Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Survey Paper

Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey

Jorge Granjal [*], Edmundo Monteiro, Jorge Sá Silva

Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal

ARTICLE INFO

Article history:
Received 14 September 2013
Received in revised form 16 May 2014
Accepted 1 August 2014
Available online xxxx

Keywords:
Internet-integrated WSN
6LoWPAN
RPL
DTLS
CoAP
Internet of Things

ABSTRACT

The integration of low-power wireless sensing and actuating devices with the Internet will provide an important contribution to the formation of a global communications architecture encompassing Wireless Sensor Networks (WSN), and to enable applications using such devices designed to bring unprecedented convenience and economical benefits to our life. Such applications also take place in the context of our current vision on an Internet of Things (IoT), which promises to encompass heterogeneous devices and communication technologies, including WSN. Due to the characteristics of the devices in WSN and to the requirements of applications, low-power wireless communications are employed and the functionalities supported must be carefully balanced against the limited resources at the disposal of applications. Low-power communication technologies are also currently being designed with the purpose of supporting the integration of WSN with the Internet and, as in isolated WSN environments, security will be a fundamental enabling factor of future applications using Internet-integrated WSN. Although various surveys currently exist addressing security mechanisms for WSN environments, our goal is to analyze how security may be addressed as an enabling factor of the integration of low-power WSN with the Internet, in the context of its contribution to the IoT. We analyze the current research and industry proposals supporting this integration, together with the security solutions and mechanisms designed in its context. Our discussion is supported by an analysis on the attack and threat model against Internet-integrated WSN, and on the security requirements to consider in this context. We believe that a survey with such goals may provide an important contribution to readers interested in embracing this important area of research and ours is, as far as our knowledge goes, the first article with such goals.

1. Introduction

The Internet of Things (IoT) is a widely used expression, currently referring to a vision of a future Internet where ubiquitous sensing applications in diverse areas will provide benefits to our daily lives. Security will be a fundamental requirement of most of such applications, and appropriate mechanisms will be required to cope with the users' expectations and requirements in terms of security. Various communication patterns will be supported by a future IoT communications infrastructure, among which Human-to-Machine (H2M) and Machine-to-Machine (M2M) communications, and this infrastructure is expected to encompass diverse communication technologies, as Near Field Communications (NFC), Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSN), among others. The integration of WSN with the Internet may play an important role in the evolution of the architecture of the

[*] Corresponding author. Tel.: +351 239790035.
E-mail addresses: jgranjal@dei.uc.pt (J. Granjal), edmundo@dei.uc.pt (E. Monteiro), sasilva@dei.uc.pt (J.S. Silva).

http://dx.doi.org/10.1016/j.adhoc.2014.08.001
1570-8705/© 2014 Elsevier B.V. All rights reserved.

Internet, since WSN deployments may be used to support the sensorial capabilities required by future applications. Such aspects also motivate our analysis throughout the article on how security may be addressed in the context of the integration of WSN with the Internet.

In an architectural context, technologies supporting IoT communications may be currently classified in three main categories, as considered later in this article: backbone, backhaul and capillary communication technologies, to which low-power WSN belong. Communication and security technologies adopted for Internet-integrated WSN are expected to share many characteristics with proposals for classic WSN environments, in which sensing devices are employed to enable applications that are proprietary and designed with very particular goals in mind. Numerous challenging aspects characterize the design of communication and security technologies for WSN environments, and such challenges will also be present from the minute we start integrating such networks with the Internet, particularly if this integration involves the exposure of constrained sensing devices and of low-power wireless communications to external or Internet-originated threats and attacks. The characteristics and constraints of WSN environments and devices typically determine the employment of technologies designed to support wireless communications with appropriate reliability and a moderate impact on the energy of sensing devices. As a consequence, such communications run at low speeds in order to reduce collision and retransmission probabilities. Such aspects will also pose difficulties to the design and adoption of appropriate security measures against security threats in Internet-integrated WSN. Despite such challenges, security will be one fundamental enabling factor of this integration, and as such is of paramount importance.

In this article we perform a detailed survey on the available mechanisms offering security in the context of Internet-integrated low-power WSN. Such mechanisms may be related with particular communication technologies developed to enable such environments, or on the other end be designed to protect them from security threats and attacks. With this goal in mind, we analyze the various approaches currently proposed to achieve this integration, which result both from research and industry efforts, and also the open issues and research challenges in this area. In particular, we consider the integration via cloud-based platforms, front-end proxies and specialized architecture frameworks. Our discussion on such approaches also lays the ground for our analysis on the integration of WSN with the Internet communications architecture via standard communication protocols currently being developed for low-power WSN environments, which we analyze in detail in respect to the technologies and recent research proposals at the various layers of the protocol stack. We must also note that the goal of this article is distinct from the numerous existing surveys on security for WSN environments [1,2], given that such works focus on proposals for WSN applications using sensing devices in isolated deployments, where such devices are unable to communicate with other entities or devices on different WSN or on the Internet. Our discussion also differs from existing surveys focusing on security on the IoT in a high-level perspective [131,133], or on the other end on its legal aspects [132]. It is also important to observe that, as mechanisms enabling the integration of WSN with the Internet result from both research and standardization efforts, our analysis along the survey reflects this duality.

The discussion on this article proceeds as follows. In Section 2 we discuss the importance of low-power WSN communications and of its integration with the Internet, while in Section 3 we discuss the attack and threat model applicable to this integration, its main security requirements and the current integration approaches. In Section 4 we perform an exhaustive analysis on the existing mechanisms to support communications and security in the various integration approaches, together with the open issues and research challenges in this area. Finally, in Section 5 we conclude the survey.

2. IoT and M2M technologies

Various communication technologies are already available or currently being designed that may be part of a future communications architecture supporting various types of devices. In this context, WSN devices serve the important purpose of providing applications with the required sensing and actuating capabilities using low-power devices and wireless communications. The technologies currently identifiable in this context are identified in Fig. 1 and, as illustrated, we consider them to be divided in three main categories: backbone, backhaul and capillary communications technologies. In this figure we also illustrate the possible interactions between technologies at different categories.

Interactions between different technologies may be in practice supported by devices implementing mechanisms for translation between different communication technologies, or on the other end supporting different communication technologies simultaneously. In the former situation we may encounter specialized devices or gateways interconnecting different communication domains and that may support different integration strategies, while on the later devices may be employed that support two or more wireless communication technology simultaneously, such as recent Wi-Fi/ZigBee single chip platforms [3]. As illustrated in Fig. 1, we also consider that low-power WSN communications based on IEEE 802.15.4 [16] and 6LoWPAN [17–19] support capillary communications for applications requiring sensing and actuating capabilities using low-power WSN devices.

2.1. Backbone communication technologies

As in the current Internet communications architecture, backbone communications can be supported by both wired and wireless communication technologies. Wired communication technologies may include IEEE 802.3 [4] Ethernet-based communications, as well as Synchronous Optical Networking (SONET) and Synchronous Digital Hierarchy (SDH) [5] fiber optic-based communications. A particularly important role in this context will be played by broadband wireless communication technologies, given the increasing

Please cite this article in press as: J. Granjal et al., Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.08.001

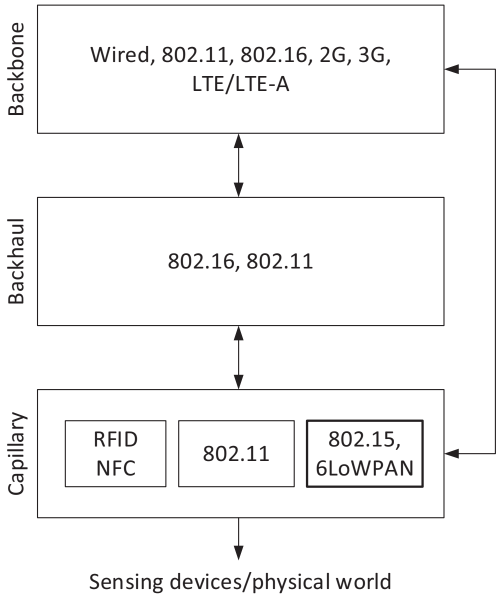
Backbone
Wired, 802.11, 802.16, 2G, 3G, LTE/LTE-A
Backhaul
802.16, 802.11
Capillary
RFID
NFC
802.11
802.15, 6LoWPAN
Sensing devices/physical world

Fig. 1. IoT communication technologies.

adoption of mobile devices requiring pervasive broadband Internet communications. Technologies in this case include second-generation GSM [6] from ETSI [7], third generation UMTS [8] from 3GPP [9] and fourth generation LTE and LTE-A [10] also from 3GPP. We may also consider the employment of IEEE 802.11 [11] and IEEE 802.16 [12] technologies to support wireless backbone communications. IEEE 802.11 [11] provides communications focused mostly on wireless local area network (WLAN) applications, but may also support applications designed for larger geographical areas, and the same reasoning may apply also to IEEE 802.16 [12] WiMax (Worldwide Interoperability for Microwave Access), which are focused on wireless metropolitan area network (WMAN) applications. Despite recognizing the subjectivity of the classification illustrated in Fig. 1, we note that its main purpose is to enable the contextualization of the various communication technologies that are our main target, and in particular of WSN in the context of capillary communication technologies. Low-power WSN may thus support capillary communications over the Internet, if integrated with the Internet communications infrastructure. We may also note that, considering the current status and the dynamism surrounding research on IoT technologies, it is fair to consider that no definitive or more informed classification can be delineated at this time.

2.2. Backhaul communication technologies

Backhaul communication technologies fulfill the important role of supporting communications between the capillary and backbone communication domains, and also of bridging between different capillary communication technologies. As in backbone communications, the technologies employed may depend on the application at hand, and on factors such as the geographical coverage of applications, its communication requirements and the types of devices employed. In this category, we also encounter WLAN technologies supported by IEEE 802.11 [11] and WMAN technologies supported by 802.16 [12] WiMax. Backhaul communication technologies may thus support geographically distributed IoT sensing applications employing distributed capillary domains and devices, as well as communications between capillary domains (as WSN) and the global Internet communications infrastructure.

2.3. Capillary communication technologies

The communication technologies employed at the capillary hierarchical level will be particularly important in the IoT, given their support of the final step in the communications path toward the sensing and actuating devices interfacing with the physical world. As previously discussed, our main focus on this article is on low-power WSN, which may support capillary communications with sensing and actuating devices if integrated with the Internet. We also observe the importance of other communication technologies in this context, as Fig. 1 illustrates, namely of RFID now becoming widely used for authentication and goods tracking, and of NFC which is increasingly being adopted to support applications such as contactless payments and ticketing. WSN may in general enable the deployment of data-collection systems using constrained low-power autonomous wireless sensing devices, and the integration of such systems with the Internet communications infrastructure promises to dramatically improve its usefulness and pervasiveness.

From a security perspective, the employment of wireless communications and of constrained sensing devices will pose difficulties to the integration of WSN with the Internet. Most of the existing communications and security proposals for WSN [1,2] are designed for very specific applications, and not to support Internet communications in the context of this integration. In fact, most research efforts on WSN in recent years were motivated by the fact that embedded devices have limited processing and communication capabilities, and in the light of such restrictions the integration of such network with the Internet was deemed unfeasible. As we observe throughout this article, this perception is currently changing and solutions are being developed to support the integration of WSN with the Internet.

As Fig. 1 illustrates, other than RFID and NFC, we may identify two main classes of capillary IoT communication technologies. IEEE 802.11-based WLAN may support communications with devices that are less resource constrained, for example mobile phones, embedded devices with continuous power sources, or devices supporting 802.11 side-by-side with low-power wireless communications [3]. 802.11 is also currently being optimized to support low-power wireless communications, in particular in 802.11ah [13] to support sub 1-GHz communications for sensor network and smart metering applications using 802.11. Thus, it is possible that future versions of the 802.11 standard may also support low-power WSN applications. Finally, low-power communications technologies are designed for communication environments currently identified as low-power wireless personal area networks (LoWPAN), to which WSN apply. The most representative family of communication technologies in this class is IEEE 802.15 [14]. In this family, of particular interest to Internet-integrated WSN is IEEE 802.15.6 [15] and IEEE 802.15.4 [16]. IEEE 802.15.6 is designed to support wireless body area networks (WBAN) applications, while IEEE 802.15.4 supports low-power and short-range wireless communications. As we observe later in the article, IEEE 802.15.4 physical (PHY) and medium access control (MAC) communications currently provide the ground for the design of Internet communications protocols for WSN.

3. Security in the integration of low-power WSN with the Internet

The integration of WSN applications and low-power sensing devices with the Internet may be accomplished with various approaches and strategies, as we analyze throughout the article. Other than indirect interconnection strategies, technologies are currently being designed with the purpose of enabling standard Internet communications on LoWPAN environments, and such technologies offer thus the possibility of enabling direct end-to-end Internet communications between low-power wireless sensing devices and external or Internet hosts. In particular, 6LoWPAN [17–19], RPL [20] and CoAP [21] are being designed over IEEE 802.15.4 with this purpose, as we analyze in detail later in the article. Irrespective of the integration approach, it is important to identify the applicable attack and threat model, as we discuss next, and also to identify the security requirements derived from such threats. We also discuss the applicability of the existing security technologies to protect Internet-integrated WSN environments, and analyze what are the current industry and research approaches to achieve this integration.

3.1. Attack and threat model

In addition to the threats that are inherent to the characteristics and constraints of low-power WSN environments, its integration with the Internet may also motivate attacks that are possible due to exposure of WSN devices and applications to global Internet communications, with the degree of exposure depending on the integration strategy and on the security mechanisms employed to protect WSN environments from internal or external threats, as we discuss later in the article. As in WSN deployments without Internet integration, we must consider the threats and attacks motivated by the wireless nature of WSN communications, the resource constraints of WSN devices and its (possible) physical exposure. The integration of WSN with the Internet raises the bar for security, as such environments must be protected from external threats and attacks, and end-to-end communications between wireless sensing devices and external entities will also require appropriate security mechanisms. The following discussion applies to Internet-integrated WSN environments, irrespective of the integration strategy considered.

As in security for isolated WSN environments, we may classify attackers against the normal functioning of Internet-integrated WSN as being either *internal* or *external*. An internal attacker is one that is able to compromise a node and to participate in a communications session as a fully legitimate entity, even if security/cryptographic mechanisms are in place, since it may have access to the secret keying material required to do so. As in this context WSN may communicate with the Internet, an internal attacker may also be an external or Internet device that has been compromised. On the other end, external attacks usually consist in the listening on the wireless channel in order to obtain knowledge about the functioning of the network. Therefore, an external attacker is usually not in the possession of secret keying material and is in principle easier to defend against, when compared with an internal attacker.

When considering how attacks may be performed and perceived by legitimate communicating entities, we may classify attacks as either *passive* or *active*. In this context, the type of attack is related to how the actions of the attacker influence or are noticeable to the other devices operating on the network. A passive attack is one in which the attacker does not interact with any of the other devices on the network, attempting to break the system solely based upon observed communications, while an active attacker may attempt to compromise the security of the network in any way, without concerns about its actions being noticed.

Considering the previous classification, we may identify various attacks against Internet-integrated WSN, either active or passive, and which may also target WSN environments not integrated with the Internet, as such having been the focus of recent research work [1,2]. An *eavesdropping* attack is a passive and external attack, which consists in listening for and possibly recording network traffic in order to derive any kind of useful information. A *spoofing* attack is an active attack that may be perpetrated by an internal or external attacker, in which an attacker masquerades as another one in order to gain an illegitimate advantage. The *physical compromise* of a sensing device constitutes a type of *denial of service* (DoS) attack, which in general consists of an attacker performing malicious actions to prevent legitimate users from using a service or functionality of the network. DoS attacks are particularly pernicious to low-energy WSN communication environments, and may target the consumption of the device's limited resources or the prevention of proper communications. Distributed Denial of Service (DDoS) attacks are also possible, which consist in the simultaneous action of many nodes flooding a target with requests. Other examples of DoS attacks are the jamming attack against communications at the physical layer, which enables an attacker to disrupt wireless communications by overwhelming the radio carrier with bogus data, and attacks against the MAC (Media Access Control), which consists in an attacker purposely creating collisions by sending its own packet when a legitimate user's packet is being transmitted. We have also to consider attacks at higher layers of the stack. For example, attacks against the routing protocols have been developed against all major Internet routing

Please cite this article in press as: J. Granjal et al., Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.08.001

protocols, and will also be of concern in future Internet-integrated WSN. WSN devices are often characterized by the presence of a routing decision at each node and are especially vulnerable to advanced attacks against routing protocols as RPL [20]. Examples of such attacks are the Sybil attack (a node illegitimately taking multiple identities), the wormhole attack (using an offline channel to tunnel routing packets between two distant points of the network) and black hole attack (sensing network traffic to a device that discards all packets), among others.

The classic approach against external attacks consists in the usage of security protocols employing cryptographic mechanisms. Protocols with this purpose are designed to guarantee fundamental security properties for the packet exchanged, in particular confidentiality, integrity, authentication and non-repudiation. In the same context, effective security through encryption requires appropriate authentication and key management mechanisms in place, since encryption algorithms are only effective as long as the keys employed are refreshed periodically. As we discuss later, cryptographic approaches are also pervasive in proposed research solutions targeting the protection of communications in the context of Internet-integrated WSN, as are related key management solutions. On the other end, protection against internal attackers usually requires other mechanisms, and in this case prevention is usually the key for success. Security procedures such as security perimeter enforcement via access control mechanisms or intrusion prevention and detection systems may be employed to defend against such attacks from their very beginning. Due to the physical exposure of sensing devices in many WSN deployments, mechanisms against the tampering of devices may also be employed.

When compared against WSN environments isolated from external communications, we may consider that the integration of a WSN with the Internet contributes to amplify the previously discussed security threats and attacks. Depending on the integration strategy followed, appropriate security mechanisms may be designed to protect communications in which sensing devices participate, which may be end-to-end with devices on other WSN deployments, or with external or Internet hosts if standard Internet communication technologies developed for WSN are in place. Mechanisms may also be designed to cope with cross-layer security aspects and to operate in a cross-layer fashion. Finally, such mechanisms may also be either distributed or based on specialized devices or security gateways, as we analyze throughout the article.

3.2. Security requirements

The integration of low-power WSN with the Internet will require appropriate security mechanisms, which are able to provide fundamental security assurances to WSN applications, devices and communications. As in classic Internet communications, the employment of end-to-end communications between low-power WSN sensing devices and other external or Internet entities will require appropriate security assurances in terms of *confidentiality*, *integrity*, *authentication* and *non-repudiation* of the transmitted data. End-to-end security may be addressed in the context of the communication protocol itself, or on the other end by external mechanisms. Another class of security requirements may relate to the exposure of WSN environments to Internet-originated threats and attacks. In this context, *availability* and *resilience* against such attacks will be important requirements for many IoT applications. Finally, we may also identify security requirements as *privacy*, *anonymity*, *liability* and *trust*, which are fundamental for the social acceptance of most of the future applications employing Internet-integrated sensing devices. It is important to note that a security architecture verifying the previously identified security requirements for Internet-integrated WSN is currently not completely defined. On the other end, future applications may require autonomous and unattended communications between devices in the absence of a security infrastructure. Despite the complexity of defining appropriate security mechanisms in this context, it is fair to consider that the design of mechanisms to support security in the context of Internet-integrated WSN may provide an important contribution to this goal.

3.3. Challenges in the usage of classic security mechanisms

In regard to the usage of existing Internet security mechanisms to protect Internet-integrated WSN, we are able to verify that the constraints of WSN devices and applications, which motivated previous research efforts on WSN security, will also apply to Internet-integrated WSN. This means that the limitations of low-power WSN devices in terms of critical resources such as memory, processing power and energy, will in practice also guide the design and adoption of highly optimized mechanisms to support communications and security on such environments. On the other end, even considering that at the same production cost of today's devices sensing platforms may support more resources in the future, security operations are likely still to be a burden in future sensing platforms, and its integration with the Internet may also promote new threats and attacks not encountered in WSN applications employing devices isolated from the Internet or external communication environments. Other aspect to consider is that of the heterogeneity of the applications and sensing devices. Most of the existing research proposals do not apply well to environments where heterogeneous devices are employed, or on the other hand must be adapted. Despite such challenges, we also observe that the design of security mechanisms in the context of a global communications and security architecture encompassing low-power WSN devices provides the almost unique chance to take into account security issues from the beginning.

3.4. Integration approaches

Our discussion proceeds with an analysis on the current proposals to address the integration of low-power WSN with the Internet [22,23]. The investigation of such approaches also helps in the identification of the open issues regarding security. We also note that different integration strategies may in practice serve different applications. Critical services available for example in SCADA industrial control environments may be exported to the Internet in a controlled fashion, for example via a security-enforcing gateway, while other applications may benefit from direct communications with sensing devices and implement more relaxed security policies. The various integration approaches also correspond to different degrees and strategies of integration of WSN with the Internet communications and security infrastructures, as we discuss next.

3.4.1. Cloud-based integration approaches

A currently popular integration approach is via cloud-based web services [24,25], and proposals in this category offer a platform as a service, in which the user may customize the tools provided in order to build a particular product. The main goal of the cloud-based integration approach is to facilitate the employment of high-performance computing and storage facilities in the processing of sensing data retrieved from WSN devices supporting distributed applications. The support of complex operations on this data may enable applications in diverse areas, for example for business intelligence purposes, and proposals in this area focus on the quick development of products, instead of the communications and security infrastructure.

Other characteristic of this integration approach is that it hides the communication technologies employed in the WSN domain from the outside. Existing proposals in this category usually employ tailor-made middleware solutions and Application Programmer Interfaces (API) designed according to the Service Oriented Architecture (SOA) principles. The middleware simplifies the development of new IoT applications, since it abstracts applications from the characteristics of the sensing devices and the complexity of the WSN communications. The sensing data retrieved from WSN sensing devices is transmitted to cloud servers via a gateway, which may also support operations such as data aggregation, protocol translation and remote management, among others. Some proposals in this category also virtualize physical sensors, with the purpose of enabling the employment of a single physical sensor by multiple applications. A virtualized sensor abstracts the physical sensor from its location or its particular characteristics and capabilities, and such proposals usually also support mechanisms to manage the service infrastructure and to publish the services available on the various devices using service templates [25].

Various industry and research proposals may be found that currently follow the cloud-based integration approach. Xively Could Services from LogMeIn [26], formerly called Pachube, is an IoT cloud service providing web-based tools and developer resources to facilitate the development and deployment of connected products using heterogeneous services. SensorCloud [27] from MicroStrain is a cloud-based data storage, visualization and remote management platform supporting user-programmable data analysis via a cloud-based math engine. SensorCloud may also be deployed using specialized WSDA (Wireless Sensor Data Aggregator) gateways that support data aggregation and remote management of the sensing devices and data. SensaTrack [28] offers a turnkey solution for monitoring services and supports a large variety of sensors and mobile devices, together with gateways supporting backhaul Internet communications using CDMA (Code division multiple access), GSM (Global System for Mobile Communications), Ethernet and WiMax communications. NimBits [29] is a free cloud-based service that may be used to record and share data on the cloud, and offers both a free service for data storage and sharing and a server platform available for users to deploy applications on their own servers. In this proposal the sensing data is sent in various formats to data points created in the cloud, and which may be configured to trigger calculations, alerts and statistics, among other operations of the data. Another free cloud-based integration product is ThingSpeak [30], which is an open source application offering an API to store sensing data on the cloud, and also to support numeric data processing operations on the data.

The integration of WSN with the Internet via cloud-based services provides a simple solution to the problem of quickly accessing the sensing data retrieved from one or various WSN deployments and processed according to the requirements of the applications. Despite the previously addressed advantages, this strategy provides a minimal integration with the Internet communications and security infrastructures. Internet communications only take place between WSN gateways and cloud servers, while WSN communications may be supported by proprietary communication protocols. The interconnection of different WSN applications using cloud-based services may also be difficult and require tailor made applications, rather than being enabled by standard Internet communication mechanisms as in other approaches. Nevertheless, this integration approach serves the purpose of providing a simple solution to the problem of accessing distributed sensing data in the context of Internet applications.

3.4.2. Front-end proxy integration approaches

An initial approach in the adoption of Internet communications facilitating the integration of WSN with the Internet communications architecture appeared in the form of research proposals employing a gateway that operates as a front-end proxy. The gateway entity is also present in other integration approaches, but in this case it is employed with the purpose of isolating and abstracting WSN communications from the Internet, while offering accesses to the services of sensing devices at the application-layer via a web services (WS) interface. Thus, this integration approach does not support direct end-to-end communications between WSN devices and other external devices or Internet hosts, as may be required by future sensing applications.

In the research proposals in this class, the proxy may obtain the data from sensing devices following two main strategies. One consists in the data being obtained from a sensing device upon the arrival of a request from an Internet client, and possibly caching the data at the proxy. The other is to employ a subscription/push protocol that enables sensing devices to update sensorial data on the proxy only when a change occurs, and in this case the proxy is required to maintain a local cache for all the relevant sensing data. The front-end proxy integration approach thus enables the indirect integration of a low-power WSN with the Internet at the application-layer, and has pioneered the employment of RESTful web services to support accesses to sensing data obtained from the WSN.

An initial research proposal in this integration class is discussed in [31,32], which describes an architecture where embedded sensing devices support web services and speak the HTTP protocol, although not supporting IP communications on the WSN, rather employing proprietary communications. The proposed architecture is also more recently discussed and evaluated in greater depth in [33]. Another proposal in this class is found in SenseWeb [34] from Microsoft Research. The proposed architecture also supports multiple gateways serving particular groups of sensing devices. SensorMap [35] is a popular practical implementation of the SenseWeb architecture, it mashes up sensing data from multiple sources on a map interface and provides interactive tools to selectively query sensors and visualize the data. SensorMap also supports authenticated accesses to sensor management functionalities. We also observe that, since SenseWeb and SensorMap support a platform to share and support computations over the sensorial data, this this sense preclude the more recent cloud-based integration proposals.

In [36] the authors propose the integration of a WSN with the Internet also via a WS API supported by a front-end proxy, which provides a virtual counterpart of WSN physical sensing devices on the Web. In this work the authors also identify the interest of supporting web services directly on sensing devices in the future. The interconnection of a WSN with the Internet via mobile communication networks is explored in [37], particularly using GPRS communications. This research proposal also employs a specialized gateway to support protocol conversion and mechanisms to control WSN sensing devices.

As previously analyzed, we may observe that the front-end proxy integration approach enables the indirect integration of WSN with the Internet at the application-layer via WS. This approach enables a standard communications interface from the point of view of external Internet clients, while in the WSN domain communications may be supported by proprietary solutions. Therefore, from the point of security this integration approach isolates WSN communications from Internet-originated threats and attacks, as applications delegate all Internet-related communications to the front-end proxy. In the same context, such proposals also do not support standard Internet communications between sensing devices on the same or different WSN domains. The proxy may also be configured to behave as a normal Internet citizen, thus supporting standard Internet security mechanisms to protect communications with other external or Internet hosts. As in cloud-based integration proposals, no Internet communication technologies are designed for the WSN domain, and as such the Internet communications and security infrastructures are not extended to encompass WSN devices and applications.

3.4.3. Architecture frameworks

Various research projects target the design of architecture frameworks supporting different strategies to enable communications between various WSN domains across the globe, over the Internet. As with the previous integration approaches, most of such proposals employ a middleware to abstract operations on sensing devices and data from the particularities of WSN communications. In consequence, rather than focusing on the design of Internet communication mechanisms for WSN environments, such proposals instead provide complex applications over distributed WSN islands, which are interconnected via the Internet.

One initial architecture framework in this class was designed in the SENSEI research project (Integrating the Physical with the Digital World of the Network of the Future) [38]. SENSEI was an Integrated Project in the EU's Seventh Framework Programme (FP7), in which 19 partners from 11 European countries participated and that ended on December 2010. This project targeted the design of an architectural framework and of protocol solutions to enable the easy plug and play integration of a large number of globally distributed WSN into a global system, and providing support for operations such as network and information management, security, privacy, trust and accounting. In order to enable interoperability of sensing devices on different WSN domains, SENSEI supports RESTful communications in the WSN parts of the system. An extensive set of security mechanisms were also designed in this project, namely to support secure code updates, jamming mitigation, secure routing, and detection of node capture and replication. The SENSEI architecture also supports the employment of a trusted hardware component to defend against a broad range of security threats resulting from compromise attacks, and introduces the middleware component FAIR [39] providing resilient in-network data processing.

SmartSantander [40] is a city-scale experimental research facility that supports typical applications and services for a smart city. This project builds on results from SENSEI [38] and on WISEBED test bed facilities [41]. WISEBED is a research effort of nine academic and research institutes across Europe aiming to provide a multi-level infrastructure of interconnected test beds of large-scale WSN for research purposes. The architecture of SmartSantander supports the controlling of sensing devices through a set of low-level API, running experiments through a web portal and application support through web services. The IoT-A project [42] builds on the results from the previous projects and targets the design of an architectural reference model for the interoperability of IoT systems. Among the goals of this project are the outline of principles and guidelines for the technical design of protocols, interfaces and algorithms, the design of mechanisms for the integration of the proposed architecture into the service layer of the future Internet, a novel resolution infrastructure, novel platform components and the experimentation of the proposed mechanisms using real implementation scenarios.

In conclusion, the previously analyzed architecture frameworks are designed with the main purpose of enabling complex and rich sets of services based on distributed WSN domains. Rather than designing mechanisms to enable Internet communications over such domains and between constrained sensing devices on a given WSN domain, they instead employ middleware approaches, and Internet communications serve the single purpose of supporting communications between different WSN gateways. As such gateways in practice support RESTful web services for data retrieved from sensing devices, new applications developed in the context of such research projects can, in the future, evolve to support WSN devices employing 6LoWPAN-based Internet communication technologies, which we analyze next.

3.4.4. Integration via standard Internet communication protocols

Contrary to the previous integration approaches, communication technologies are currently being designed based on 6LoWPAN [17–19] that will facilitate the complete integration of WSN with the Internet communications and security infrastructures, as we proceed to discuss. In this integration approach, mechanisms may be designed to support or adapt Internet communications and security technologies to WSN environments.

As previously noted, standard mechanisms designed by working groups of the IETF are particularly relevant in this context. The IETF is an international community of network designer, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. Outside the IETF, relevant standardization activities are also being conducted in other organizations. The ETSI (European Telecommunications Standard Institute) TC (Technical Committee) on M2M communications [43] is working to develop an end-to-end high-level architecture for M2M and standard fulfilling the gaps where other standards bodies or groups are unable to do so. On the other end, the ITU-T (Telecommunication Standardization Sector of the International Telecommunication Union) [44] is working on recommendations related to USN (Ubiquitous Sensor Networks) and NGN (next generation networks), with the goal of designing a conceptual network built over existing physical networks, which provides knowledge services by making use of sensing data.

As previously discussed, a group of protocols based on 6LoWPAN is currently being designed at various working groups of the IETF, which will enable Internet communications on low-power WSN environments. This means that such protocols may provide the technological basis to extend the existing Internet communications architecture to encompass WSN applications. The 6LoWPAN-based communication mechanisms are also based on the employment of a gateway to support communications between the WSN and Internet domains, and enable the identification of the reference integration architecture that we illustrate in Fig. 2. This figure illustrates the employment of 6LoWPAN-based communication technologies on the WSN gateway and also on sensing devices.

We also note that the integration architecture illustrated in Fig. 2 supports the previously discussed integration approaches. From the perspective of the communication technologies employed, such approaches are based on the usage of a web services interface supported by the WSN gateway, and/or on the employment of a customized middleware solution. As also illustrated in Fig. 2, 6LoWPAN-based technologies enable direct end-to-end communications between WSN devices and external/Internet hosts, as will be beneficial to enable communications in the context of future distributed sensing applications. This architecture also enables the employment of the previously identified communication technologies, at the capillary (low-power WSN), backhaul and backbone categories.

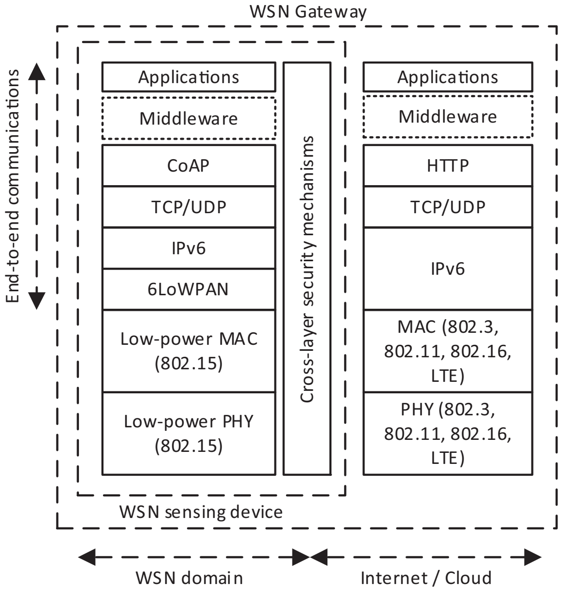

Fig. 2. Reference architecture for the integration of WSN with the Internet via 6LoWPAN-based communication technologies.

The adoption of 6LoWPAN-based communication technologies to support the integration of WSN with the Internet is also starting to become visible in commercial offerings. For example, the popular ZigBee-2006 [45] specification is currently evolving to support IPv6 by adopting 6LoWPAN, using the so-called ZigBee IP stack [46] that will also support RPL and CoAP. Despite the adoption of a standards-oriented stack, we may note that ZigBee communications over the Internet remain restricted to ZigBee applications, and thus Internet interoperability is not a current goal of this closed specification. Other products from companies such as Sensinode [47] also adopt IP-based 6LoWPAN communications. Sensinode was recently acquired by ARM [48] and offers the NanoStack 6LoWPAN protocol stack, together with and the NanoRouter platform, to support applications requiring 6LoWPAN-Internet routing infrastructures. The integration architecture illustrated in Fig. 2 can also be employed with multiple gateways, as in existing research works focusing on the interconnection of WSN with the Internet with fault-tolerance and redundancy [49–51]. Also, the strategic placement of the gateway in respect to the communications infrastructure can be useful in supporting security mechanisms.

We observe that this integration approach was initially addressed in the design of mechanisms to enable communications with web services running directly on constrained WSN sensing devices [52,53]. Such proposals precluded the full integration approach that is currently becoming a reality thanks to 6LoWPAN-based communication technologies, supported by the integration architecture illustrated in Fig. 2. The employment of web services on constrained sensing devices was initially proposed and evaluated in [52], and in [53] the authors describe a RESTful web service architecture allowing external servers to communicate directly with IP-enabled sensor devices using web services, over the TCP transport protocol. The architecture proposed in this work is based on a session-aware power-saving MAC protocol running over X-MAC [54], which synchronizes wake-up periods with TCP control messages, and on the employment of the HTTP conditional mechanisms to avoid the transmission of non-changing data from the server to the client. Stream Feeds [55] identifies streams of data from sensing devices using URLs that may be hyperlinked to other objects on the web, thus enabling such streams to be indexed by search engines.

We are able to observe that this integration approach enables the full integration of WSN communications with the Internet, while it can certainly be more complex in respect to security. As end-to-end communications with sensing devices may take place from the network-layer up, such devices may be more open to a plethora of threats and attacks originated at the Internet, if not appropriately protected. As most WSN devices are expected to remain constrained, end-to-end communication and security technologies must be employed parsimoniously, and be complemented by appropriate security mechanisms supported by more resourceful devices. In reality, end-to-end security is only part of the problem, as many cross-layer security aspects will also require appropriate solutions, as previously discussed. 6LoWPAN-based communication technologies are discussed in greater detail in the context of our analysis on security in the various integration approaches, later in the survey.

4. Security for Internet-integrated WSN

Our following discussion focuses on how security may be addressed in the context of the previously identified integration approaches, and of its enabling communication technologies. We analyze the security mechanisms designed to protect communications using such technologies, and also identify the open issues and research challenges in this context.

4.1. Security in the cloud-based integration approach

Regarding the previously discussed cloud-based integration proposals, we are able to observe that they lack, in general, important security mechanisms and assurances, as we proceed to analyze. In Xively [26] devices write and read data from cloud-based applications using various API provided by the platform. One security service provided is the secure provisioning of devices for their initial boot up in the context of a given application. Each device is provisioned with a Feed Identifier and an API key to be able to send data to the cloud-based application, after contacting a device activation API. In order to obtain the Feed ID and API key, the device submits a secure activation request constructed by producing an HMAC-SHA1 hash of the device's serial number plus a secret key associated with the application in the context of which the device is being activated. After this initial procedure, the feed ID and the API key are stored also on the application. At the end, the feed ID enables the device and the application to communicate with each other and with Xively. Xively also employs keys to control accesses to all API resources. A key may be associated with a particular permission of accessing a resource or feed, also by a particular user or machine with a particular IP address. Keys are sent in API requests, either in the HTTP request header or as part of the URL. Of course, the usage of keys in this way is inherently insecure if not using encryption, and Xively also supports TLS/SSL to support end-to-end confidentiality, integrity, authentication and non-repudiation for communications between sensing devices and the cloud servers, while we must notice that HTTPS is optional.

The support of TLS/SSL with the same purpose in SensorCloud [27] is mandatory across the platform, including for data uploads and downloads via HTTPS, as in the previous proposal. The platform also provides mechanisms to state who is authorized to access sensing data stored on the cloud, in the context of a given application. All sensing data is private by default, and data owners can also send invites to other users they want to bring to the application, for example to assist in analyzing and building custom-tailored data processing applications. SensaTrack [28] provides mechanisms for the setup of user accounts and corresponding security access privileges, and some of the provided gateways also support IPSec VPN accesses to the cloud servers, in order to support confidentiality, integrity, authentication and non-repudiation for end-to-end communications with such servers.

Regarding free cloud-based integration solutions, NimBits [29] also supports HTTPS protected requests to web services. Access keys can also be created and employed in access URLs to get access to protected resources. The administrator of a given application may create a key and associate it with a particular data point or with all of his data points. Access permission may also determine read-only accesses, rather than read and write accesses. ThinkSpeak [30] supports management of privileges to control accesses to data, as well as to define who is able to build and use applications, providing control of accesses to data and applications considered private. In this proposal data channels are used to store and retrieve data, and each channel has private and public views. Accesses to the private view are controlled via authorized accesses to the web server, while the public view is what other viewers see when they visit the channel. The administrator of a channel is able to define the information that is available on each view, customize the view with plugins, or even disable the public view. Accesses to resources may also be controlled via write or read API keys. By default, a channel is private and requires a read API key to access its feed. ThingSpeak also supports HTTPS accesses to API web services.

In Table 1 (at the end of the article) we resume the properties of the previously analyzed integration proposals in respect to security. We may observe that, despite the support of security mechanisms to protect end-to-end communications between sensing devices and cloud servers as previously discussed, cloud-based integration proposals do not address important security requirements such as privacy, trust and anonymity, among others. Such proposals also do not support the secure integration with other data sources, and as such are not employable in the context of future sensing applications integrating heterogeneous WSN domains and sensing devices. As previously discussed, this integration approach is a simple strategy to support applications that require the retrieval, storage and processing of large sets of sensorial data but, other than supporting web services communications at the application-layer, do not adopt other communication technologies that would facilitate its integration with the Internet communications and security architectures.

4.2. Security in the front-end proxy integration approach

Most of the research proposals interconnecting WSN with the Internet do not consider security to be a major concert. For example, research proposals using web mashups [31–33] focus on device abstraction and on making sensing data available via a simplified web services API, while not addressing particular security threats nor the design of security mechanisms. SenseWeb [34] identifies the importance of addressing security issues as the trustworthiness of the data, the privacy of the users and the reliability and verifiability of the shared data against malicious intervention or inadvertent errors. This work also discusses the challenges in terms of security and trust of building a sensing infrastructure out of shared resources, but does not propose nor define any specific mechanisms to target such aspects. In [36,37] the authors identify the design of security management functions for the proposed IoT gateway as future work.

In general, we may fairly consider that the exploratory nature of the previous research proposals motivated a focus primarily on the communication aspects of the proposed solutions, rather than on its security requirements. On the other end, we also observe that such proposals have provided an important contribution in paving the way to the acceptance of the viability of the interconnection of constrained low-power WSN with the Internet, even if indirectly via services supported by a front-end proxy. This integration strategy is also present in specialized architecture frameworks and supports the integration via standard Internet communication technologies developed for WSN environments, as previously discussed.

4.3. Security in architecture frameworks

As previously analyzed, various architectures have been designed supporting different approaches to integrate a WSN with the Internet, in the context of commercial approaches and research projects. As we have observed, most of such proposals employ specialized middleware layers designed with very-specific purposes, instead of supporting generic Internet communication mechanisms that are able to support heterogeneous applications based on Internet-integrated WSN. A consequence of this design approach and of the purpose of such proposals is that security is designed according to the particular goals of each project, as we proceed to address.

The SENSEI architecture [38] introduces the notion of a community, which is formed by various actors taking up one or more business roles. Actors may be resource providers (the owners of the resources), framework providers (the owners of framework components), service providers (the owners of the services that use the resources and support services), or resource users (who are the main users of such resources and services). The proposed framework also offers community management functions, which include user account management, identity management, security and privacy functions, among others. In order to support secure interactions between different SENSEI members, the architecture supports authentication, authorization and accounting (AAA), as well as privacy and trust management. In particular, the AAA architecture of SENSEI supports a security token service (STS), which provides entities with the security assertions (tokens) required to access resources on the network. The auditing and billing service supports accounting in the context of the AAA architecture, while the resource access proxy service supports authentication, token request and resource access on behalf of the user. Regarding privacy, the SENSEI architecture addresses real world privacy issues and electronic privacy issues. The former includes the privacy of personal information collected by sensors, and access to this information is controlled by use of the AAA architecture previously described. Electronic privacy issues include people leaving digital traces of their movement and actions in various places, and the architec-

Table 1
Security properties of cloud-based integration proposals.

Integration proposal	End-to-end security properties for communications with cloud servers	Secure provisioning of devices	Access control mechanisms	Security integrated into the API
Xively [26]	Confidentiality, integrity, authentication and non-repudiation via TLS/SSL (optional)	Supported via authentication hashes	Supported, by user and IP client address	Supported, using access control keys
SensorCloud [27]	Confidentiality, integrity, authentication and non-repudiation via TLS/SSL	Not supported	Supported, by user	Not supported
SensaTrack [28]	Confidentiality, integrity, authentication and non-repudiation via IPSec VPN (optional)	Not supported	Supported, by user	Not supported
NimBits [29]	Confidentiality, integrity, authentication and non-repudiation via TLS/SSL (optional) (TLS/SSL)	Not supported	Supported, by user and data points	Supported, using access control keys
ThingSpeak [30]	Confidentiality, integrity, authentication and non-repudiation via TLS/SSL (optional) (TLS/SSL)	Not supported	Supported, by user and public or private cryptographic keys	Supported, using access control keys

ture provides a range of features to allow users to control how difficult it is to link their traces to them, for example the use of pseudonyms or attributes instead of recognizable identities. The SENSEI research project also produced work regarding the secure programming of sensing devices [56–58], resilient in-network data processing [39] and mechanisms against capture attacks [59].

One of the goals of the SmartSantander [40] project is to implement and evaluate security as one of the key building blocks of the IoT architecture. The architecture currently being designed includes security requirements related with the AAA (authentication, user account management and authorization) model. Trust and privacy requirements are also being considered in the context of session management in test bed servers, gateways and sensing devices. Researchers may access the test bed provided by the project via a specialized web portal, and the control of authorizations and accesses to the test bed is supported both by this portal and in the set of low-level API supported by the architecture. The administrator of the experimental facility will be able to grant and revoke user access privileges. As we have previously discussed, SmartSantander is an ongoing research project and as such work related with the design of appropriate security mechanisms is ongoing and results may be expected in the future.

Regarding security, the main outputs of the IoT-A [42] research project are related with the resolution infrastructure being designed to allow scalable look up and discovery of resources, entities, and their associations. In the context of this project, the project is designing mechanisms to support privacy and security in the resolution infrastructure. The original architecture was extended with a security component to ensure privacy and security for the resolution functions, as well as to offer the basis for other security functionalities outside the resolution infrastructure. A set of components are introduced in the IoT-A architecture to support security, namely an authorization component to perform access control decisions based on access control policies, an authentication component, an identity management component that manages pseudonyms and accessory information to trusted subjects so that they can operate anonymously, and a key exchange and management component. The project is also designing a trust and reputation architecture and the relationships of the various security-related components to the other mechanisms of the architecture. Table 2 at the end of the article resumes the main characteristics in terms of security of the previously analyzed architecture framework integration proposals. In conclusion, we may observe that the mechanisms developed in the context of the previously analyzed architecture frameworks may provide useful guidance in the design and adoption of future IoT applications and mechanisms, even if the proposed solutions currently do not support Internet communications on WSN environments.

4.4. Security with integration via standard Internet communication protocols

Before discussing security in the technologies enabling the full-integration integration architecture illustrated in Fig. 2, we analyze previous research proposals that have contributed to the acceptance of this integration approach. In general, the exploratory nature of such works motivates the absence of appropriate security solutions, as we proceed to discuss. The research proposals in [49,50] do not address security, instead focusing on the intelligent placement of gateways in order to reduce data latency in scalable and sustainable WSN deployments. In [51] the authors discuss the secure interconnection of a WSN with the Internet and propose an integration model to be employed both on sensing devices and the gateway. This model supports end-to-end secure communications at the network-layer and the usage of multiple gateways to support distributed mechanisms as intrusion detection, authentication and key management, although not proposing any specific solutions to target such aspects.

A few initial research proposals focused on the integration of WSN with the Internet via web services. Contrary to proposals belonging to the front-end proxy integration category, in this case the main goal is to explore the usage of web services directly on constrained sensing devices. As in the previous works, we also observe the lack of appropriate security solutions in such proposals. In [52,53] the authors focus only on the communication aspects of the integration. In Stream Feeds [55] the authors discuss that applications are able to inherit security mechanisms supporting authentication and privacy services from the web services technology employed in the Internet, but does not specify how this may be achieve in practice. In conclusion, we observe that such previous works are exploratory of the integration of WSN with the Internet via standard Internet communication protocols. Given the preliminary nature of such proposals, security is either absent or mostly undefined.

4.4.1. Support technologies

The communication technologies supporting standard Internet communications on low-energy WSN environments are being designed over IEEE 802.15.4 [16]. IEEE 802.15.4 sets the rules for communications at the lower (MAC and PHY) layers on LoWPAN environments, and lays the ground for standard IoT communication and security mechanisms designed for higher layers. Among the various recent addendums to the standard, IEEE 802.15.4e [60] is particularly interesting for applications in critical environments employing Internet-integrated low-power WSN devices, as it supports communications for applications with strict timing requirements. Applications with such characteristics can thus also be supported with the technologies previously discussed and illustrated in Fig. 2.

IEEE 802.15.4 supports link-layer (hop-by-hop) communications in low-power WSN and security with AES/CCM cryptography at the hardware, as currently available in reference wireless sensing platforms such as the TelosB [61]. In Fig. 3 we illustrate the usage and availability of payload space for Internet communication protocols designed for IEEE 802.15.4 low-power WSN environments. As we may clearly observe, payload space is a scarce resource in such environments, and in consequence upper-layer communication mechanisms are required to employ efficient compression mechanisms whenever possible. Fig. 3 also illustrates the requirements of payload

Table 2
Security properties of integration architecture frameworks.

Architecture framework	User and privilege management	Privacy and trust management	AAA (Authentication, Authorization and Accounting)	Other security properties
SENSEI [38]	Supported	Supports privacy via user pseudonyms	Supported (via security tokens)	Confidentiality, integrity, authentication and non-repudiation for end-to-end communications; secure device reprogramming; secure data aggregation; resistance against sensing device capture attacks
SmartSantander [40] (ongoing work)	Supported	Supports trust and privacy planned for all components	Supported	Confidentiality, integrity, authentication and non-repudiation for end-to-end communications
IoT-A [42]	Supported	Supports user privacy via pseudonyms, and privacy on resolution mechanisms	Supported	Confidentiality, integrity, authentication and non-repudiation for end-to-end and hop-by-hop communications; key exchange and management; reputation management

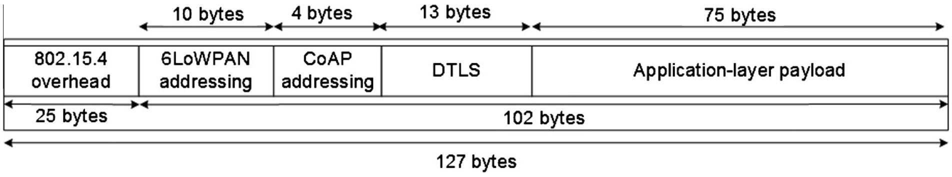

Fig. 3. Payload space usage and availability in low-power WSN 6LoWPAN environments.

space of the various 6LoWPAN-based communication technologies.

The payload space available for upper-layer communication protocols depends on the overhead introduced by addressing and security at the IEEE 802.15.4 link-layer. Addressing requires 25 bytes and security in IEEE 802.15.4 supports three modes: AES-CCM-128, AES-CCM-64 or AES-CCM-32. Such modes produce a Message Integrity Code (MIC) of 128, 64 and 32 bits respectively, and in consequence require 21, 13 and 6 bytes of payload space [16]. We may note that despite the availability of security at the link-layer, communications protocols at higher layers of the stack may in practice dispense it and instead support new mechanisms designed to enable end-to-end security. On the other end, efficient AES/CCM encryption at the hardware is very useful in supporting security mechanisms at higher layers, because it provides an efficient cryptographic basis for the usage of such mechanisms. The employment of AES/CCM with this purpose involves its usage in the standalone mode, rather than in the inline mode. Encryption in the standalone mode separates security processing from link-layer packet transmission or reception. The usefulness of AES/CCM in the standalone mode also enables us to consider the interest of investigating the implementation of hardware optimizations at the hardware of sensing platforms to support other security and cryptography operations, for example public-key cryptography and tamper-proof storage mechanisms.

AES/CCM as supported by IEEE 802.15.4 devices uses 128-bit keys to provide confidentiality, data authenticity, integrity and replay protection for the data packets exchanged at the link-layer between two communicating entities. Confidentiality is achieved with AES in the Counter (CTR) mode, while data authenticity is achieved using the Cipher Block Chaining (CBC) mode to produce a Message Integrity Code (MIC) that is appended to the data to be transmitted. The CTR and CBC modes are jointly supported by the combined Counter with CBC-MAC (CCM) encryption mode of AES. In particular, IEEE 802.15.4 sensing platforms support the CCM* variant, which supports also integrity-only and encryption-only.

In addition to the previous security assurances, IEEE 802.15.4 also supports access control mechanisms via access control information maintained in an ACL (access control list) table. The MAC uses the source and destination addresses of the frame to lookup the correct ACL, which contains information on how to process security for the packet. Each ACL entry contains information on the security suite to protect the frame (one of the suites available in the standard), the cryptographic key to use with encryption/decryption using AES/CCM (for suites supporting encryption) and the nonce (IV) value preserved across packet encryption invocations. The same entry may also store a high water mark of the most recently received packet identifier, for replay protection purposes. The ACL tables can also store a default ACL entry with information that is employed to process security in the absence of more detailed security entries.

The IEEE 802.15.4e [60] addendum to the standard introduces small modifications to adapt MAC security mechanisms to time-synchronized channel-hopping communications, by adapting replay protection and semantic security to time-synchronized communication networks as supported by IEEE 802.15.4e. We may also observe that the information stored in the ACL table is not managed by the MAC layer, rather is the responsibility of the application. Therefore, even if not using AES/CCM link-layer security, applications can efficiently manage and store such information using the MAC security services and the memory of the sensing device.

Despite the maturity of the IEEE 802.15.4 [16] standard, a few limitations may be identified in respect to how it implements security services at the MAC layer. One limitation is the absence of a keying model, as the standard assumes that key management operations are the responsibility of the application. The ACL table also does not provide adequate support for all keying modes, in particular group keying and network-shared keying. Group keys are hard to implement, since each ACL entry may only be associated with a single destination address, and thus require that various ACL entries use the same key, which may promote nonce reuse and the consequent breaking of confidentiality. Network-shared keying is incompatible with replay protection, as this mode may be supported only using the default ACL entry, and in this situation the transmitter nodes would have to somehow coordinate their usage of replay counter space.

As the integration of WSN with the Internet will enable end-to-end communication mechanisms from the network-layer up, appropriate key management mechanisms will be required and, given the limitations of IEEE 802.15.4 in this respect, may be implemented as cross-layer protocols not depending on the usage of the information stored in the ACL table at the MAC. Other limitation is related with the management of IV values at the MAC, which may be problematic in case 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, which is potentially dangerous with stream ciphers such as AES/CCM that encrypt in the CRT mode, as it may enable an adversary to recover plaintexts from cipher texts. The reuse of nonce values may also be a consequence of the loss of ACL state after a power interruption, or when a node wakes up from a low-power mode. A final limitation of the standard in respect to security is that it does not specify security for link-layer acknowledgment messages.

Overall, the previously discussed limitations may be addressed in future versions of the standard but, in the context of the integration of low-power WSN with the Internet, are being addressed at higher layers of the communications stack, as we discuss next. As previously discussed, standalone AES/CCM hardware encryption provides an efficient cryptographic basis for security mechanisms designed to protect communications at higher layers of the stack, as we may encounter in many research proposals currently available in the literature and analyzed next.

4.4.2. Network-layer low-energy communication and security mechanisms

Although IP communications were once considered impractical for LoWPAN environments, the IPv6 over Low Power Personal Area Networks (6LoWPAN) working group [62] of the IETF (Internet Engineering Task Force) has radically changed this perception. 6LoWPAN designs an adaptation layer [16–18] employed between the link and network layers, which enables the transportation of IPv6 packets over IEEE 802.15.4 environments. This is achieved through the efficient compression of the addressing and control information inside IPv6 packets. 6LoWPAN header compression uses information from the link-layer and also from the transport-layer, currently supporting only UDP. The adaptation layer also implements IPv6 packet fragmentation in order to support the IPv6 minimum MTU requirement of 1280 bytes. Overall, 6LoWPAN is an excellent example of how cross-layer optimizations may enable the employment of standard Internet communication technologies on constrained low-power WSN environments.

The mechanisms implemented in 6LoWPAN are described in RFC 4919 [17], which discusses the group's general goals and assumptions, and on RFC 4944 [18] and RFC 6282 [19], which describes the mechanisms of the adaptation layer and header compression, respectively. 6LoWPAN also supports the fundamental networking procedures required for the operation of WSN sensing devices on IPv6 environments, in particular neighbor discovery (ND) and address auto-configuration. We may also note that, although 6LoWPAN currently also supports IEEE 802.15.4 link-layer communications, other technologies will be adopted in the future. This is also visible in recent Internet Drafts (I-D) proposing the adaptation of 6LoWPAN to other technologies such as Bluetooth Low Energy (BLE) [63], Digital Enhanced Cordless Telecommunications Ultra Low Energy (DECT-ULE) communications [64] and ITU-T G. 9959 [65]. 6LoWPAN thus represents an important convergence technology supporting an increasingly growing ecosystem of PHY/MAC communications technologies optimized for particular capillary communication environments and applications. On the other end, devices such as RFID tags are unable to support 6LoWPAN, and currently require different approaches to security. Such devices may either evolve to also support 6LoWPAN in the future, or on the other end be supported by future standard communication mechanisms designed in the context of the an integration stack such as the one illustrated in Fig. 2.

The 6LoWPAN adaptation layer enables different header compression scenarios, according to the scope of the communications employed. Using IPHC shared-context header compression [19] an UDP/IPv6 header may be compressed down to 10 bytes, as illustrated in Fig. 3. In this situation, the prefix of the source and destination addresses can be compressed, and the same applies to the source address IID (Interface Identifier) in case it is derivable from the link-layer header. Since 102 bytes are available at the link-layer, this scenario results in the availability of 92 bytes of payload space for protocols and applications at higher layers. As the adaptation layer operates below the network-layer, 6LoWPAN mechanisms are in practice transparent to upper-layer protocols.

The employment of security mechanisms in the context of the 6LoWPAN adaptation layer could enable transparent end-to-end network-layer security in WSN environments, which could be a helpful resource to communication protocols and applications at higher layers of the stack.

Please cite this article in press as: J. Granjal et al., Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.08.001

Despite the advantages of 6LoWPAN security, we verify that no particular mechanisms are currently adopted as standards in the context of the adaptation layer at this time. 6LoWPAN security was initially discussed in the form of an I-D [66] that, while not proposing any particular approaches or security mechanisms, discusses and identifies the main difficulties in adopting standard network-layer solutions such as IPSec and IKE in constrained 6LoWPAN environments. Similar challenges are also identified and discussed in previous research contributions [128,129]. RFC 4919 [17] discusses the importance of addressing security at various complementary protocol layers of the IoT communications stack, and that the most appropriate approach may depend on the application requirements and on the constraints of particular devices. RFC 4944 [18] identifies the possibility of forging or accidentally duplicating EUI-64 interface addresses, which may consequently compromise the global uniqueness of 6LoWPAN interface identifiers. This RFC also discusses the importance of protecting Neighbor Discovery and mesh routing operations against security threats. RFC 6282 [19] focuses on the security issues posed by the usage of a mechanism inherited from RFC 4944 to compress a particular range of 16 UDP port numbers down to 4 bits. This document discusses that the overload of ports in this range may increase the risk of an application getting the wrong type of payload or of an application misinterpreting the content of a message, if employed with applications not honoring the reserved set for port compression. This RFC recommends that the usage of such ports be associated with a security mechanism employing integrity codes. RFC 6568 [67] discusses the design and application spaces for 6LoWPANs, and regarding security it focuses on the possible approaches to adopt security mechanisms in the adaptation layer, in the light of the characteristics and constraints of wireless sensing devices. This document discusses the threats due to the physical exposure of sensing devices, and on how wireless IEEE 802.15.4 communications may facilitate attacks against the confidentiality, integrity, authenticity and availability of 6LoWPAN devices and communications. RFC 6775 [68] proposes optimizations to enable Neighbor Discovery (ND) operations in 6LoWPAN environments, and includes a discussion on the threat model applicable to IPv6 ND operations, defined by RFC 4861 [69], as also being applicable to 6LoWPAN environments. This includes the proposal of adapting the SEcure Neighbor Discovery (SEND) [70] and Cryptographically Generated Addresses (CGA) [71] mechanisms to 6LoWPAN low-power WSN environments.

Regarding research proposals targeting 6LoWPAN security, in [72] the authors propose new compressed security headers for the 6LoWPAN adaptation layer, namely compressed versions of the AH (Authentication Header) [73] and ESP (Encapsulated Security Payload) [74] headers currently defined in the Internet Protocol Security (IPSec) [75] suite. This proposal is theoretically evaluated in [72], while in [76] the same authors discuss its experimental evaluation. This proposal considers the employment of cryptographic suites with different algorithms and keys of different sizes, and also of security in the tunnel and transport modes defined in IPSec [75]. The authors discuss that a proposal of this type can enable transparent end-to-end network-layer security involving constrained low-power WSN devices, similarly to what may be achieved using IPSec. End-to-end security in this context provides confidentiality, integrity, data authenticity and non-repudiation, also considering the usage of AES/CCM encryption in the standalone mode to support encryption and decryption at the 6LoWPAN adaptation layer. A more recent proposal with the same goal using 6LoWPAN shared-context IPHC header compression is described and experimentally evaluated in [77]. The experimental evaluation of this proposal and its comparison against IEEE 802.15.4 link-layer security is discussed by the same authors in [78]. The employment of network-layer security by adapting IPSec to the IoT is also discussed and supported in a recent I-D [79] submitted for discussion at the 6LoWPAN IETF group. We may observe that, despite the existence of proposals evaluating and defending the effectiveness of security in the context of the 6LoWPAN adaptation layer, mechanisms as the previously discussed are currently not officially part of 6LoWPAN, and therefore further work must be conducted in this area.

Other than network-layer end-to-end security, other research contributions have been published addressing 6LoWPAN security. Authors in [80] discuss the consequences of packet fragmentation attacks against the 6LoWPAN fragmentation and reassembly mechanisms. Such mechanisms render buffering, forwarding and processing of fragmented packets challenging on resource-constrained devices, and consequently a malicious or misconfigured node that is able to send forged, duplicate or overlapping fragments may threat the normal functioning or the availability of such devices. Such security threats could be addressed with the employment of appropriate authentication mechanisms at the 6LoWPAN adaptation layer, since recipients currently are unable to distinguish undesired fragments from legitimate ones when performing packet reassembly [80]. The effects of such attacks include receiving buffer overflow and misusage of the available computational capability, among others. The authors also propose the addition of new fields to the 6LoWPAN fragmentation header, namely of a timestamp providing protection against unidirectional fragment replays and of a nonce providing protection against bidirectional fragment replays, in order to deal with such threats.

Also in the context of fragmentation attacks, a more recent contribution [81] proposes the employment of mechanisms supporting per-fragment sender authentication, and to purge messages from the receiver's buffer for transmitter devices that are 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. Similarly to the proposals previously discussed addressing end-to-end network-layer security in 6LoWPAN, mechanisms of this type would have to receive acceptance from the community and be formally adopted as part of the 6LoWPAN adaptation layer. In the future, as 6LoWPAN-based technologies become accepted and employed to support applications employing Internet-integrated low-power WSN, we may expect that security issues such as the previously discussed will call for further research efforts in designing and adopting new security mechanisms in the context of the adaptation layer.

4.4.3. Transport-layer low-energy communication and security mechanisms

The 6LoWPAN adaptation layer currently supports only UDP [82] transport-layer communications, although it is possible to envision the support of TCP [83] in the future. UDP is currently the adopted transport-layer protocol for 6LoWPAN due to its simplicity and low impact on the limited packet payload space available at the adaptation layer. In consequence, the current approach to address transport-layer security in 6LoWPAN environments is to use the Datagram Transport Layer Security (DTLS) [84] protocol. DTLS is in practice the Transport Layer Security (TLS) [85] protocol with added features to deal with the unreliable nature of transport-layer communications. DTLS is also the current approach to address security with the CoAP [20] application-layer protocol, as we discuss later.

Despite the adoption of DTLS as previously discussed, the effectiveness of its employment in constrained low-energy WSN environments is currently not consensual among researchers in the area. In consequence, research is currently targeting the investigation of the impact of DTLS in WSN constrained sensing devices, and the design of mechanisms to adapt or optimize the protocol for such constrained environments. Other aspects being currently investigated include the impact of ECC public-key cryptography on sensing platforms to support authentication and key agreement as required for CoAP, the design of mechanisms to support the online verification of the validity of certificates, the modification of DTLS to support multicast communications and group keying, and the usage of DTLS with reverse proxies as enabled by the CoAP application-layer protocol. We observe that research efforts addressing the previous issues currently follow two complementary lines of research. One consists in considering the modification or optimization of the DTLS protocol to cope with the constraints of existing wireless sensing platforms, and the other the usage of alternative approaches to support security with the CoAP protocol, which we analyze later in the context of application-layer security.

The problem of the impact of DTLS on the resources of constrained wireless sensing platforms has recently motivated the formation of the DTLS In Constrained Environments (dice) working group of the IETF. Various features of the protocol have also been identified in the literature that may complicate its adoption in low-power WSN environments integrated with the Internet. Such features are discussed in a recent I-D [86], which identifies features of the protocol that may not be appropriate to low-power WSN environments. The identified problems are related to the employment of large messages in the handshake, which may cause fragmentation at the 6LoWPAN adaptation layer, and to the cost of computing the *Finished* message at the end of the handshake at the client and server devices. Fragmentation implies that the retransmission and the reordering of messages from the handshake may result in added complexity. Research approaches to such issues may include the design of appropriate reliability mechanisms to support the transportation of DTLS handshake messages, or of alternative transport-layer approaches to security.

In [87] the authors also identify two open issues when using DTLS to support Internet-integrated WSN, one being the inexistence of mechanisms for mapping between TLS and DTLS at an interconnection gateway, and the other that DTLS is currently unable to support multicast communications. Secure multicast communications may be required by many applications employing Internet-integrated WSN devices, and will also require the establishment of appropriate session keys among the participating devices. This can be achieved with an external key management solution appropriate to CoAP and DTLS, or by modifying the DTLS handshake to support session key negotiation for a group of devices. In the context of such an approach, the DTLS record layer may also be modified to enable secure CoAP group communications with confidentiality, integrity and replay protection, as proposed in [88]. This proposal does not address how the required group keying material is negotiated, particularly the client and server read and write MAC keys, encryption keys and IV values. The design of mechanisms to support the negotiation of such security-related parameters prior to normal multicast communications currently represents an opportunity for research.

Other features of the DTLS protocol can difficult its employment with constrained wireless sensing devices in Internet-integrated WSN. In [88] the authors also discuss the inadequateness of the timers for message retransmission defined in the current DTLS specification, which may require the usage of large buffers on the receiver to hold data for retransmission purposes, and impact on the size of the code required to support DTLS in constrained sensing platforms. This work also discusses the employment of stateless compression of the DTLS headers with the goal of reducing the overhead of DTLS records and handshake messages. Authors in [89] follow this approach and propose the compression of the DTLS headers using LOWPAN_IPHC 6LoWPAN header compression [19]. Similarly to the previous proposals on 6LoWAN network-layer security, DTLS header compression would require appropriate support from existing DTLS implementations for constrained sensing devices, or on the other end the usage of mechanisms to map between TLS/DTLS and compressed DTLS at a security gateway.

Another approach to support DTLS security is to offload costly operations to more capable devices such as a security gateway. A few proposals consider this approach, focusing particularly on the delegation of operations related with the DTLS handshake to a gateway such as the one considered in the architecture illustrated in Fig. 2. Authors in [87] propose a mechanism for mapping between TLS and DTLS at the gateway, while also supporting mappings between CoAP and HTTP. In [90] 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 [91,92] 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. Authors in [93] propose the employment of a security gateway 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 the security gateway. In this proposal the gateway is in possession of the keying material it may use to decrypt communications between the two CoAP parties, after the initial DTLS handshake. This material may subsequently support additional security mechanisms involving the analysis of encrypted traffic, for example intrusion detection or detection of attacks at the application-layer. Regarding the inadequacy of DTLS to multicast communications, an I-D [94] proposes the usage of a security gateway supporting a controller responsible for the set up of a multicast group. The controller establishes an initial DTLS handshake with each device in the group and subsequently sends to each device the keying material required to support secure CoAP group communications.

The impact of the processing of security certificates using current low-power WSN sensing platforms is also being investigated. Authors in [95] discuss possible design approaches to address the computational burden of supporting certificates in such platforms, also considering the employment of a security intermediary. The proposed approaches in this work are certificate pre-validation and session resumption. The former involves a security gateway supporting the validation of certificates in the context of the handshake, before the handshake messages are forwarded to the final sensing device, while the later allows the communication peers to maintain minimal session state after session teardown, which may be used later to resume secure communications without the need of performing again the DTLS handshake. For very constrained sensing platforms, this proposal discusses the full delegation of the DTLS handshake to a proxy, using a mechanism based on TLS session resumption without server-side state.

Finally, we may observe that regarding alternative transport-layer communication protocols for low-power WSN environments, the usage of TCP is currently an open topic of research. Existing research proposals such as [96,97] target the employment of TCP on WSN environments, although not considering 6LoWPAN-based communication technologies. We may also observe that if TCP is ever adopted for Internet-integrated WSN environments, SSL is a natural candidate to support end-to-end transport-layer security. Two previous research works addressed the employment of SSL in constrained sensing environments. SSNAIL [98] proposed a light-weighted version of SSL to be supported by Internet hosts and WSN sensing devices, while Sizzle [99] proposes the employment of a security gateway supporting partial SSL end-to-end security between an Internet host and the gateway, with communications with the final WSN device being supported by a proprietary WSN protocol. Such proposals may offer guidance in the design of alternative approaches to support SSL/TLS for end-to-end communications in the context of Internet-integrated low-power WSN.

4.4.4. Security for low-energy routing protocols

The Routing Over Low-power and Lossy Networks (ROLL) [100] working group of the IETF was formed with the purpose of designing routing solutions for LoWPAN environments. The current solution is the Routing Protocol for Low power and Lossy Networks (RPL) [20] Protocol. Rather than providing a generic solution to support routing, RPL provides a framework that is adapted to the requirements of applications in particular domains. Requirements for routing have been defined for urban applications in RFC 5548 [101], for industrial applications in RFC 5673 [102], for home automation applications in RFC 5826 [103] and for building automation applications in RFC 5867 [104]. RPL metrics appropriate to 6LoWPAN environments are also defined in RFC 6551 [105]. The specification of routing metrics and requirements as appropriate to each application area is due to the fact that appropriate routing strategies are in fact very challenging and application-specific. Each particular RFC document thus documents an objective function mapping the optimization requirements of the target application area.

Regarding security in the context of RPL, the current specification [20] defines secure versions of the various routing control messages employed by the protocol, together with three basic security modes. The secure versions of RPL routing control messages support confidentiality, integrity, delay and replay protection for the routing messages exchanged in the context of the various routing operations. Similarly to other 6LoWPAN-based communication technologies, RPL adopts AES/CCM as the cryptographic basis to support security, thus facilitating the support of RPL security on IEEE 802.15.4 sensing platforms.

The first RPL security mode is the *unsecured* mode, in which routing control messages are sent without any security guarantees, while the two security modes providing effective security are the *preinstalled* and the *authenticated* security modes. In the *preinstalled* security mode a device uses a preconfigured symmetric key to join an existing RPL instance, either as a host or as a router. In the *authenticated* security mode, a device initially joins the network using a symmetric key and next obtains a different key from a key authority, with which it may start operating as an RPL router. The key authority is thus responsible for authenticating and authorizing the device for this purpose, although the current RPL specification [20] does not address how it may be implemented. The specification also states that the authentication of an RPL device in the *authenticated* mode may not be supported by symmetric cryptography, and does not defines any requirements on how a node should alternatively employ public key cryptography for that purpose. Such aspects may of course be clarified in future versions of the standard, or on the other end in the context of future key management infrastructures adopted for Internet-integrated WSN. Other than the previously discussed security modes and the secure versions of the various routing control messages, no other mechanisms for security are defined in the current specifications. The various documents specifying routing requirements and metrics for particular application areas [101–104] only discuss generic security aspects and the necessity of handling routing information and routing control messages securely.

The RPL protocol currently lacks mechanisms to support other important operations as the secure bootstrapping of devices, key management and management of routing security policies. RPL currently only considers the usage of devices that are pre-configured with the required symmetric key to support the *preinstalled* security mode, or on the other hand that are able to learn the key from a received DIS (DODAG Information Solicitation) configuration message. DIS messages are employed by RPL to establish upward routes in the RPL routing tree. The DODAG is the Destination Oriented Directed Acyclic Graph built by RPL, which is identified by a DODAGID for each root device in the routing tree. Therefore, other authentication and secure joining mechanisms will be required in the future to support 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 for particular applications.

A previous I-D discussing the current open issues in respect to RPL security is [106], and more recently in [107], which performs an analysis on the main threats against ROLL routing mechanisms, together with recommendations on how to address security. This threats identified in this work are described according to the ISO 7498-2 security reference model [130]. This model enables the consideration of the authentication, access control, data integrity and non-repudiation, data confidentiality and availability security requirements. Security for RPL environments may also be addressed in the context of a security framework for ROLL routing protocols, as proposed in [108], which is built upon previous work on security for routing protocols adapted to the constraints of 6LoWPAN WSN communication environments. This framework enables the identification of security measures that can be activated in the context of RPL communications, and also of system security aspects that may impact on the normal functioning of routing, and which may require considerations beyond the routing protocol, as well as potential approaches in addressing them. We also note 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.

Other important aspect of RPL security, as currently proposed [20], is that the services defined in the current specification 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 [109] discuss the issue of internal attacks on RPL, particularly on the rank concept as employed by the protocol. The rank concept is used for 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. The authors also discuss that this limitation in RPL is due to the fact that a child node receives parent information through control messages, while being unable to check the services provided by the parent, so it will follow a bad quality route if it has a malicious parent. 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, despite not proposing any specific solution.

Internal attacks against RPL are also discussed in [110], 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 (DODAG Information Object) 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. The proposed mechanisms are evaluated against its impact on computational time, while the authors do not consider its impact on aspects such as the energy and memory of constrained sensing devices [110]. Those mechanisms have also been recently proposed in the form of a recent I-D [111].

In another contribution focusing on internal attacks against RPL [112], the authors discuss the effects of sinkhole attacks, particularly regarding the end-to-end data delivery performance of the network in the presence of attacks. A sinkhole attack consists of a compromised node that purposely captures and drops messages. In this work 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. The rank-verification technique in these proposals is also based on one-way hash chains as in [110,111], while the parent fail-over scheme employs an end-to-end acknowledgment scheme controlled by the DODAG root node.

In conclusion, the previously discussed open research issues represent opportunities to design and adopt new security mechanisms as part of the RPL standard in the future. As extensive research has been performed in the area of security for routing protocols in sensor and ad hoc networks in the past, such approaches may also guide future research efforts on RPL security, as long as the solutions adopted are able to cope with the constrains and characteristics of 6LoWPAN environments. Research may also focus on the design of new security mechanisms for RPL that are able to integrate with existing Internet secure routing solutions, thus more effectively supporting secure routing in the context of Internet-integrated low-power WSN.

4.4.5. Application-layer low-energy communication and security mechanisms

The efforts toward the design of application-layer standard communication mechanisms for constrained sensing environments are very recent. The Constrained RESTful Environments (CoRE) [113] working group of the IETF is currently designing the Constrained Application Protocol (CoAP) [21] to support Representational State Transfer (RESTful) web services and communications on low-power WSN 6LoWPAN environments. The main goal of CoAP is to extend the currently predominant REST web service design model to LoWPAN environments, and therefore materializes an evolution of the previously analyzed research proposals for the integration of WSN with the Internet via modified or adapted WS services.

As with communications at other layers, the gateway illustrated in the context of the integration architecture illustrated in Fig. 2 can support end-to-end CoAP communication between WSN sensing devices and other external/Internet hosts. Side-by-side with CoAP end-to-end communications, the proxy may also support mechanisms to map between HTTP (on the Internet domain) and CoAP (on the low-power WSN). As previously discussed, payload space is a scarce resource in 6LoWPAN environments, and applications are required to use it frugally. Adding to the space required for IEEE 802.15.4 and 6LoWPAN/UDP addressing, CoAP requires a 4 bytes header, as illustrated in Fig. 3.

The CoAP protocol implements a set of techniques to compress application-layer protocol metadata without compromising application inter-operability, in conformance with the representational state transfer architecture of the web. The 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. CoAP messages are exchange asynchronously between two endpoints and are used to transport CoAP requests and responses. Since such messages are transported over unreliable UDP communications, CoAP implements a lightweight reliability mechanism. The CoAP messages may be marked as *Confirmable*, for which the sender activates a simple stop-and-wait retransmission mechanism with exponential back off. 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. The *Acknowledge* or *Reset* message is related to a *Confirmable* message by means of a Message ID, along with additional information on 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.

Regarding security, as previously discussed the protocol adopts transport-layer security using DTLS, rather than adopting security at the application-layer protocol itself. Thus, our previous discussion on the limitations and open research issues in the context of DTLS are also important in the context of CoAP security. It also important to analyze other approaches to CoAP security, namely by addressing security at the application-layer rather than at the transport-layer. The CoAP protocol [21] currently defines three security modes: *PreSharedKey*, *RawPublicKey* and *Certificates* modes. All such security modes rely on DTLS to provide confidentiality, data authenticity, data integrity and non-repudiation to CoAP messages, while differing in how authentication and key negotiation is handled, as we proceed to discuss.

In the *PreSharedKey* security mode devices 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 which are unable to support public-key cryptography, or for which it is convenient to pre-configure security for the devices. Applications may use one key per destination device or a single key for a group of destination devices. In the *RawPublicKey* security mode, devices use authentication based on public keys, but without being able to be part of public key infrastructures. Devices are preprogrammed (for example as part of the manufacturing process) with an asymmetric key pair that can be validated using an out-of-band mechanism such as [114], while without using certificates. The identity of the device is obtained from its public key and the device also possesses a list of identities and public keys of the nodes it can communicate with. This security mode is currently defined as mandatory to implement.

In the *Certificates* security mode authentication is also based on public keys, but for devices that are able to participate in a certification chain for certificate validation purposes. A security infrastructure must thus be available to support this security mode. The devices use an asymmetric key pair stored in a X.509 certificate, which 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 purposes. Regarding its employment of cryptography, the CoAP protocol currently adopts Elliptic Curve Cryptography (ECC) [115] to support public key computations for the *RawPublicKey* and *Certificates* security modes. ECC supports device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA) and key agreement using the ECC Diffie-Hellman counterpart, the Elliptic Curve Diffie-Hellman Algorithm (ECDHE).

One immediate goal for research on CoAP security may be the experimental evaluation of the proposed CoAP security modes, as currently such proposals are under discussion. The viability of ECC cryptography on constrained sensing platforms is currently uncertain, and optimizations could be designed at the hardware of new platforms to support efficient ECC cryptography. Other problem may reside in the lack of memory on most constrained low-power WSN platforms, which may difficult the support of security plus all the other required 6LoWPAN-related software modules. Alternative approaches to ECC public key authentication may also be investigated, for example involving device pre-configuration and pre-shared keys, as TLS supports with TLS-PSK.

Alternative research approaches can be investigated to support CoAP security. One would be to use the application-layer protocol to support costly DTLS handshake operations. In [116] the authors proposes the usage of a RESTful DTLS handshake, to deal with the problem of the impact of large handshake messages on 6LoWPAN fragmentation. The proposed mechanism enables the efficient transmission of DTLS handshake messages in the payload of CoAP messages, using CoAP block-wise transfers [117] when required for larger messages. A DTLS session is thus 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.

Another approach consists in designing security into to CoAP protocol itself. This approach was first discussed in an I-D [118], which proposes new CoAP security options for the setup of security contexts between CoAP communicating entities, and for the identification of CoAP messages which have security applied. Authors in [119] also propose the employment of CoAP security options designed to support granular security. In this proposal, one security option allows the identification of how security is applied to a given CoAP message, other identifies the entity responsible for the processing of security for the message and authenticates the client, and another transports the data required to process security for the message. This proposal supports granular security and the usage of CoAP across different security domains. The I-D in [120] also proposes the addition of two new CoAP options related to security, the *Profile* and *Sec-flag* options. Contrary to [118,119], these options complement DTLS security rather than providing an alternative. The *Profile* option enables the attribution of a CoAP message to a particular application and the processing of security at an intermediary accordingly, while the *Sec-flag* option enables the usage of IEEE 802.15.4 link-layer security rather than DTLS in a particular segment of the communications path. This document also proposes an initial authentication and security negotiation scheme using CoAP messages transporting the *Sec-flag* option. From the previous proposals we are able to verify that application-layer security may provide various benefits when contrasted against DTLS transport-layer security to protect CoAP communications. The support of granular security implies that applications may opt to protect only particular application-layer messages or messages with particular contents, rather than being forced to process security for all messages using DTLS. Also, application-layer security may support end-to-end security in the presence of multiple WSN deployments and interconnecting gateways, as well as CoAP proxies in reverse or forward modes [21] supported by such gateways.

4.4.6. Cross-layer security aspects

An essential cross-layer security aspect in the context of the reference integration architecture illustrated in Fig. 2 is *key management*, and one that will play a fundamental role in the support of security mechanisms for Internet-integrated low-power WSN. Key management is in fact a cross-layer security issue and one that is interrelated with authentication, since security mechanisms designed to protect communications require that keys are negotiated in the context of the initial authentication of the communicating devices 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 [67] identifies the possibility of adopting simplified versions of current Internet key management solutions, such as the minimal IKEv2 proposed in [121]. RFC 6568 describes the requirements for minimal implementations of IKEv2, together with possible optimizations promoting its adaptability to constrained WSN environments. Other approaches may be pursuit to adapt IKEv2 to Internet-integrated low-power WSN environments. One is to compress the IKE headers and related payload data using 6LoWPAN IPHC compression, as proposed in [122]. The other is to adopt new lightweight key management mechanisms that are more close to the capabilities of WSN environments and to the characteristics of IoT applications [123,134]. In this work the authors also discuss that public-key approaches require nodes with less resource constraints than current reference wireless sensing platforms, and propose the adaptation of mathematical-based key management approaches.

The gateway illustrated in Fig. 2 can also support standard Internet key negotiation mechanisms with Internet hosts, while abstracting such key negotiation operations from the constraints and characteristics of WSN devices. The gateway may deal with the identification and authorization of sensing devices prior to key management, therefore acting as a trusted broker for end-to-end key negotiation purposes. Alternatively, key negotiation may be performed in a truly end-to-end fashion, as such having key management mechanisms dealing with the constraints and characteristics of sensing devices and applications. The applicability of existing key management mechanisms designed to support link-layer security on sensor networks [124] to Internet-integrated WSN can also be investigated. Proposals based on mathematical techniques such as linear algebra, combinatory or algebraic geometry may be of interest as discussed in [124], as they may contribute or at least provide the ground for the adoption of new key-management mechanisms. Research work may also target the extension of such proposals to global environments in the context of its integration with the IKE standard, or its adaptation to the usage of a trusted third-party, as this would provide support for the usage a of security infrastructure supporting authentication and key negotiation for Internet-integrated WSN.

Other important security services in the context of the integration architecture previously discussed and illustrated in Fig. 2 are those to guarantee *authentication*, *authorization* and *control of accesses*. Such services will be fundamental, as not all services provided by applications employing Internet-integrated WSN will be public, and some applications may require that accesses to data available on sensing devices be carefully controlled. Access control mechanisms may be designed to operate on packet header information related with 6LoWPAN-based communication protocols, as this would enable a fine-grained control of communications between the Internet and WSN domains. In the same context, compressed 6LoWPAN security headers, DTLS headers and CoAP security options can be inspected and processed in cooperation with security-mapping and key management mechanisms.

The creation of a worldwide object network will require a security infrastructure to support mutual object authentication and operations related with identity management,

Table 3
Proposals on security for 6LoWPAN-based communication technologies.

Research proposal	Operational layer	Security properties and functionalities supported	Application context of security	Implementation details
[51,72,76]	6LoWPAN adaptation layer	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
[77,78]	6LoWPAN adaptation layer	Confidentiality, integrity, authentication, non-repudiation	Transparent end-to-end (network layer) security	IPHC (6LoWPAN) compression of AH and ESP security headers; preprogrammed 128-bit keys
[80]	6LoWPAN adaptation layer	Resistance against 6LoWPAN fragmentation attacks	Communications between 6LoWPAN devices with fragmentation	Addition of a timestamp plus a nonce to the 6LoWPAN fragmentation header to support security against unidirectional and bidirectional fragment replays
[81]	6LoWPAN adaptation layer	Resistance against 6LoWPAN fragmentation attacks	6LoWPAN communications between sensing devices or end-to-end communications with external devices	Mechanisms to support per-fragment sender authentication using hash chains and purging of messages from suspicious senders based on the observed behavior
[89]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation repudiation using DTLS. Reduction of the overhead of the DTLS records	Transparent end-to-end (transport layer) security	IPHC compression of the DTLS headers in the context of the 6LoWPAN adaptation layer
[91,92]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation repudiation using DTLS. Reduction of the overhead of DTLS with hardware assistance	Transparent end-to-end (transport layer) security	End-to-end DTLS using mutual authentication with hardware support provided by specialized trusted-platform modules (TPM) with RSA cryptography
[93]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation repudiation using DTLS. Reduction of the overhead of DTLS by offloading operations to a powerful device	Transparent end-to-end (transport layer) security	Transparent interception and mediation of the DTLS handshake, offloading of ECC public key computations to the gateway
[95]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation repudiation using DTLS. Reduction of the overhead of DTLS by offloading operations to a powerful device	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
[94]	Transport-layer	Confidentiality, integrity, authentication, non-repudiation repudiation using DTLS for multicast communications	Support secure multicast communications on sensing devices	Setup of multicast groups by the gateway, each sensing device performs the initial DTLS handshake with the gateway and receives the required keying material
[117]	Transport layer	Support of DTLS handshake with block-wise communications	Support authentication and initial key agreement with sensing devices employing DTLS	DTLS handshake messages transported in the payload of CoAP application-layer messages using CoAP block-wise transfers to reduce 6LoWPAN fragmentation
[110,111]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Version number and rank authentication security scheme based on one-way hash chains providing security against internal attackers
[112]	Routing layer	Resistance against internal attacks	Protection of RPL routing operations against falsified routing updates	Security mechanism combining parent fail-over with a rank authentication scheme to combat sinkhole attacks
[118]	Application layer	Confidentiality, integrity, authentication, non-repudiation repudiation for CoAP web messages	Transparent end-to-end (application layer) security	CoAP security options allow for the setup of security contexts between CoAP communicating entities and protection of CoAP messages
[119]	Application layer	Confidentiality, integrity, authentication, non-repudiation repudiation for CoAP web messages	Transparent and granular end-to-end (application layer) security	CoAP security options for granular security, authentication of clients and secure transversal of multiple security domains
[120]	Application layer	Confidentiality, integrity, authentication, non-repudiation repudiation for CoAP web messages	Application layer security with application identification and support for link-layer security	CoAP security options to complement DTLS security, identification of particular applications and employment of link-layer security when appropriate

anonymization, authentication and authorization. While not all IoT applications will require or be able to access such an infrastructure, research and standardization work will be required for its design and integration with current certification infrastructures. Authentication and authorization mechanisms will also be dependent on the adoption of suitable and scalable identification mechanisms to provide unique identifiers and virtual identifiers to users, sensors and other types of devices [125,126].

As with any other Internet device, it is fair to expect that a low-power WSN sensing device exposed to Internet communications will be targeted by malicious entities trying to hinder the availability of its services. In this context, *fault tolerance* in Internet-integrated WSN devices may involve making all objects secure by default, giving all objects to know the state of the network and its services, and making objects able to defend themselves against network failures and attacks. Despite such desirable properties, the employment of a security gateway as in the reference integration architecture illustrated in Fig. 2 will be important and in many situation unavoidable, and the gateway may provide valuable support in the enforcement of appropriate security perimeters.

Other fundamental aspect related with fault-prevention is intrusion detection. Despite the existence of preliminary works on intrusion detection systems for 6LoWPAN WSN environments [127,135,136], further research still needs to be performed in this area. Intrusion detection mechanisms can be extended to understand possible attacks against 6LoWPAN-based communication and security mechanisms, and be developed symbiotically with other mechanisms required to guaranteeing the availability and robustness of the WSN, such as load balancing. In conclusion, In Table 3 at the end of the survey we resume the main characteristics of the previously analyzed research proposals targeting security aspects in 6LoWPAN-based communication protocols and environments.

5. Conclusions

The integration of low-power WSN with the Internet may enable future sensing applications in the context of which sensing and actuating devices interface with the physical world and are able to communicate with remote devices over the Internet. This aspect is also currently motivating the design and adoption of new communication technologies appropriate to support Internet communications on low-power wireless sensing devices, as analyzed throughout the article. Technologies such as 6LoWPAN, RPL and CoAP belong in this context, and already enable end-to-end communications between constrained wireless sensing devices and external or Internet entities. Along with such communication technologies, security mechanisms are required to protect end-to-end communications, and also to address security aspects that may be considered to be cross-layer. Appropriate security solutions will thus be required to support the integration of WSN with the Internet at an architectural level.

As we have discussed throughout the survey, security will be a fundamental enabling factor of the integration of WSN applications with the Internet, irrespective of the integration approach considered. With this in mind, we perform an exhaustive analysis of the communication and security technologies already available or currently being designed with this goal. We analyze existing research proposals and open issues that constitute research opportunities in the area. 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, and helping readers interested in developing new solutions to address security in the context of the integration of low-power WSN with the Internet.

References

[1] Yun Zhou, Yuguang Fang, Yanchao Zhang, Securing wireless sensor networks: a survey, IEEE Commun. Surv. Tutorials 10 (3) (2008) 6–28 (Third Quarter) http://dx.doi.org/10.1109/COMST.2008.4625802.

[2] Chen Xiangqian, K. Makki, Yen Kang, N. Pissinou, Sensor network security: a survey, IEEE Commun. Surv. Tutorials 11 (2) (2009) 52–73 (Second Quarter).

[3] GainSpan Unveils Industry First Wi-Fi® and ZigBee® IP Single Chip. <http://www.gainspan.com/news/news_20130226_gs2000> (accessed May 2014).

[4] IEEE 802.3 ETHERNET WORKING GROUP, <http://www.ieee802.org/3/> (accessed May 2014).

[5] ITU – SDH & SONET Related Recommendations and Standards. <http://www.itu.int/ITU-T/2001-2004/com15/otn/SDH-rec.html> (accessed May 2014).

[6] Guifen Gu, Guili Peng, The survey of GSM wireless communication system, in: 2010 International Conference on Computer and Information Application (ICCIA), IEEE, 2010.

[7] ETSI, The European Telecommunications Standards Institute. <http://www.etsi.org/> (accessed May 2014).

[8] S. Chia, The Universal Mobile Telecommunication System, Communications Magazine, IEEE 30 (12) (1992) 54–62. http://dx.doi.org/10.1109/35.210356.

[9] The 3rd Generation Partnership Project (3GPP). <http://www.3gpp.org/> (accessed May 2014).

[10] LTE-Advanced. <http://www.3gpp.org/lte-advanced> (accessed May 2014).

[11] IEEE 802.11™: Wireless LANs. <http://standards.ieee.org/about/get/802/802.11.html> (accessed May 2014).

[12] IEEE 802.16™: BROADBAND WIRELESS METROPOLITAN AREA NETWORKS (MANs). <http://standards.ieee.org/about/get/802/802.16.html> (accessed May 2014).

[13] Stefan Aust, R. Venkatesha Prasad, Ignas G.M.M. Niemegeers, IEEE 802.11 ah: advantages in standards and further challenges for sub 1 GHz Wi-Fi, in: 2012 IEEE International Conference on Communications (ICC), IEEE, 2012.

[14] IEEE 802.15™: WIRELESS PERSONAL AREA NETWORKS (PANs). <http://standards.ieee.org/about/get/802/802.15.html> (accessed May 2014).

[15] Kyung Sup Kwak, Sana Ullah, Niamat Ullah, An overview of IEEE 802.15.6 standard, in: 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), IEEE, 2010.

[16] 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), September 5, 2011, pp. 1, 314, http://dx.doi.org/10.1109/IEEESTD.2011.6012487.

[17] N. Kushalnagar et al., IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, RFC 4919, 2007.

[18] Montenegro G. et al., Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944, 2007.

[19] J. Hui, P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-based Networks, RFC 6282, 2011.

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

[21] Bormann Carsten, Angelo P. Castellani, Zach Shelby, CoAP: an application protocol for billions of tiny internet nodes, IEEE Internet Comput. 16.2 (2012).

[22] R. Roman, P. Najera, J. Lopez, Securing the internet of things, IEEE Comput. 44 (9) (2011) 51–58.
[23] Rodrigo Roman, Javier Lopez, Integrating wireless sensor networks and the internet: a security analysis, Internet Res. 19 (2) (2009) 246–259.
[24] Luigi Atzori, Antonio Iera, Giacomo Morabito, The internet of things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805.
[25] M. Anwar Hossain, A survey on sensor-cloud: architecture, applications, and approaches, Int. J. Distrib. Sens. Netw. 2013 (2013).
[26] Xively by LogMeIn. <https://xively.com/> (accessed May 2014).
[27] SensorCloud powered by LORD MicroStrain. <http://www.sensorcloud.com/> (accessed May 2014).
[28] SensaTrack. <http://www.sensatrack.com/> (accessed May 2014).
[29] NimBits – The Open Source Internet of Things on a Distributed Cloud. <http://www.nimbits.com/> (accessed May 2014).
[30] ThingSpeak. <https://www.thingspeak.com/> (accessed May 2014).
[31] Dominique Guinard et al., Towards physical mashups in the web of things, in: 2009 Sixth International Conference on Networked Sensing Systems (INSS), IEEE, 2009.
[32] Dominique Guinard, Vlad Trifa, Towards the web of things: Web mashups for embedded devices". Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in: proceedings of WWW (International World Wide Web Conferences), Madrid, Spain, 2009.
[33] Vlad Trifa et al., Design and implementation of a gateway for web-based interaction and management of embedded devices, Submitted to DCOSS, 2009.
[34] W. Grosky, A. Kansal, S. Nath, Liu Jie, Zhao Feng, SenseWeb: an infrastructure for shared sensing, IEEE Multimedia 14 (4) (2007) 8–13.
[35] Suman Nath, Jie Liu, Feng Zhao, Sensormap for wide-area sensor webs, Computer 40 (7) (2007) 0090–0093.
[36] Guinard Dominique, Vlad Trifa, Erik Wilde, A resource oriented architecture for the web of things, Internet of Things (IOT), IEEE (2010).
[37] Qian Zhu et al., Iot gateway: bridging wireless sensor networks into internet of things, in: 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC), IEEE, 2010.
[38] SENSEI (Integrating the Physical with the Digital World of the Network of the Future). <http://www.sensei-project.eu/> (accessed May 2014).
[39] Emiliano De Cristofaro, Jens-Matthias Bohli, Dirk Westhoff, FAIR: fuzzy-based aggregation providing in-network resilience for real-time wireless sensor networks, in: Proceedings of the second ACM Conference on Wireless Network Security, ACM, 2009.
[40] Santander on FIRE – Future Internet Research and Experimentation. <http://www.smartsantander.eu/> (accessed May 2014).
[41] WISEBED. <http://wisebed.eu/site/> (accessed May 2014).
[42] Internet of Things – Architecture. <http://www.iot-a.eu/public> (accessed May 2014).
[43] ETSI Machine to Machine Communications. <http://www.etsi.org/technologies-clusters/technologies/m2m> (accessed May 2014).
[44] ITU Telecommunication Standardization Sector. <http://www.itu.int/en/ITU-T/Pages/default.aspx> (accessed May 2014).
[45] ZigBee, Alliance, ZigBee Specification. ZigBee Document 053474r13, 2006, pp. 344–346.
[46] ZigBee IP Specification Overview. <http://www.zigbee.org/Specifications/ZigBeeIP/Overview.aspx> (accessed May 2014).
[47] Sensinode. <http://www.sensinode.com/> (accessed May 2014).
[48] ARM – The Architecture for the Digital World. <http://www.arm.com/> (accessed May 2014).
[49] J. Hui et al., Trio: enabling sustainable and scalable outdoor wireless sensor network deployments, in: Proceedings of the Information Processing in Sensor Networks (2006) 407–415.
[50] W. Youssef, M. Younis, Intelligent gateways placement for reduced data latency in wireless sensor networks, in: Proceedings of the IEEE Conference on Communications ICC '07, June 2007, pp. 3805–3810.
[51] J. Granjal, E. Monteiro, J. Sá Silva, A secure interconnection model for IPv6 enabled wireless sensor networks, in: Proceedings of the IFIP Wireless Days 2010, October 2010, pp. 1–6.
[52] Simon Duquennoy, Gilles Grimaud, J-J. Vandewalle, The Web of Things: interconnecting devices with high usability and performance, in: ICESS'09. International Conference on Embedded Software and Systems, 2009, IEEE, 2009.
[53] Adam Dunkels, Efficient application integration in IP-based sensor networks, in: Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, ACM, 2009.
[54] Michael Buettner et al., X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, ACM, 2006.
[55] R. Dickerson, J. Lu, K. Whitehouse, Stream feeds: an abstraction for the World Wide Sensor Web, in: Proceedings of the Conference on the Internet of Things (IOT 2008), March 2008, pp. 360–375.
[56] Law Yee Wei et al., Secure rateless deluge: pollution-resistant reprogramming and data dissemination for wireless sensor networks, EURASIP J. Wireless Commun. Netw. (2011) 5.
[57] Nicola Bui et al., An integrated system for secure code distribution in wireless sensor networks, in: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, 2010.
[58] J-M. Bohli et al., Security enhanced multi-hop over the air reprogramming with fountain codes, in: IEEE 34th Conference on Local Computer Networks, 2009, LCN 2009, IEEE, 2009.
[59] Conti Mauro et al., Mobility and cooperation to thwart node capture attacks in manets, EURASIP J. Wireless Commun. Netw. (2009).
[60] 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), April 16, 2012, pp. 1, 225, http://dx.doi.org/10.1109/IEEESTD.2012.6185525.
[61] Joseph Polastre, Robert Szewczyk, David Culler, Telos: enabling ultra-low power wireless research, in: Information Processing in Sensor Networks, 2005. Fourth International Symposium on IPSN 2005, IEEE, 2005.
[62] IPv6 over Low power WPAN (6lowpan). <https://datatracker.ietf.org/wg/6lowpan/charter/> (accessed May 2014).
[63] J. Nieminen, B. Patil, T. Savolainen, M. Isomaki, Z. Shelby, C. Gomez, Transmission of IPv6 Packets over Bluetooth Low Energy, draftietf-6lowpan-btle, 2013.
[64] Peter Mariager, Jens Petersen, Transmission of IPv6 Packets over DECT Ultra Low Energy, draft-mariager-6lowpan-v6over-dect-ule-02, 2012.
[65] A. Brandt, J. Buron, Transmission of IPv6 Packets over ITU-T G.9959 Networks, draft-brandt-6man-lowpanz-02.txt, 2013.
[66] S. Park et al., IPv6 over Low Power WPAN Security Analysis, draft-6lowpan-security-analysis-05, 2011.
[67] Eunsook Kim, Dominik Kaspar, Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6568, April 2012.
[68] Z. Shelby et al., Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, November 2012.
[69] T. Narten, E. Nordmark, W. Simpson, H. Soliman, Neighbor Discovery for IP version 6 (IPv6), RFC 4861, September 2007.
[70] Jari Arkko et al., Secure Neighbor Discovery (SEND), RFC 3971, March 2005.
[71] T. Aura, Cryptographically Generated Addresses, RFC 3972, 2005.
[72] Jorge Granjal, Edmundo Monteiro, J. Silva, Enabling network-layer security on IPv6 wireless sensor networks, in: 2010 IEEE Global Telecommunications Conference (GLOBECOM 2010), IEEE, 2010.
[73] Stephen Kent, Randall Atkinson, IP Authentication Header, RFC 2402, 1998.
[74] S. Kent, R. Atkinson, Encapsulating Security Protocol, RFC 2406, 1998.
[75] S. Kent, K. Seo, Security Architecture for the Internet Protocol, RFC 4301, 2005.
[76] J. Granjal, E. Monteiro, J.S. Silva, Network-layer security for the Internet of Things using TinyOS and BLIP, Int. J. Commun. Syst. (2012). http://dx.doi.org/10.1002/dac.2444.
[77] S. Raza et al., Securing communication in 6LoWPAN with compressed IPsec, in: Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11), June 2011, pp. 1–8.
[78] Shahid Raza et al., Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN, Secur. Commun. Netw. (2012).
[79] Carsten Bormann, Using CoAP with IPsec, draft-bormann-core-ipsec-for-coap-00, 2012.
[80] HyunGon Kim, Protection against packet fragmentation attacks at 6lowpan adaptation layer, in: International Conference on Convergence and Hybrid Information Technology, 2008, ICHIT'08, IEEE, 2008.
[81] René Hummen, Jens Hiller, Hanno Wirtz, Martin Henze, Hossein Shafagh, Klaus Wehrle, 6LoWPAN fragmentation attacks and mitigation mechanisms, in: Proceedings of the Sixth ACM

Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '13), ACM, New York, NY, USA, pp. 55–66.
[82] J. Postel, User Datagram Protocol, RFC 768, August 1980.
[83] Transmission Control Protocol, RFC 793, September 1981.
[84] Eric Rescorla, Nagendra Modadugu, Datagram Transport Layer Security, RFC 4347, 2006.
[85] T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.1, RFC 4346, 2006.
[86] Oscar Garcia-Morchon et al., Security Considerations in the IP-based Internet of Things, draft-garcia-core-security-06, 2013.
[87] Martina Brachmann et al., End-to-end transport security in the IP-based Internet of Things, in: 2012 21st International Conference on Computer Communications and Networks (ICCCN), IEEE, 2012.
[88] Klaus Hartke, Practical Issues with Datagram Transport Layer Security in Constrained Environments, draft-hartke-dice-practical-issues-00, 2013.
[89] Shahid Raza, Daniele Trabalza, Thiemo Voigt, 6LoWPAN compressed DTLS for COAP, in: 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2012.
[90] Mohit Sethi, Jari Arkko, Ari Keranen, End-to-end security for sleepy smart object networks, in: 2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops), IEEE, 2012.
[91] Thomas Kothmayr et al., A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication, in: 2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops), IEEE, 2012.
[92] Thomas Kothmayr et al., DTLS based security and two-way authentication for the Internet of Things, Ad Hoc Netw. (2013).
[93] Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva, End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication, IFIP Netw. (2013).
[94] Sandeep Kumar, Esko Dijk, Sye Keoh, DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs), draft-keoh-tls-multicast-security-00, 2012.
[95] René Hummen, Jan H. Ziegeldorf, Hossein Shafagh, Shahid Raza, Klaus Wehrle, Towards viable certificate-based authentication for the internet of things, in: Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy (HotWiSec '13), ACM, New York, USA, 2013, pp. 37–42.
[96] Z. Zheng, A. Ayadi, J. Xiaoran, TCP over 6LoWPAN for industrial applications: an experimental study, in: Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS), February 2011, pp. 1–4.
[97] T, Braun, T. Voigt, A. Dunkels, TCP support for sensor networks", in: Proceedings of the Fourth Annual Conference on Wireless on Demand Network Systems and Services, WONS '07, January 2007, pp. 162–169.
[98] Jung et al., SSL-based Lightweight Security of IP-based Wireless Sensor Networks, in: Proceedings of the International Conference on Advanced Information Networking and Applications Workshop (WAINA '09), May 2009, pp. 1112–1117.
[99] V. Gupta et al., Sizzle: a standards-based end-to-end security architecture for the embedded Internet, in: Proceedings of the Third IEEE International Conference on Pervasive Computing for the Embedded Internet (PERCOM '05), March 2005, pp. 246–256.
[100] Routing Over Low power and Lossy networks (ROLL). <http://tools.ietf.org/wg/roll/charter/> (accessed May 2014).
[101] M. Dohler et al., Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548, May 2009.
[102] K. Pister et al., Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673, October 2009.
[103] A. Brandt, J. Buron, G. Porcu, Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826, 2010.
[104] J. Martocci et al., Building Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5867, 2010.
[105] J. Vasseur et al., Routing Metrics used for Path Calculation in Low Power and Lossy Networks, RFC 6551, 2012.
[106] Tzeta Tsao, A Security Design for RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, draft-sdt-roll-rpl-security-00, 2010.
[107] Tzeta Tsao et al., A Security Threat Analysis for Routing over Low Power and Lossy Networks, draft-ietf-roll-security-threats-06, 2013.
[108] Roger Alexander et al., A Security Framework for Routing over Low Power and Lossy Networks, draft-ietf-roll-security-framework-07, 2012.
[109] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, M. Chai, The impact of rank attack on network topology of routing protocol for low-power and lossy networks, Sens. J. IEEE 99 (2013) 1, 1, 0. http://dx.doi.org/10.1109/JSEN.2013.2266399.
[110] A. Dvir, T. Holczer, L. Buttyan, VeRA – version number and rank authentication in RPL, in: 2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS), 17–22 October 2011, pp. 709, 714. http://dx.doi.org/10.1109/MASS.2011.76.
[111] Laszlo Dora et al., Version Number and Rank Authentication, draft-dvir-roll-security-authentication-01, 2012.
[112] K. Weekly, K. Pister, Evaluating sinkhole defense techniques in RPL networks, in: 2012 20th IEEE International Conference on Network Protocols (ICNP), October 30, 2012–November 2, 2012, pp. 1, 6. http://dx.doi.org/10.1109/ICNP.2012.6459948.
[113] Constrained RESTful Environments (core). <https://datatracker.ietf.org/wg/core/charter/> (accessed September 2013).
[114] Paul Wouters et al., Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), draft-ietf-tls-oob-pubkey-11, 2013.
[115] SECG-Elliptic Curve Cryptography-SEC 1. <http://www.secg.org> (accessed May 2014).
[116] Sandeep Kumar, Zach Shelby, Sye Keoh, Profiling of DTLS for CoAP-based IoT Applications, draft-keoh-dtls-profile-iot-00, 2013.
[117] C. Bormann, Z. Shelby, CoRE Working Group, Blockwise Transfers in CoAP, draft-ietf-core-block-14, 2013.
[118] Alper Yegin, Zach Shelby, CoAP Security Options, draft-yegin-coap-security-options-00, 2011.
[119] Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva, Application-layer security for the WoT: extending CoAP to support end-to-end message security for internet-integrated sensing applications, Wired/Wireless Internet Communication, Springer, Berlin Heidelberg, 2013. pp. 140–153.
[120] Wendong Wang et al., CoAP Option Extensions: Profile and Sec-flag, draft-wang-core-profile-secflag-options-02, 2012.
[121] Tero Kivinen, Minimal IKEv2, draft-kivinen-ipsecme-ikev2-minimal-01, 2012.
[122] Shahid Raza, Thiemo Voigt, Vilhelm Jutvik, Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 Security, in: Proceedings of the IETF Workshop on Smart Object Security, 2012.
[123] Rodrigo Roman et al., Key management systems for sensor networks in the context of the Internet of Things, Comput. Electr. Eng. 37 (2) (2011) 147–159.
[124] M. Simplicio et al., A survey on key management mechanisms for distributed Wireless Sensor Networks, Comput. Netw.: Int. J. Comput. Telecommun. Netw. 45 (15) (Oct. 2010) 2591–2612.
[125] GS1 Identification Keys (ID Keys). <http://www.gs1.org/barcodes/technical/id_keys> (accessed May 2014).
[126] Learning about ucode. <http://www.uidcenter.org/learning-about-ucode> (accessed May 2014).
[127] S. Amini, Y. Yoon, M. Siddiqu, C. Hong, A novel intrusion detection framework for IP-based sensor networks, in: Proceedings of the 23rd International Conference on Information Networking (ICOIN'09), January 2009, pp. 285–287.
[128] Rabia Riaz, Ki-Hyung Kim, H. Farooq Ahmed, Security analysis survey and framework design for ip connected lowpans, in: International Symposium on Autonomous Decentralized Systems, 2009, ISADS'09, IEEE, 2009.
[129] Shelby Zach, Carsten Bormann, 6LoWPAN: The Wireless Embedded Internet, vol. 43, John Wiley & Sons, 2011.
[130] International Organization for Standardization, Information Processing Systems – Open Systems Interconnection Reference Model – Security Architecture, ISO Standard 7498-2, 1988.
[131] Gan Gang, Lu Zeyong, Jiang Jun, Internet of things security analysis, in: 2011 International Conference on Internet Technology and Applications (iTAP), IEEE, 2011.
[132] Rolf H. Weber, Internet of Things–new security and privacy challenges, Comput. Law Secur. Rev. 26 (1) (2010) 23–30.
[133] Daniele Miorandi et al., Internet of things: vision, applications and research challenges, Ad Hoc Netw. 10 (7) (2012) 1497–1516.
[134] Muhamed Turkanović, Boštjan Brumen, Marko Hölbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion, Ad Hoc Netw. (2014).
[135] Anhtuan Le et al., 6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach, Int. J. Commun Syst 25 (9) (2012) 1189–1212.
[136] Raza Shahid, Linus Wallgren, Thiem. Voigt, SVELTE: real-time intrusion detection in the Internet of Things, Ad Hoc Netw. 11 (8) (2013) 2661–2674.

Please cite this article in press as: J. Granjal et al., Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.08.001

Jorge Granjal is an Invited Assistant Professor at the Department of Informatics Engineering of the University of Coimbra, Portugal, and a Researcher of the Laboratory of Communication and Telematics of the Centre for Informatics and Systems of the University of Coimbra, Portugal. His main research interests are Computer Networks, Network Security and Wireless Sensor Networks. He is a member of IEEE and ACM communications group. He is currently pursuing a PhD in the area of security in Wireless Sensor Networks and the Internet of Things.

Edmundo Monteiro is Full Professor at the University of Coimbra, Portugal, form where he got his PhD in Electrical Engineering, Informatics Specialty. His research interests are Computer Networks, Wireless Communications, Service Oriented Infrastructures and Security. He is author of several publications including books, patents, and over 200 papers in national and international refereed books, journals and conferences. Edmundo Monteiro is the Portuguese representative in IFIP-TC6, he is member of IEEE Communications, IEEE Computer, and ACM Communications groups.

Jorge Sá Silva received his PhD in Informatics Engineering in 2001 from the University of Coimbra, where is an Assistant Professor at the Department of Informatics Engineering of the University of Coimbra and a Senior Researcher of Laboratory of Communication and Telematics, Portugal. His main research interests are Mobility, Network Protocols and Wireless Sensor Networks. He has been serving as a reviewer and publishing in top conferences and journals in his expertise areas. His publications include 2 book chapters and over 70 papers in refereed national and international conferences and magazines. He is a member of IEEE, and he is a licensed Professional Engineer. His homepage is at http://www.dei.uc.pt/~sasilva.

Please cite this article in press as: J. Granjal et al., Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Netw. (2014), http://dx.doi.org/10.1016/j.adhoc.2014.08.001