ICMPv6 type of 155.

The current RPL specification recognizes the importance of supporting mechanisms to secure routing messages exchanged between sensing devices and, in consequence, RPL defines secure versions of the various routing control messages previously discussed, as well as three security modes, as we discuss next.

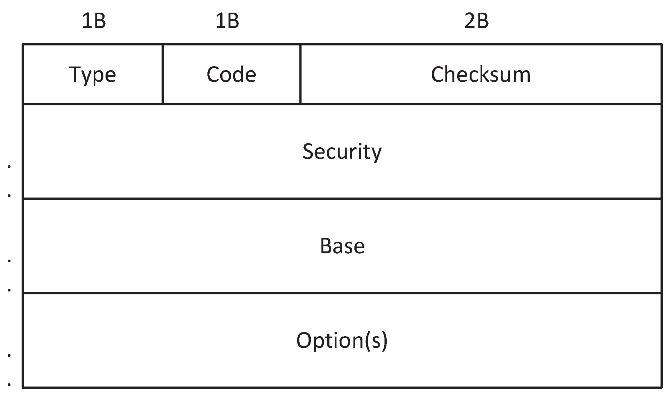

Fig. 7. Secure RPL control message.

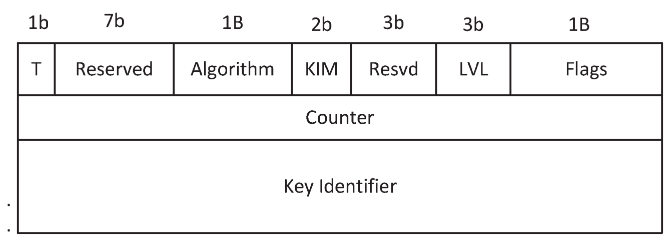

Fig. 8. Security section of a secure RPL control message.

B. Security in RPL

The RPL specification [11] defines secure versions of the various routing control messages, as well as three basic security modes. In Fig. 7 we illustrate the format of a secure RPL control message, transporting a *Security* field after the 4-byte ICMPv6 message header. The high order bit of the RPL *Code* field identifies whether or not security is applied to a given RPL message, which may thus be a secure DIS, DIO, DAO or DAO-ACK message. The format of the *Security* field is illustrated in Fig. 8.

The information in the *Security* field indicates the level of security and the cryptographic algorithms employed to process security for the message. What this field doesn't include is the security—related data required to process security for the message, for example a Message Integrity Code (MIC) code or a signature. Instead, the security transformation itself states how the cryptographic fields should be employed in the context of the protected message.

Support of Integrity and Data Authenticity: The current RPL specification [11] defines the employment of AES/CCM with 128-bit keys for MAC generation supporting integrity, and of RSA with SHA-256 for digital signatures supporting integrity and data authenticity. The *LVL* (Security Level) field indicates the provided packet security and allows for varying levels of data authentication and, optionally, of data confidentiality. RFC 6550 also defines various values to identify the presence of confidentiality, integrity and data authenticity with MAC-32 and MAC-64 authentication codes, as well as of 2048 and 3072-bit signatures using RSA.

Support of Semantic Security and Protection Against Replay Attacks: A Consistency Check (CC) control message enables a sensing node to issue a challenge-response with the goal of validating another node's current counter value, for example in situations when a received message has an initialized (zero value) counter value and the receiver has an incoming counter currently maintained for the message originator. In this case the receiver initiates counter resynchronization by sending a CC message to the message source. Semantic security and protection against packet replay attacks is provided with the help of the *Counter* field, which may be used to transport a timestamp, as indicated by the T in Fig. 8. The next byte in the *Security* section of the RPL control message identifies the security suite employed to provide security, while the *Flags* field is currently reserved.

Support of Confidentiality: The secure variant of the various RPL control messages may also support confidentiality and delay protection. Regarding the employment of cryptographic algorithms in RPL, AES/CCM is adopted as the basis to support security in the current specification [11], while we note that other algorithms may be adopted in the future and appropriately identified in the *Security* section of a secure RPL control message. RPL control messages may be protected using both an integrated encryption and authentication suite, such as with AES/CCM, as well as schemes employing separate algorithms for encryption and authentication.

The entire RPL message is within the scope of RPL security. MAC codes and signatures are calculated over the entire unsecured IPv6 packet, with the mutable fields of the packet zeroed. When a RPL ICMPv6 message is encrypted, encryption starts at the first byte after the *Security* section and continues to the last byte of the packet. The IPv6 header, the ICMPv6 header and the RPL message, up to the start of the *Security* field, are not encrypted, since those fields are required to correctly decrypt the packet.

Support for Key Management: The *KIM* (Key Identifier Mode) field of the *Security* section illustrated in Fig. 8 indicates whether the cryptographic key required to process security for this message may be determined implicitly or explicitly. RFC 6550 [11] currently defines different values for this field to thus supports different key management approaches, namely group keys, keys per pair of sensing devices, and digital signatures. This field supports various levels of granularity of packet protection, and is divided in a *key source* and *key index* subfields. The *key source* subfield indicates the logical identifier of the originator of a group key, while the *key index* subfield, when present, allows unique identification of keys with the same originator.

Security Modes in RPL: As previously discussed, RPL defines how security is applied to routing control messages, and the current specification also defines the following three security modes:

- *Unsecured*: in this mode no security is applied to routing control messages, and this is the default usage mode of RPL.
- *Preinstalled*: this security mode may be employed by a device using a preconfigured symmetric key in order to join an existent RPL instance, either as a host or a router. This key is employed to support confidentiality, integrity and data authentication for routing control messages.

- *Authenticated*: this security mode is appropriate for devices operating as routers. A device may initially join the network using a preconfigured key and the *preinstalled* security mode, and next obtain a different cryptographic key from a key authority with which it may start functioning as a router. The key authority is responsible for authenticating and authorizing the device for this purpose.

The RPL specification [11] currently defines that the *authenticated* security mode must not be supported by symmetric cryptography, although it doesn't specify how asymmetric cryptography may be employed to support node authentication and key retrieval by the device intending to operate as a router. A more clear definition of such mechanisms is thus required, and future versions of the RPL standard may more clearly define how to support them.

While not introducing additional security mechanisms, other documents relevant to RPL also include analysis on security aspects. This is the case of the informational RFC documents discussing routing requirements for the various application areas [37]–[40]. Such documents discuss the importance of protecting routing control messages with appropriate confidentiality, authentication and integrity. RFC 6551 [41] specifies a set of link and node routing metrics appropriate to the characteristics and constraints of 6LoWPAN environments, and discusses the necessity of handling such metrics in a secure and trustful manner, including protection against nodes being able to falsify or lie in the advertisement of metrics, as a way to protect against attacks on routing operations.

VI. Security for IoT Application-Layer Communications

As previously discussed, application-layer communications are supported by the CoAP [12] protocol, currently being designed by the Constrained RESTful Environments (CoRE) working group of the IETF. We next discuss the operation of the protocol as well as the mechanisms available to apply security to CoAP communications.

A. Application-Layer Communications With CoAP

The CoAP [12] protocol implements a set of techniques to compress application-layer protocol metadata without compromising application inter-operability, in conformance with the representational state transfer (REST) architecture of the web. CoAP is currently defined only for UDP communications over 6LoWPAN, although the adoption of transport-layer approaches with characteristics more close to protocols such as the Transmission Control Protocol (TCP) [43] is still open to debate, with ongoing research addressing the adaptation of TCP for 6LoWPAN environments [44].

Application-layer communications may enable IoT sensing applications to interoperate with existing Internet applications without requiring specialized application oriented code or translation mechanisms. CoAP restricts the HTTP dialect to a subset that is well suited to the constraints of 6LoWPAN sensing devices, and may enable abstracted communications

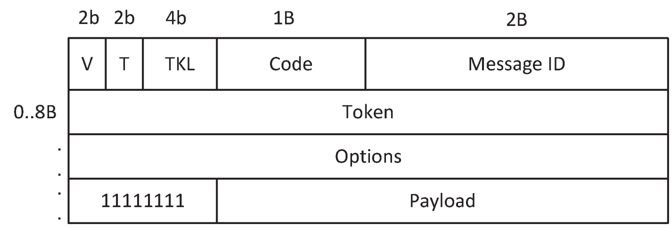

Fig. 9. Format of a CoAP message header.

between users, applications and such devices, in the context of IoT applications. The CoAP protocol provides a request and response communications model between application endpoints and enables the usage of key concepts of the web, namely the usage of URI addresses to identify the resources available on constrained sensing devices. The protocol may support end-to-end communications at the application-layer between constrained IoT sensing devices and other Internet entities, using only CoAP or in alternative by translating HTTP to CoAP at a reverse or forward gateway.

Messages in the CoAP protocol are exchange asynchronously between two endpoints, and used to transport CoAP requests and responses. Since such messages are transported over unreliable UDP communications, CoAP provides a lightweight reliability mechanism. Using this mechanism CoAP messages may be marked as *Confirmable*, for which the sender activates a simple stop-and-wait retransmission mechanism with exponential backoff. The receiver must acknowledge a *Confirmable* message with a corresponding *Acknowledge* message or, if it lacks context to process the message properly, reject it with a *Reset* message. *Acknowledge* or *Reset* messages are related to a *Confirmable* message by means of a Message ID, along with the address of the corresponding endpoint. CoAP messages may also be transmitted less reliably if marked as *Non-Confirmable*, in which case the recipient does not acknowledge the message. Similarly to HTTP, CoAP defines a set of method and response codes available to applications.

Other than a basic set of information, most of the information in CoAP is transported using options. Options defined for the CoAP Protocol may be critical, elective, safe or unsafe. A critical option is one that an endpoint must understand, while an elective option may be ignored by an endpoint not recognizing it. Safe and unsafe options determine how an option may be processed by an intermediary entity. An unsafe option needs to be understood by the proxy in order to be forwarded, while a safe option may be forwarded even if the proxy is unable to process it.

The CoAP header and message format is illustrated in Fig. 9. The message starts with a 4-byte fixed header, formed by the *Version* field (2 bits), the T (message type) field (2 bits), the *TKL* (Token Length) field (4 bits), the *Code* field (8 bits) and the *Message ID* (16 bits). The token in practice enables a CoAP entity to perform matching of CoAP requests and replies, while the message ID supports duplicate detection and optional reliability.

The options adopted in CoAP are defined in the Type-length-value (TLV) format, by specifying its option number followed by its length and value. CoAP currently defines the *Uri-Host*,

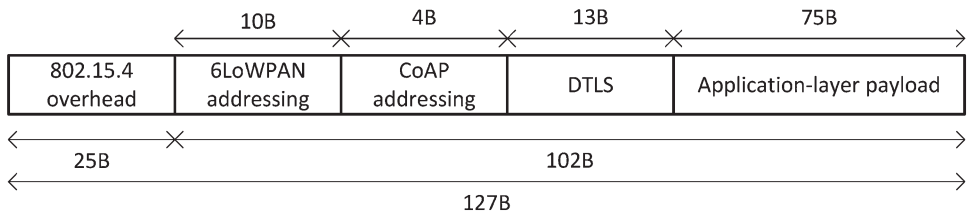

Fig. 10. Payload space with DTLS on 6LoWPAN environments.

Uri-Port, *Uri-Path* and *Uri-Query* options enabling the identification of the target resource of a request, *Content-Format* to specify the representation format of the message payload, and *Max-Age* to indicate the maximum time a CoAP response may be cached before being considered not fresh, among others [12]. Regarding security, rather than designing mechanisms to support (object) security directly in the context of application-layer communications, CoAP adopts DTLS at the transport-layer to transparently apply security to all CoAP messages in a given communications session. The protocol also defines four security modes, as we analyze next.

B. Security in CoAP

The CoAP Protocol [12] defines bindings to DTLS (Datagram Transport-Layer Security) [45] to secure CoAP messages, along with a few mandatory minimal configurations appropriate for constrained environments.

Support for Confidentiality, Authentication, Integrity, Non-Repudiation and Protection Against Replay Attacks: The adoption of DTLS implies that security is supported at the transport-layer, rather than being designed in the context of the application-layer protocol. DTLS provides guarantees in terms of confidentiality, integrity, authentication and non-repudiation for application-layer communications using CoAP. DTLS is in practice TLS [46] with added features to deal with the unreliable nature of UDP communications. Fig. 10 illustrates the availability of payload space for applications in IEEE 802.15.4 and 6LoPWAN communication environments in the presence of CoAP and DTLS.

Once the initial DTLS handshake is completed, DTLS adds a limited per-datagram overhead of 13 bytes, not counting any initialization vectors, integrity check values or the padding that may be required by the cipher suite employed. As considered in Fig. 10, shared-context 6LoWPAN header compression requires 10 bytes for an UDP/IPv6 header, while the CoAP fixed header requires 4 bytes. The impact of DTLS on constrained wireless sensing devices is due to the cost of supporting the initial handshake plus the processing of security for each exchanged CoAP messages. The impact of DTLS on constrained wireless sensing devices is due to the cost of supporting the initial handshake plus the processing of security for each exchanged CoAP messages. Similarly to other approaches to security in 6LoWPAN environments, AES/CCM is adopted as the cryptographic algorithm to support fundamental security requirements in the current CoAP [12] specification. Security against replay attacks may also be achieved in the context of DTLS, using a different nonce value for each secured CoAP packet.

Security Modes in CoAP: In addition to the adoption of DTLS, CoAP currently defines four security modes that applications may employ. Those security modes essentially differ on how authentication and key negotiation is performed, as follows:

- *NoSec*: this mode in practice provides no security, and CoAP messages are transmitted without security applied.
- *PreSharedKey*: this security mode may be employed by sensing devices that are pre-programmed with the symmetric cryptographic keys required to support secure communications with other devices or groups of devices. This mode may be appropriate to applications employing devices that are unable to support public-key cryptography, or for which it is convenient to employ security pre-configuration. Applications may use one key per destination device or in alternative a single key for a group of destination devices.
- *RawPublicKey*: this security mode is appropriate for devices requiring authentication based on public keys, but which are unable to participate in public-key infrastructures. A given device must be preprogrammed with an asymmetric key pair that may be validated using an out-of-band mechanism [47] and possibly programmed as part of the manufacturing process, while without a certificate. The device has an identity calculated from its public key and a list of identities and public keys of the nodes it can communicate with. This security mode is defined as mandatory to implement in CoAP.
- *Certificates*: this security mode also supports authentication based on public-keys, but for applications that are able to participate in a certification chain for certificate validation purposes. This security mode thus assumes the availability and usage of a security infrastructure. The device has an asymmetric key pair with an X.509 certificate that binds it to its Authority Name and is signed by some common trusted root. The device also has a list of root trust anchors that can be used for certificate validation.

An important aspect of CoAP security using DTLS is that Elliptic Curve Cryptography (ECC) [48] is adopted to support the *RawPublicKey* and *Certificates* security modes. ECC supports device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA), and also key agreement using the ECC Diffie-Hellman counterpart, the Elliptic Curve Diffie-Hellman Algorithm with Ephemeral keys (ECDHE). The *NoSec* security mode corresponds to a device sending packets without security, using the “coap” scheme in URI addresses identifying resources available on CoAP servers. On the other end, accesses to resources with DTLS use the “coaps” scheme, and in this case a security association at the transport-layer using DTLS must exist between the CoAP client and the CoAP server.

The current CoAP specification defines a mandatory-to-implement cipher suite for each security mode, based on the usage of AES/CCM and ECC cryptographic operations, as follows:

- Applications supporting the *PreSharedKey* security mode are required to support at least the TLS_PSK_WITH_AES_128_CCM_8 [49] suite, which supports authentication using pre-shared symmetric keys and 8-byte nonce values, and encrypts and produces 8-byte integrity codes.
- Applications supporting the *RawPublicKey* CoAP security mode are required to support the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 [46], [50] security suite using ECDSA-capable public keys. This security mode also employs SHA-256 to compute hashes.
- Applications supporting the *Certificates* security mode are also required to support the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 cipher suite. Regarding the usage of public-keys transported in X.509 certificates, the *SubjectPublicKeyInfo* field in a X.509 certificate defines how the corresponding public key must be employed for ECC computations. The certificate must also contain a signature created using ECDSA and SHA-256. Applications using devices with a shared key plus a certificate must also support TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA.

In addition to the cipher suites previously discussed, we may expect that further security suites may be adopted in future versions of CoAP, as this would enable a better adaptation of the various security modes to different applications and types of sensing platforms. CoAP also doesn't currently define or adopt any solution to address key management, other than the assumption that initial keys are available resulting from the DTLS authentication handshake.

VII. Open Research Issues

The protection of communications on the IoT using the previously analyzed technologies raises challenges and opportunities for further research work. In our following analysis we address existing proposals as well as opportunities in this very active area of research.

A. Security for PHY and MAC Layer Communications

Limitations of Security With IEEE 802.15.4: Despite the maturity of the IEEE 802.15.4 [6] standard, various limitations may be identified in respect to how it implements the security services supported by the MAC layer:

- As for the remaining communication protocols analyzed throughout this survey, the IEEE 802.15.4 does not specify any keying model. As discussed in the standard [6], this is mostly motivated by the fact that the most appropriate keying model is considered to be dependent on the threat model applicable to a particular application, and on the resources available on sensing devices to support key management operations.
- The management of IV values on IEEE 802.15.4 ACL entries may be problematic if the same key is used in two or more ACL entries. In this situation, it is possible that the sender will accidentally reuse the nonce value. This situation is potentially dangerous with stream ciphers encrypting in the CRT mode as AES/CCM, as it may enable an adversary to recover plaintexts from cipher texts. The reuse of nonce values is also possible due to the loss of ACL state after a power interruption, or when a node wakes up from a low-power mode.
- Tables storing ACL entries in IEEE 802.15.4 may not provide adequate support for all keying models, in particular group keying and network-shared keying. Group keying is in fact difficult to implement, since each ACL entry must be associated with a single destination address. Thus, the support of group keying requires various ACL entries using the same key, again promoting nonce reuse and the breaking of confidentiality, as previously discussed. On the other end, network shared keying is incompatible with replay protection. This mode may be supported only through the usage of the default ACL entry, and as such transmitter nodes would have to somehow coordinate their usage of replay counter space.
- As currently defined, IEEE 802.15.4 is unable to protect acknowledgment messages in respect to integrity or confidentiality. An adversary may therefore forge acknowledgments, for which it only needs to learn the sequence number of the packet to be confirmed that is sent in the clear, in order to perform DoS attacks.

The previously identified limitations in practice offer opportunities for improvements in future versions of the standard, and may also be circumvented by adopting security at other layers of the protocol stack illustrated in Fig. 1, as we proceed to discuss.

Research Challenges and Proposals for Security With IEEE 802.15.4: Key management mechanisms may be designed to support end-to-end security mechanisms at higher layers, thus circumventing the limitations of ACL management at the link-layer in respect to the support of group and network-shared keying. Key management approaches can also be designed to benefit from ACL storage space available in IEEE 802.15.4 sensing devices, even without supporting link-layer security. In the same context, AES/CCM available at the hardware in such platforms already provides the efficient cryptographic basis that security mechanisms at upper layers may benefit from. Standalone AES/CCM hardware encryption in fact provides an efficient cryptographic basis for research proposals addressing security at the network and higher layers.

Research opportunities also lie in the context of security in time-bounded link-layer communication environments employing IEEE 802.15.4e. As previously discussed, the applications are responsible for the definition of the communication schedules in such networks, and security mechanisms may be designed to benefit from the fact that the MAC layer operates using time-synchronized and channel-hopping communications. A possible approach is to design a communication schedule with slots reserved a priori for security, which can support normal security-management operations such as key management and the identification of misbehaving nodes for intrusion detection. New security solutions can also be proposed and discussed in the context of the recently formed IPv6 over the TSCH mode of IEEE 802.15.4e (6tisch) working group of the IETF.

B. Research Challenges and Proposals for Security at the Network-Layer

As previously analyzed, the current 6LoWPAN specification only discusses general security threats and requirements, despite RFC 4944 [9] clearly identifying the interest of adopting appropriate security mechanisms in the context of the 6LoWPAN adaptation layer. The research proposals discussed next offer solutions to the protection of IoT network-layer communications using 6LoWPAN.

Proposals for Confidentiality, Integrity, Authentication and Non-Repudiation: The Internet Protocol Security (IPSec) [51]–[53] architecture enables the authentication and encryption, at the network-layer, of the IP packets exchanged in the context of a given communication session, and provides support for Virtual Private Networks (VPN) in various usage modes. End-to-end network-layer security may also find useful usage scenarios in future IoT applications, in the context of which constrained sensing devices will be required to communicate with backend devices or with other Internet entities. Despite the advantages of end-to-end network-layer security, no specific security mechanisms have been adopted so far for the 6LoWPAN adaptation layer.

The challenges in the adoption of network-layer security approaches such as IPSec and IKE in 6LoWPAN environments are related to the resource constraints of typical wireless sensing platforms, and have been analyzed in previous research contributions [54], [55]. On the other end, the design of appropriate security mechanisms to work in tandem with the mechanisms at the 6LoWPAN adaptation layer would enable secure end-to-end communications at the network-layer and provide assurances in terms of confidentiality, integrity, authentication and non-repudiation.

A few research proposals currently exist with this purpose, focusing on the design of compressed security headers for the 6LoWPAN adaptation layer, with the same purpose as the existing Authentication Header (AH) and Encapsulating Security Payload (ESP) headers of the Internet Protocol Security (IPSec) [51]–[53]. This approach was initially proposed in [56], where the authors discuss that the employment of compressed security headers at the adaptation layer is a viable option, as long as carefully designed and sensing platforms are able to support efficient hardware security optimizations. The same authors later proposed and experimentally evaluated the usage of AH and ESP compressed security headers for 6LoWPAN in tunnel and transport modes [57], [58], considering predefined application security profiles and AES/CCM encryption at the hardware.

A more recent research work [59] also considers the design of compressed security headers for 6LoWPAN, in this case using shared-context LOWPAN_IPHC header compression. The experimental evaluation of this proposal and its comparison against IEEE 802.15.4 link-layer security is described in [60]. One advantage of this more recent proposal lies in the employment of the more recent IPHC compression scheme, as this provides support for global and multicast IPv6 addresses. Regarding the previous proposals, we must also consider that the support of 6LoWPAN network-layer security will also require appropriate support from external Internet entities, either by introducing support for compressed security headers and related security mechanisms in existing IPSec stacks, or in the other hand by designing mechanisms to support end-to-end network security with the help of a security gateway. Both aspects represent opportunities for research, for example in the design of mechanisms to support translation between IPSec and 6LoWPAN security, or of key management mechanisms mediated by the same gateway supporting such mapping operations.

Proposals for Security Against Packet Fragmentation Attacks: Regarding other security proposals for 6LoWPAN, authors in [61] discuss the consequences of packet fragmentation attacks against the 6LoWPAN fragmentation and reassembly mechanisms. As such mechanisms render buffering, forwarding and processing of fragmented packets challenging on resource-constrained devices, a malicious or misconfigured node sending forged, duplicate or overlapping fragments may threat the normal functioning or the availability of such devices. This is due to the lack of authentication at the 6LoWPAN adaptation layer, since recipients are unable to distinguish undesired fragments from legitimate ones when performing packet reassembly. The effects of fragmentation attacks include receiving buffer overflow and misusage of the available computational capability, among others. The paper proposes the addition of new fields to the 6LoWPAN fragmentation header to deal with such threats, namely of a timestamp providing protection against unidirectional fragment replays and of a nonce providing protection against bidirectional fragment replays.

Also in the context of fragmentation attacks, a more recent contribution [62] proposes the usage of mechanisms supporting per-fragment sender authentication and purging of messages from the receiver's buffer, for transmitter devices considered suspicious. The former employs hash chains enabling a legitimate sender to add an authentication token to each fragment during the 6LoWPAN fragmentation procedure, while in the later the receiver decides on which fragments to discard in case a buffer overload occurs, based on the observed sending behavior. This decision is based on per-packet scores, which capture the extent to which a packet is completed along with the continuity in the sending behavior. While this proposal does not require any modification to the 6LoWPAN packet formats, we may observe that the proposed security mechanisms would have to be adopted for the adaptation-layer.

Proposals for Key Management: An important security functionality discussed in the 6LoWPAN specification is key management, which may in reality be considered a cross-layer security aspect and interrelated with authentication, since keys must be negotiated and periodically refreshed in order to guarantee effective and long-term security, independently of the layer at which communications take place. While not proposing any specific key management solution, RFC 6568 [25] identifies the possibility of adopting simplified versions of current Internet key management solutions. For example, minimal IKEv2 [63] adapts Internet key management to constrained sensing environments, while maintaining compatibility with the existing Internet standard. Other approach consists in compressing of the IKE headers and payload information using 6LoWPAN IPHC compression, as proposed in [64]. New lightweight key management mechanisms appropriate to the IoT may also be designed. In [65] the authors discuss that public-key management approaches still require nodes more powerful than current reference sensing platforms, particularly if supporting services. The authors also discuss that mathematical-based key management solutions may also be adapted to support IoT applications [65].

C. Research Challenges and Proposals for Routing Security

The IETF RPL defines secure versions of routing control messages, together with a few basic security operations, but currently lacks mechanisms to support important operations. We proceed by discussing current research works focusing on security for RPL.

Limitations of RPL Security: We observe that, other than the secure versions of the routing control messages and the security modes previously discussed, no further security mechanisms are designed in the current version of the RPL Protocol standard [11]. The remaining documents produced in the IETF ROLL group discuss only general security requirements and goals, without defining particular security mechanisms. Considering that RPL already provides mechanisms to secure routing communications against external attacks, research efforts may be focused on the definition of threat models for RPL appropriate to particular application areas, and also on mechanisms to protect RPL communications and operations from internal attackers.

Identification of Threat Models: The current RPL specification [11] only addresses the handling of keys with applications employing device pre-configuration, discussing how such devices should be able to join a network using a preconfigured default shared group key or a key learned from a received DIS configuration message, while not defining how authentication and secure joining mechanisms may be designed to support other more dynamic or security-critical application contexts. Similarly to routing profiles defined for particular application areas, research and standardization may also target the definition of security policies stating how security must be applied to protect routing operations in a particular application context. Such policies may identify the requirements of applications in terms of confidentiality, integrity, authenticity and replay protection for control messages, among others.

A discussion on the open issues in respect to security in RPL is expressed in [66], which performs an analysis on the main threats against ROLL routing mechanisms, together with recommendations on how to address security. This document identifies such threats by employing the ISO 7498-2 security reference model [67], which includes Authentication, Access Control, Data Confidentiality, Data Integrity and Non-Repudiation, and to which Availability is added. This model enables the identification of the assets to protect, of its security needs, and of the points of access through which security may be compromised. The model enables the categorization and discussion of the threats and of the specific attacks regarding confidentiality, integrity and availability of routing message exchanges in the context of ROLL routing protocols. This document also proposes a security framework for ROLL routing protocols, which is built upon previous work on security for routing and adapting the assessments to the constraints of 6LoWPAN environments. In the context of this framework, security measures are identified that can be activated in the context of the RPL routing protocol, together with system security aspects that may impact routing but that also require considerations beyond the routing protocol, as well as potential approaches in addressing them. The assessments in this document may provide the basis of the security recommendations for incorporation into ROLL routing protocols as RPL. We also observe that the implications of the various security requirements, defined as appropriate for each application, to the routing protocol itself, is also a topic for future research and standardization work.

Proposals for Solutions Against Internal Attacks: Other important aspect of RPL security, as currently proposed, is that the services defined in the current specification [11] offer security against external attacks only. An internal attacker is in possession of a node and in consequence of the required security keys, and as such may selectively inject routing messages with malicious purposes. Authors in [68] discuss the issue of internal attacks on RPL, particularly on the rank concept as employed by the protocol. The rank serves the purposes of route optimization, loop prevention and management of routing control overhead. The paper discusses various possible attacks against the rank property, together with its impact on the performance of the network. Authors also discuss that this limitation in RPL is due to the fact that a child node receives parent information through control messages, but is unable to check the services provided by the parent, so it will follow a bad quality route if it has a malicious parent. While not proposing specific measures or mechanisms for this purpose, the paper discusses that mechanisms could be adopted in RPL to allow a node to monitor the behavior of its parents and defend against such threats.

Internal attacks against RPL are also discussed in [69], particularly that an internal attacker is able to compromise a node in order to impersonate a gateway (the DODAG root) or a node that is in the vicinity of the gateway. The authors propose a version number and rank authentication security scheme based on one-way hash chains, which binds version numbers with authentication data (MAC codes) and signatures. This scheme offers protection against internal attackers that are able to send DIO messages with higher version number values or that are able to publish a high rank value. The former attack enables an attacker to impersonate the DODAG root and initiate the reconstruction of the routing topology, while in the later a large part of the network may be forced to connect to the DODAG root via the attacker, thus providing the ability to eavesdrop and manipulate part of the network traffic. The security data enable intermediate nodes to validate DIO messages containing new version numbers and rank values. While an evaluation is performed against the impact of these mechanisms on computational time, the paper doesn't discuss its impact on aspects such as energy or memory of constrained sensing devices.

In another contribution focusing on internal attacks against RPL [70], the authors discuss the effects of sinkhole attacks on the network, particularly regarding its end-to-end data delivery performance in the presence of an attack. A sinkhole consists of a compromised node that purposely captures and drops messages. The authors propose the combination of a parent fail-over mechanism with a rank authentication scheme and, based on simulation results, argue that the combination of the two approaches produces good results, and also that by increasing the network density the penetration of sinkholes may be combated without needing to identify the sinkholes. The rank-verification technique is also based on one-way hash chains as in [69], while the parent fail-over scheme employs an end-to-end acknowledgment scheme controlled by the DODAG root node.

The previous research proposals represent approaches to address open security issues in RPL, particularly regarding the definition of a threat model applicable to RPL and mechanisms against internal attackers and threats. Such proposals may provide contributions to the adoption of other security mechanisms at the RPL standard itself in the future. As extensive research has been performed in the area of security for routing protocols for sensor networks and ad hoc networks in the past, approaches in such research proposals may also guide future approaches regarding RPL security, as long as appropriately designed to cope with the characteristics of 6LoWPAN devices and the internal operations of RPL. Finally, security mechanisms for the employment of asymmetric cryptography with RPL may also be proposed, given that the current specification of the protocol [11] does not define how node authentication and key retrieval are performed using public-keys or digital certificates.

D. Research Challenges and Proposals for Application-Layer Security

As previously discussed, DTLS is being considered to support security at the application-layer using CoAP. We may observe that DTLS presents some limitations motivating other approaches to security at the application-layer, as discussed next. In this context, work is also ongoing in the CoRE working group, in the context of which new approaches to security may be proposed and evaluated.

Limitations of CoAP Security: The impact of DTLS on current sensing platforms currently motivates research proposals on alternative approaches to protect IoT communications at the application layer using CoAP. One important aspect is that it is important to evaluate the impact of DTLS on sensing platforms with different characteristics because, if it is true that AES/CCM is efficiently available at the hardware in IEEE 802.15.4 sensing platforms, the DTLS handshake (for authentication and key agreement) can pose a significant impact on the resources of constrained devices, particularly considering the adoption of ECC public-key cryptography to support authentication and key agreement.

We verify that there is currently much interest in investigating optimizations for DTLS in IoT environments, and also on conducting interoperability testing of DTLS implementations using 6LoWPAN and CoAP [71], [72]. The DTLS In Constrained Environments (dice) working group of the IETF was also formed in 2013 to develop work in this context. Various features of the protocol have been identified as posing challenges to the adoption of DTLS in constrained sensing environments:

- The DTLS handshake [45] may be problematic to support, as large messages cause fragmentation at the 6LoWPAN adaptation layer and the cost of the computation of the *Finished* message at the end of the handshake is high [73], [74]. Fragmentation implies that retransmission and reordering of handshake messages at the DTLS communicating entities may result in added complexity and reliability.
- The support of ECC public-key cryptographic on 6LoWPAN environments requires further investigation, as the viability of ECC cryptography on constrained sensing platforms is not currently consensual.
- Devices in future IoT applications may require mechanisms supporting the online verification of the validity of X.509 certificates, particularly for the CoAP *Certificates* security mode. The design and adoption of mechanisms with this purpose requires further investigation.
- The employment of DTLS is not well suited to the usage of CoAP proxies in forward or reverse modes. Although end-to-end communications are at the hearth of IPv6, the exposure of constrained IoT devices to the Internet may call for security mechanisms based on the usage of security gateways, which may also support the roles of border routers for 6LoWPAN and CoAP communications.
- As discussed in [73], [74], other limitation is that DTLS is unable to support multicast communications, which will be required in many IoT environments. Secure CoAP multicast communications will also require appropriate group-keying mechanisms supporting the establishment of appropriate session keys among the various participating devices.

The previous issues motivate research proposals promoting the effectiveness of DTLS to protect CoAP communications, and also alternative approaches to security for IoT application-layer communications, as we analyze next.

Proposals for Key Management: As previously discussed, DTLS does not support group key management, and this poses a problem to the support of multicast communications using CoAP. Authors in [75] propose the adaptation of the DTLS record layer to enable multiple senders in a multicast group to securely send CoAP messages using a common group key, while providing confidentiality, integrity and replay protection to group messages. This proposal considers that the required group keying material is already available in the context of a given group security association, particularly the appropriate client and server read and write MAC keys, encryption keys and IV values.

Proposals for the Modification of DTLS: Other features of the protocol may be inappropriate to IoT applications and devices, and as such a suitable DTLS profile may be identified and adopted. In [76] the authors discuss various issues that may impede the usage of DTLS in constrained sensing devices, for example, the inadequateness of the timers for message retransmission as defined in the protocol, which may require large buffers on the receiver to hold data for retransmission purposes, and the size of the code required to support DTLS in constrained sensing platforms. The same document also discusses the usage of stateless compression of the DTLS headers with the goal of reducing the overhead of DTLS records and handshake messages. Authors in [77] follow this approach, and propose the compression of the DTLS headers using LOWPAN_IPHC 6LoWPAN header compression.

Other approach is to use CoAP to support costly DTLS handshake operations, as in [78]. In this proposal the authors define a RESTful DTLS handshake to deal with the problem of message fragmentation at the 6LoWPAN adaptation layer. The proposed mechanism enables the efficient transmission of DTLS handshake messages in the payload of CoAP messages using blockwise transfers when required for larger messages. In this proposal a DTLS session is modeled as a CoAP resource and a well-known URI path is used to identify a collection resource that models the set of active security sessions.

Proposals Offloading Costly DTLS Operations: Other proposals do exist based on the employment of gateways to support security-related mechanisms in the context of DTLS communications. As discussed in [73], [74], one issue to be addressed for CoAP security is the inexistence of mechanisms for mapping between TLS and DTLS. With this goal, authors in [79] propose a mechanism for mapping between TLS and DTLS at a security gateway, and the same gateway may also support mapping between CoAP and HTTP.

Another approach is to offload costly operations required by DTLS to more powerful devices, in particular using security gateways, as we analyze next. A few proposals consider this approach, focusing particularly on the delegation of operations performed in the context of the DTLS handshake. In [80] a mechanism is proposed also based on a proxy to support sleeping devices, using a mirroring mechanism to serve data on behalf of sleeping smart objects. In [81] the authors propose an end-to-end architecture supporting mutual authentication with DTLS, using specialized trusted-platform modules (TPM) supporting RSA cryptography on sensing devices, rather than ECC public-key cryptography as currently required for CoAP. This proposal is also described and more thoroughly evaluated in [82] using an experimental wireless sensor network. Authors in [83] also employ a security gateway, in this case to transparently intercept and mediate the DTLS handshake between the CoAP client and server, allowing the offloading of ECC public-key computations from constrained sensing devices to a security gateway without resource constrains. In this proposal the gateway, after the initial handshake, is in possession of the keying material it may use to decrypt communications between the two CoAP parties, thus supporting additional security mechanisms involving traffic analysis, for example intrusion detection and detection of attacks at the CoAP application-layer.

Proposals for the Support of Public-Keys and Digital Certificates: The impact of the processing of certificates using current sensing platforms is an aspect that also requires proper evaluation studies in a near future. Authors in [84] discuss possible design approaches to address the computational burden of supporting certificates in constrained sensing platforms, also by considering the usage of a security intermediary. The proposed approaches are certificate pre-validation and session resumption. Certificate pre-validation involves a security gateway supporting the validation of certificates in the context of the handshake, before forwarding the handshake messages to the destination sensing device. Session resumption allows communication peers to maintain minimal session state after session teardown, which they may use to later resume secure communications without the need of performing again the DTLS handshake. For very constrained sensing devices, this proposal addresses the full delegation of the DTLS handshake to a proxy using a mechanism based on TLS session resumption without server-side state.

Proposals for Object Security With CoAP: Recent research work is also considering the employment of alternative approaches to secure CoAP communications, in particular the employment of object security approaches rather than transport-layer security. This may be achieved by integrating security into to CoAP protocol itself using new security options. Authors in [85] propose the usage of new CoAP options to support security, in particular of three new options: one enabling the identification of how security is applied to a given CoAP message and of the entity responsible for the processing of security for the message, other enabling the transportation of data required to authenticate and authorize a CoAP client, and a third option enabling the transportation of security-related data required for the processing of cryptography for a CoAP message. This approach enables granular security on a per-message basis, and also supports the secure transversal of different domains and the usage of multiple authentication mechanisms.

Research Challenges in CoAP Security: Despite the previously analyzed research proposals, various issues remain to be addressed in the context of CoAP security. One important aspect to consider is the lack of appropriate key management mechanisms for the support of secure CoAP multicast communications. Group key management mechanisms may be designed either externally to CoAP, or on the other hand integrated with the DTLS handshake to support session key negotiation for a group of devices. Regarding the usage of DTLS header compression mechanisms [77], appropriate support will also be required from existing implementations, or on the other end mechanisms for mapping between DTLS and compressed DTLS may be designed. Such mechanisms may be supported by security gateways interconnecting low-energy sensing devices with the Internet, which may also support mapping between TLS and DTLS for end-to-end secure CoAP communications. Security gateways may also offer the possibility of supporting intrusion detection and attack tolerance mechanisms, and existing works on intrusion detection for sensor networks [86]–[88] may provide useful guidance in developing appropriate mechanisms for 6LoWPAN-based IoT communications.

Future research work may also target the support of public-keys and certificates in the context of CoAP security. Online validation of certificates may be achieved by investigating the applicability of existent Internet approaches such as the Online Certificate Status Protocol (OCSP) [89] or OCSP stapling through the TLS Certificate Status Request extension defined in RFC 6066 [90], considering that such mechanisms could be adapted or simplified to support constrained 6LoWPAN environments. OCSP stapling enables the presenter of a certificate to bear the resource cost involved in serving OCSP validation requests, instead of the issuing Certification Authority (CA).

TABLE II
SECURITY MECHANISMS AND PROPOSALS FOR IoT COMMUNICATION TECHNOLOGIES

Mechanisms and proposals	Operational layer	Security properties and functionalities supported	Context of application of security	Details
[57][58]	6LoWPAN adaptation	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (network layer) security	Stateless compression of AH and ESP security headers for 6LoWPAN; security in tunnel and transport modes; preprogrammed keys with varying sizes
[59][60]	6LoWPAN adaptation	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (network layer) security	6LoWPAN IPHC compression of AH and ESP security headers; preprogrammed 128-bit keys
[61]	6LoWPAN adaptation	Resistance against fragmentation attacks	Communications between 6LoWPAN devices using fragmentation	Addition of a timestamp plus a nonce to the 6LoWPAN fragmentation header to support security against unidirectional and bidirectional fragment replays
[62]	6LoWPAN adaptation	Resistance against fragmentation attacks	6LoWPAN communications between sensing devices or end-to-end communications with external devices	Usage of mechanisms to support per-fragment sender authentication using hash chains and purging of messages from suspicious senders based on the observed behavior
[76]	Transport-layer	Confidentiality, integrity and replay protection	Security for CoAP multicast communications	Adaptation of the DTLS record layer to enable multiple senders in a multicast group to securely send CoAP messages using a common group key
[77]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (transport layer) security	Compression of the DTLS headers in the context of 6LoWPAN using IPHC
[79]	Transport-layer	TLS and DTLS mapping for end-to-end secure communications	Transparent end-to-end (transport-layer) security	Mapping between TLS and DTLS using a gateway also providing HTTP to CoAP mapping
[80]	Transport-layer	Support of end-to-end transport-layer security for sleepy devices	Transparent end-to-end (transport-layer) security for inactive devices	Usage of a proxy to support secure end-to-end communications and data retrieval from devices that may be inactive
[81][82]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (transport layer) security	End-to-end DTLS using mutual authentication with hardware support provided by specialized trusted-platform modules (TPM) supporting RSA cryptography
[83]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (transport layer) security	Transparent interception and mediation of the DTLS handshake, enabling the offloading of ECC public key computations to the gateway
[84]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation	End-to-end (transport layer) security with certificates and sessions managed at the gateway	Usage of the certificate pre-validation and session resumption to offload public key authentications to the gateway
[11]	Routing layer	Confidentiality, integrity, authentication, non-repudiation	Protection of RPL routing control messages	Definition of secure versions of the RPL routing control messages, together with two security modes to protect routing updates
[66]	Routing layer	Security framework for ROLL routing protocols	Identification of security measures appropriate to the RPL routing protocol	Identification of security measures that can be activated in the context of RPL and of the system aspects that may impact on routing, as well as potential approaches in addressing them
[69]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Usage of a version number and rank authentication security scheme based on one-way hash chains providing security against internal attackers
[70]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Usage of a security mechanism combining parent fail-over with a rank authentication scheme to combat sinkhole attacks
[12]	Application layer	Confidentiality, integrity, authentication, replay protection	Protection of CoAP application-layer messages using DTLS at the transport-layer	Definition of bindings to DTLS to protect CoAP messages, together with three security modes with different approaches to cryptographic key management
[78]	Application layer	Support of DTLS handshake using CoAP communications	Support authentication and initial key agreement with sensing devices employing DTLS	DTLS handshake messages are transported in the payload of CoAP application-layer messages using CoAP blockwise transfers to reduce 6LoWPAN fragmentation
[85]	Application layer	Confidentiality, integrity, authentication, non-repudiation	Transparent and granular end-to-end (application layer) security	CoAP security options allow for granular security, authentication of clients and secure transversal of multiple security domains

Other important issue to consider is the computational impact of ECC cryptography on existing sensing devices. In this context, optimizations may be designed at the hardware of sensing platforms to support ECC computations, similarly to the support of AES/CCM in IEEE 802.15.4 platforms.

VIII. CONCLUSION

A glimpse of the IoT may be already visible in current deployments where networks of sensing devices are being interconnected with the Internet, and IP-based standard technologies will be fundamental in providing a common and well-accepted ground for the development and deployment of new IoT applications. Considering that security may be an enabling factor of many of such applications, mechanisms to secure communications using communication technologies for the IoT will be fundamental. With such aspects in mind, in the survey we perform an exhaustive analysis on the security protocols and mechanisms available to protect communications on the IoT. We also address existing research proposals and challenges providing opportunities for future research work in the area.

In Table II we summarize the main characteristics of the mechanisms and proposals analyzed throughout the survey, together with its operational layer and the security properties and functionalities supported. In conclusion, we believe this survey may provide an important contribution to the research community, by documenting the current status of this important and very dynamic area of research, helping readers interested in developing new solutions to address security in the context of communication protocols for the IoT.

REFERENCES

[1] M. Palattella *et al.*, "Standardized protocol stack for the Internet of (Important) things," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1389–1406, 2013.

[2] G. Gan, L. Zeyong, and J. Jun, "Internet of things security analysis," in *Proc. IEEE Conf. iTAP*, 2011, pp. 1–4.

[3] C. Medaglia and A. Serbanati, "An overview of privacy and security issues in the Internet of things," in *The Internet of Things*. New York, NY, USA: Springer-Verlag, 2010, pp. 389–395.

[4] R. Weber, "Internet of things-new security and privacy challenges," *Comput. Law Security Rev.*, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[5] C. Xiangqian, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 52–73, 2009.

[6] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4-2011 (Revision of IEEE Std. 802.15.4-2006), (2011) 1-314, 2011.

[7] *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. 802.15.4e-2012 (Amendment to IEEE Std. 802.15.4-2011), (2011) 1-225, 2012.

[8] N. Kushalnagar, G. Montenegro, and C. Schumacher, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, Goals, RFC 4919, 2007.

[9] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, Transmission of IPv6 Packets Over IEEE 802.15.4 Networks, RFC 4944, 2007.

[10] J. Hui and P. Thubert, Compression Format for IPv6 Datagrams Over IEEE 802.15.4-Based Networks, RFC 6282, 2011.

[11] P. Thubert *et al.*, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, 2012.

[12] C. Bormann, A. Castellani, and Z. Shelby, "CoAP: An application protocol for billions of tiny Internet nodes," *IEEE Internet Comput.*, vol. 1, no. 2, pp. 62–67, Mar./Apr. 2012.

[13] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4a-2007 (Amendment to IEEE Std. 802.15.4-2006), 2007, pp. 1, 203.

[14] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 2: Alternative Physical Layer Extension to Support One or More of the Chinese 314–316 MHz, 430–434 MHz, 779–787 MHz Bands*, IEEE Std. 802.15.4c-2009 (Amendment to IEEE Std. 802.15.4-2006), Apr. 17, 2009, pp. c1, 21.

[15] *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 3: Alternative Physical Layer Extension to support the Japanese 950 MHz Bands*, IEEE Std. 802.15.4d-2009 (Amendment to IEEE Std. 802.15.4-2006), 2009, pp. 1–27.

[16] ZigBee Alliance, pp. 344–346 ZigBee specification, 2006, pp. 344–346.

[17] ZigBee Alliance, ZigBee PRO Specification, 2007.

[18] The International Society of Automation, Wireless Systems for Industrial Automation: Process Control and Related Applications ISA 100.11a, 2009.

[19] A. Kim *et al.*, "When HART goes wireless: Understanding and implementing the WirelessHART standard," in *Proc. IEEE Int. Conf. ETFA*, 2008, pp. 899–907.

[20] The IEEE Standard Association, Guidelines for 64-bit Global Identifier (EUI-64), (accessed Nov. 2014), 2013. [Online]. Available: http://standards.ieee.org/db/oui/tutorials/EUI64.html

[21] K. Pister and L. Doherty, "TSMP: Time synchronized mesh protocol," in *Proc. IASTED Distrib. Sensor Netw.*, 2008, pp. 391–398.

[22] Texas Instruments, Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee Ready RF Transceiver, (accessed Nov. 2014). [Online]. Available: http://www.ti.com/product/cc2420

[23] MEMSIC, TelosB Mote Platform, (accessed Nov. 2014). [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf

[24] F. Miller, A. Vandome, and J. McBrewster, Advanced Encryption Standard, 2009.

[25] E. Kim, D. Kaspar, and J. Vasseur, Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6568, 2012.

[26] E. Kim, D. Kaspar, C. Gomez, and C. Bormann, Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing, RFC 6606, 2012.

[27] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, 2012.

[28] J. Hui and D. Culler, "Extending IP to low-power, wireless personal area networks," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 37–45, Jul./Aug. 2008.

[29] G. Carles, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012.

[30] I. Ishaq, D. Carels, G. Teklemariam *et al.*, "IETF standardization in the field of the Internet of things (IoT): A survey," *J. Sensor Actuator Netw.*, vol. 2, no. 2, pp. 235–287, Apr. 2013.

[31] International Telecommunication Union (ITU), G.9959: Short Range Narrow-Band Digital Radiocommunication Transceivers—PHY and MAC Layer Specifications, (accessed Nov. 2014). [Online]. Available: http://www.itu.int/rec/T-REC-G.9959-201202-I/en

[32] Y. Hong, Y. Choi, J. Youn, D.-K. Kim, and J.-H. Choi, Transmission of IPv6 Packets Over Near Field Communication, draft-hong-6lo-ipv6-over-nfc-02, 2014.

[33] D. Trček, "Lightweight protocols and privacy for all-in-silicon objects," *Ad Hoc Netw.*, vol. 11, no. 5, pp. 1619–1628, Jul. 2013.

[34] T. Narten, E. Nordmark, and W. Simpson, Neighbor Discovery for IP version 6 (IPv6), RFC 4861, 2007.

[35] J. Arkko, J. Kempf, B. Zill, J. Arkko, and P. Nikander, Secure Neighbor Discovery (SEND), RFC 3971, 2005.

[36] T. Aura, Cryptographically Generated Addresses, RFC 3972, 2005.

[37] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548, 2009.

[38] K. Pister, P. Thubert, S. Dwars, and T. Phinney, Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673, 2009.

[39] A. Brandt, J. Buron, and G. Porcu, Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826, 2010.

[40] J. Martocci, P. De Mil, N. Riou, and W. Vermeylen, Building Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5867, 2010.

[41] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, Routing Metrics Used for Path Calculation in Low Power and Lossy Networks, RFC 6551, 2012.

[42] A. Conta, S. Deering, and M. Gupta, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC 4443, 2006.

[43] J. Postel, Transmission Control Protocol, RFC 793, 1981.

[44] T. Zheng, A. Ayadi, and X. Jiang, "TCP over 6LoWPAN for industrial applications: An experimental study," in *Proc. IEEE 4th IFIP Int. Conf. NTMS*, 2011, pp. 1–4.

[45] E. Rescorla and N. Modadugu, DTLS: Datagram Transport Layer Security, RFC 4347, 2006.

[46] T. Dierks and E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.1, RFC4346, 2006.

[47] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, and T. Kivinen, Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), RFC 7250, 2014.

[48] SECG-Elliptic Curve Cryptography-SEC 1, (accessed Nov. 2014). [Online]. Available: http://www.secg.org

[49] D. McGrew and D. Bailey, AES-CCM Cipher Suites for Transport Layer Security (TLS), RFC 6655, 2012.

[50] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Moeller, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS),RFC 4492, 2006.

[51] S. Kent and K. Seo, Security Architecture for the Internet Protocol, RFC 4301, 2005.

[52] S. Kent and R. Atkinson, IP Authentication Header, RFC 2402, 1998.

[53] S. Kent and R. Atkinson, Encapsulating Security Protocol, RFC 2406, 1998.

[54] R. Riaz, K. Kim, and H. Ahmed, "Security analysis survey and framework design for ip connected lowpans," in *Proc. ISADS*, 2009, pp. 1–6.

[55] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43. Hoboken, NJ, USA: Wiley, 2011.

[56] J. Granjal, J. Silva, E. Monteiro, J. Sa Silva, and F. Boavida, "Why is IPSec a viable option for wireless sensor networks," in *Proc. 5th IEEE Int. Conf. MASS*, 2008, pp. 802–807.

[57] J. Granjal, E. Monteiro, and J. Silva, "Enabling network-layer security on IPv6 wireless sensor networks," in *Proc. GLOBECOM*, 2010, pp. 6–10.

[58] J. Granjal, E. Monteiro, and J. Silva, "Network-layer security for the Internet of Things using TinyOS and BLIP," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 1938–1963, Oct. 2012.

[59] S. Raza, S. Duquennoy, and T. Voigt, "Securing communication in 6LoWPAN with compressed IPsec," in *Proc. Int. Conf. DCOSS Workshops*, 2011, pp. 1–8.

[60] S. Raza, S. Duquennoy, J. Hoglund, U. Roedig, and T. Voigt, "Secure communication for the Internet of Things—A comparison of link-layer security and IPsec for 6LoWPAN," *Security Commun. Netw.*, vol. 7, no. 12, pp. 2654–2668, Dec. 2014.

[61] H. Kim, "Protection against packet fragmentation attacks at 6lowpan adaptation layer," in *Proc. ICHIT*, 2008, pp. 796–801.

[62] R. Hummen *et al.*, "6LoWPAN fragmentation attacks and mitigation mechanisms," in *Proc. 6th ACM Conf. WiSec*, 2013, pp. 55–66.

[63] H. René, H. Wirtz, J. H. Ziegeldorf, J. Hiller, and W. Wehrle, "Tailoring end-to-end IP security protocols to the Internet of things," in *Proc. 21st IEEE ICNP*, 2013, pp. 1–10.

[64] R. Shahid, T. Voigt, and V. Jutvik, "Lightweight IKEv2: A key management solution for both the compressed IPsec and the IEEE 802.15. 4 security," in *Proc. IETF Workshop Smart Object Security*, 2012. **AQ1**

[65] R. Rodrigo, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of things," *Comput. Elect. Eng.*, vol. 37, no. 2, pp. 147–159, Mar. 2011.

[66] T. Tsao *et al.*, A Security Threat Analysis for Routing over Low Power and Lossy Networks, draft-ietf-roll-security-threats-11, 2014 (active, work in progress).

[67] *Information Processing Systems—Open Systems Interconnection Reference Model—Security Architecture*, ISO Standard 7498-2, 1988.

[68] A. Le *et al.*, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3685–3692, Oct. 2013.

[69] A. Dvir, T. Holczer, and L. Buttyan, "VeRA—Version number and rank authentication in RPL," in *Proc. IEEE 8th Int. Conf. MASS*, 2011, pp. 709–714.

[70] K. Weekly and K. Pister, "Evaluating sinkhole defense techniques in RPL networks," in *Proc. 20th IEEE ICNP*, 2012, pp. 1–6.

[71] IoT CoAP Plugtests, 28–30 November 2012, (accessed Nov. 2014). [Online]. Available: http://www.etsi.org/plugtests/coap2/Home.htm

[72] 6LoWPAN Plugtests, 27–28 July 2013, (accessed Nov. 2014). [Online]. Available: http://www.etsi.org/news-events/events/663-2013-6lowpan-plugtests

[73] O. Garcia-Morchon, S. Kumar, R. Hummen, and M. Brachmann, Security Considerations in the IP-Based Internet of Things, draft-garcia-core-security-06, 2013.

[74] M. Brachmann, O. G. Morchon, S. Keoh, and S. Kumar, "Security considerations around end-to-end security in the IP-based Internet of things," in *Proc. Workshop Smart Object Security Conjunction IETF83*, 2012, pp. 1–3.

[75] S. Keoh, S. Kumar, O. Garcia-Morchon, and E. Dijk, DTLS-Based Multicast Security for Low-Power and Lossy Networks (LLNs), draft-keoh-dice-multicast-security-08, (active, work in progress) 2014.

[76] K. Hartke, Practical Issues With Datagram Transport Layer Security in Constrained Environments, draft-hartke-dice-practical-issues-01, 2014.

[77] R. Shahid, T. Daniele, and T. Voigt, "6LoWPAN compressed DTLS for COAP," in *Proc. 8th IEEE Int. Conf. DCOSS*, 2012, pp. 287–289.

[78] S. Keoh, S. Kumar, and Z. Shelby, Profiling of DTLS for CoAP-Based IoT Applications, draft-keoh-dice-dtls-profile-iot-00, 2013.

[79] M. Brachmann, S. Keoh, O. G. Morchon, and S. S. Kumar, "End-to-end transport security in the IP-based Internet of things," in *Proc. 21st Int. Conf. Comput. Commun. Netw.*, 2012, pp. 1–5.

[80] M. Sethi, A. Jari, and K. Ari, "End-to-end security for sleepy smart object networks," in *Proc. 37th IEEE Local Comput. Netw. Workshops*, 2012, pp. 964–962.

[81] T. Kothmayr, C. Schmitt, W. Hu, M. Brunig, and G. Carle, "A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication," in *Proc. 37th IEEE Conf. LCN Workshops*, 2012, pp. 956–963.

[82] T. Kothmayr, C. Schmitt, W. Hu, M. Brunig, and G. Carle, "DTLS based security and two-way authentication for the Internet of things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2710–2723, Nov. 2013.

[83] J. Granjal, E. Monteiro, and J. Sá Silva, "End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication," in *Proc. IFIP Netw.*, 2013, pp. 1–9.

[84] R. Hummen, J. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards viable certificate-based authentication for the Internet of things," in *Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Security Privacy*, 2013, pp. 37–42.

[85] J. Granjal, E. Monteiro, and J. Sá Silva, "Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications," in *Wired/Wireless Internet Communication*. Berlin, Germany: Springer-Verlag, 2013, pp. 140–153.

[86] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 2014.

[87] M. Young and E. Boutaba, "Overcoming adversaries in sensor networks: A survey of theoretical models and algorithmic approaches for tolerating malicious interference," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 617–641, 2011.

[88] A. Abduvaliyev, A. Pathan, Z. Jianying, R. Roman, and W. C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1223–1237, 2013.

[89] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, X. 509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP, RFC 2560, 1999.

[90] D. Eastlake, Transport Layer Security (TLS) Extensions: Extension Definitions, RFC 6066, 2011.

Authors' photographs and biographies not available at the time of publication. **AQ2**

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = Please provide page range in Ref. [64].
AQ2 = Please provide photographs and biographies of all authors.

END OF ALL QUERIES