Preserving Privacy in secured ZigBee Wireless Sensor Networks

Jessye dos Santos, Christine Hennebert, Cédric Lauradoux

Preserving Privacy in secured ZigBee Wireless Sensor Networks

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

Cédric Lauradoux
INRIA Rhône-Alpes
655, avenue de l'Europe
38330 Montbonnot Saint-Martin, FRANCE
cedric.lauradoux@inria.fr

***Abstract*—We expose concretely the information leakage occurring in an IEEE 802.15.4-based ZigBee meshed network. We deploy an IoT platform and used a `killerbee` sniffer to eavesdrop the communication between the motes. Metadata and control traffic are exploited in depth to recover protocol instances, routes, identity, capability and activity of the devices. We experiment different levels of security for the communications from none to the best available. Even when security is enforced, information leakages are not avoided. We propose simple countermeasures to prevent an outsider from monitoring a ZigBee network.**

***Keywords*-ZigBee WSN; security; privacy; attacks; eavesdropping; IDS**

I. Introduction

In 2020, it has been projected that about 26 billion devices will be connected to the Internet through different communication standards. Many of these objects are expected to use low-power communication and to be resource and energy constrained. ZigBee standard [1], based on IEEE 802.15.4, enables these objects to be organized into meshed Wireless Sensor Network (WSN) and to distribute their resources on the Internet via a gateway.

Among many technologies available in the domain of connected objects, ZigBee is very popular. It is promoted on the SmartHome market, in particular SmartLighting, providing consumers with Plug & Play solutions and do-it-yourself concept. The democratization of this technology allows increasing comfort, efficiency, reducing energy consumption but it is not without raising different issues concerning privacy.

The densification and the coexistence within a single area of several private and independent WSNs increase the security and privacy risks. Indeed, ZigBee communications can be observed in close proximity as for WiFi. It becomes therefore necessary to protect the network from outsiders and to prevent any device from connecting without authorization. The confidentiality of the data exchanged over the wireless link is a first challenge solved by encryption. But encryption does not prevent an adversary from recovering many sensitive information about the network users: their lifestyles, their presence or their activity. Protecting consumer privacy is now a necessity to avoid massive collection of metadata from third parties.

This paper exposes the information leakage occurring in ZigBee network. We use `killerbee` [2] platform that for $40 enables data collection and sensitive information recovery such as the network topology and the motes activity. We describe simple and effective solutions to prevent these massive information leaks.

Our paper is organized as follows. ZigBee WSNs and their security are overviewed in the Section II. In Section III, our WSN platform and our experiments are described. The information we gather using our sniffer are described when the network is operated without any security (Section IV). In Section V, we consider the case of secured ZigBee WSN. The Section VI proposes privacy enhancing countermeasures. Then, we conclude.

II. Related Work

Two complementary mechanisms could contribute to the security and the privacy protection of a WSN: prevention and detection of attacks. Prevention techniques are based on the use of cryptography and protocols to ensure appropriate confidentiality, authentication, integrity, freshness or non-repudiation of data exchanged. Intrusion Detection Systems (IDS) may be implemented to alert in case of internal or external attacks.

A. Preliminary: ZigBee communication layers

The IEEE 802.15.4 standard was introduced to enable resource constrained objects to communicate over a wireless channel. ZigBee is based on the physical and MAC layers of IEEE 802.15.4. It allows the deployment of routed WSN according to a topology in star, mesh or cluster tree.

The ZigBee stack consists of several layers including the PHY, MAC, Network, Application Sublayer (APS), and ZigBee Device Objects (ZDO) layers (Fig. 1). Technically, an Application Framework (AF) layer also exists, but will be grouped with the APS layer in the remaining discussions.

[1] ZigBee Alliance, http://zigbee.org/

[2] https://code.google.com/p/killerbee/

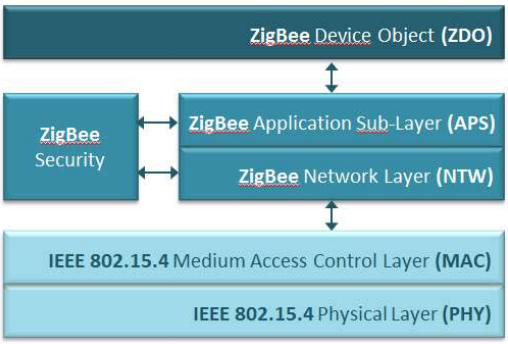

Figure 1: ZigBee communication stack

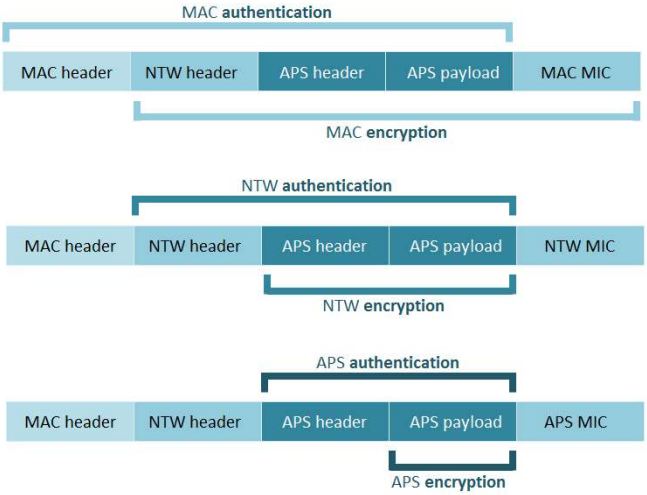

Figure 2: Encryption & Authentication at all layers

The PHY layer supports the receiving sensitivity, the wave modulation, the channel management and the transmission rate. ZigBee generally operates on the 2.4 GHz ISM band at 250 kbps data rate. The MAC layer manages data exchange between neighbor motes (point-to-point). It includes services as reply, acknowledgment and collision avoidance (CSMA-CA) techniques. The NTW layer adds routing capabilities to allow multi-hops data exchanges between two devices (peer-to-peer). The APS layer offers various addressing supports to define profiles, clusters or endpoints. The ZDO layer provides service discovery and advanced management services for the ZigBee objects.

ZigBee handles five kinds of devices: the Coordinator (ZC), the Router (ZR), the End Device (ZED), the Gateway (ZG) and the Trust Center (ZTC). There is always at least one ZC in a ZigBee WSN that selects a PAN ID and a channel to start the network. Both ZC and ZR can allow other devices to join the network and route data. After a ZED joins a ZR or ZC, it can transmit or receive RF data through that device. The ZC and ZR cannot sleep as they are able to buffer the incoming packets destined to a potentially sleeping ZED. The ZG ensures the interoperability between the ZigBee WSN and other communication protocols. The ZTC is responsible for authenticating devices that join the network. The ZTC also manages Link key distribution in the network. A ZC can be defined as a trust center, and thus is alerted to all new join attempts in the network.

B. Prevention in ZigBee WSN

As for 6LoWPAN technology [1], the security is present in each layer of the ZigBee communication stack (Fig. 2). The IEEE 802.15.4 standard provides 8 security modes at the MAC layer for encryption with 128-bits AES, authentication and control of data integrity with a Message Integrity Code (MIC). These security functions are generally performed in hardware into the radio front-end. At the MAC layer, both MAC payload and MIC are encrypted. However, the key management and the authentication policies remain the issue of the upper layers.

ZigBee supports various features of security: software implementation of 128-bit AES encryption, two security keys that can be preconfigured or obtained during joining, support for a trust center.

ZigBee introduces three kinds of keys in [2]:

- the Master key is often factory installed and used to generate the Link keys. This key is only available with ZigBee PRO version.
- the Link key secures peer-to-peer exchanges at the application layer.
- the Network key is shared by sub-network devices at the network layer.

The network key is used to encrypt the APS layer. Network security is also applied to *Route Request* and *Route Reply* messages, APS commands, and ZDO commands. However, network encryption is not applied to some MAC frames such as Beacon. If security is enabled in a network, all data packets will be encrypted with the network key. Packets with NTW layer encryption are encrypted and decrypted at each hop in a route, implying latency. APS security provides end-to-end security using an APS link key known only by the source and the destination devices.

The coordinator is responsible for selecting a network encryption key. This key can either be preconfigured or randomly selected. In addition, the coordinator generally operates as a trust center and must therefore select the trust center link key. Devices that join the network must obtain the network key. If the joining device has a pre-configured trust center link key, the network key will be sent encrypted with the link key. Otherwise, the network key is sent unencrypted. To maximize security, devices should be pre-configured with the correct link key.

Table I: Known attacks on WSN and their countermeasures

Layers	Attacks	Prevention	Detection
Physical	Jamming (DoS)	Channel Hopping, Blacklisting & CSMA-CA	Fuzzy Logic Systems
	Collision	Time-Slot IEEE 802.15.4e & CSMA-CA	Fuzzy Logic Systems
	Tampering	secure firmware	Neighbor voting protocols
	Cloning	secure element	Neighbor voting protocols
Link	Exhaustion	Protection for joining devices	Fuzzy Logic Systems
	ACK	ciphering ACK frame with IEEE 802.15.4e	Secure Implicit Sampling
Network	Selective Forwarding	Secure Routing algorithm	Watchdog Approach, Monitoring using Source routing
	Sinkhole/Blackhole	Trust & Reputation	Watchdog Approach, Fuzzy Logic Systems
	Sybil	Authentication, Updating the keys	Centralized authority
	Hello flood	Challenge/Response with Cookie	Bi-directional verification of links, Energy monitoring
	Routing Cycles	Routing algorithm adapted to the topology	Traffic Analysis
	Wormhole	Authentication, Secure Routing algorithm	Connectivity information, Statistical Scheme Monitoring using Source routing, Energy monitoring
Application	Flooding	Freshness, Filtering of incoming packets	Game theory, Statistical Scheme
	Desynchronisation	Centralized clock	Neighbor monitoring, Genetic Algorithms
	Injection	Authentication	Traffic Analysis, Game theory, Statistical Scheme
	Replay	Freshness & Nonce & encryption	Game theory, Statistical Scheme
	Fuzzing	Freshness, Filtering of incoming packets	Traffic Analysis, Genetic Algorithms
	Over the Air Re-Programming	Encryption, Error Correction	Statistical Scheme

C. *Detection-based mechanisms*

The survey [3] proposes a classification of IDS into three categories: 1) misuse detection which is based on a database of patterns of known attacks; 2) anomaly detection which flags any mote activities that deviates from what is expected as a normal behavior; 3) specification-based detection which analyzes protocols deviations via machine learning techniques or training data. While misuse detection techniques need substantial amount of memory for storing the patterns of known attacks, specification-based techniques require computing power and rely on learning, few adapted to the hostile and constrained environment that generally is the real context of WSNs. Thus, the anomaly detection technique is the most widely used and a complete survey is dedicated to it [4]. This paper divides anomaly detection techniques as they are prior-knowledge based or prior-knowledge free, and according to the flat or hierarchical topology of the network. Each algorithm presented seeks to unmask an attack and can be embedded in a constrained and remote object. But, the capitalization of several techniques in order to cover the detection of several types of attacks in a WSN remains in practice inadequate to the field of embedded systems.

In the following, we use the ontology applied to WSN's attacks proposed by [5] to take an inventory of the major known attacks at each communication layer, as well as technical prevention and detection that can be used to counter them.

Prevention and detection techniques presented Table I concern security attacks. But these IDS, designed for wired networks, are resource-intensive and are more focused on security than on privacy issues. Their adaptation in a network holding constrained resources such as memory, computing power, energy or bandwidth, is tricky.

III. The data set capture and analysis

This section presents the IoT platform implemented and experimental process.

A. *Sniffer*

To eavesdrop ZigBee communications, we use the `killerbee` framework on top of an AVR RZ RAVEN chip. It provides tools to launch attacks on ZigBee WSNs. It is designed to passively intercept frames and to launch active attacks exploiting security vulnerabilities known to the IEEE 802.15.4 standard. The implementation of the PHY and MAC layers of the IEEE 802.15.4e version 2011 [7] and its amendment in 2012 [8] corrects many of these known security problems and must be fully implemented. Despite these improvements, successful eavesdropping is still possible [9]. We show that the full topology of a secured ZigBee WSN can be reconstruct by a single `killerbee` chip in passive listening mode. To achieve this, we use a specification-based technique without learning, considering the known join, association and data exchange protocols of the ZigBee standard.

B. *IoT platform including a ZigBee WSN*

Our "victim" ZigBee WSN is composed of a ZigBee Gateway implemented on a Raspberry Pi that performs a bridge between the Internet and the WSN (Fig. 3). The WSN is routed over Waspmote motes embedding an XBee front-end with the ZigBee PRO protocol, which supports only the mesh routing.

From a remote terminal, an end-user can query the sensing motes via a RESTful environment. A request is sent using the HTTP protocol through an URI:
http://199.166.244.133:8080/butler/device/⟨device_name⟩
/service/⟨service_name⟩/resource/⟨resource_name⟩

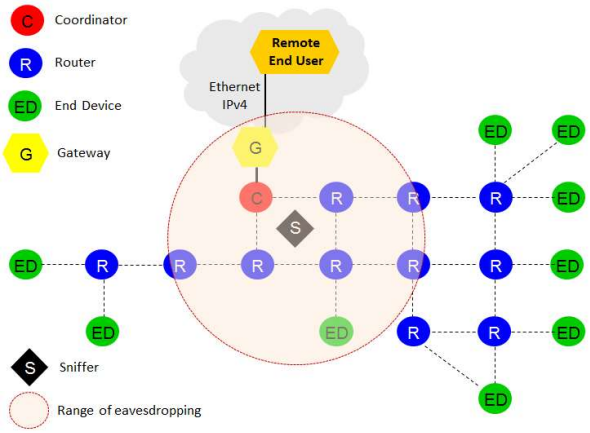

Figure 3: IoT platform including a ZigBee WSN.

C. Description of the experiments

We conducted three experimental campaigns which follow the same process: the ZG embeds an OSGi [3] which is connected to the ZC. The ZR and ZED are powered the one after the others to form a meshed network. The sniffer is placed near the ZC and eavesdrop the wireless communication into its range in the network as shown on Fig. 3.

Once performed the join and the association of devices, a script is executed on the end-user terminal to access successively to various resources located in the WSN. The sniffer intercepts the traffic in its range in the WSN to form a dataset containing around 3000 IEEE 802.15.4 frames. The dataset is then analyzed with Wireshark.

In the first experiment, no security is activated in order to build a reference scheme. The second experiment is launched with the security enabled and the provision of a NTW key in the ZC. In the third experiment, both NTW and Link keys are provisioned.

D. Data set analysis

The analysis of the intercepted frames is performed using Wireshark [4]. Wireshark is an open source software that supports many communication standards including IEEE 802.15.4. The data set analysis is lead in several steps. The first step consists in identifying the erroneous frames with bad Frame CheckSum (FCS) or MIC, and knowing whether the frames are encrypted. The second step consists in extracting and grouping the frames belonging to the same protocol instance such as join, association, route discovery or data exchange. This is done using regular expressions looking for typical values or patterns into the frame headers. Typical values are detailed on Fig. 4. The third step consists in a depth analysis of the metadata composed of the header content of each layer of the frames constituting a complete protocol instance.

```
wpan.cmd == 0x07 (Beacon Request, Join protocol),
wpan.frame_type == 0x00 (Beacon, Join protocol),
wpan.cmd == 0x01 (Association Request, Association protocol)
wpan.frame_type == 0x02 (Ack, Rejoin & Association protocol),
wpan.cmd == 0x04 (Data Request, Association protocol)
wpan.cmd == 0x02 (Association Response, Association protocol)
zbee.nwk.cmd.id == 0x08 (Link Status, Route Discovery protocol)
zbee.nwk.cmd.id == 0x01 (Route Request, Route Discovery protocol)
zbee.nwk.cmd.id == 0x02 (Route Reply, Route Discovery protocol)
```

Figure 4: Example of values identifying some type of frames

IV. Data leaks in insecure ZigBee WSN

This section deals with insecure ZigBee network.

A. Join & Association phase

When a device wants to join a nearby ZigBee networks, it broadcasts a beacon request (PAN scan). All nearby ZC and ZR, operating in a given channel and already belonging to the corresponding WSN, respond by sending the PAN ID and the join permission status. The device selects the valid ZC or ZR that responds with the best Link Quality Indicator (LQI) and sends an association request. It receives a 16-bit network address randomly selected by the device that has allowed the join.

The *Beacon Request* and *Beacon Response* messages exchanged during the join are MAC Beacon frames. For both frames, no security can be applied by the upper layers. The field *Super Frame Specification* of the MAC header handles information such as *PAN Coordinator* and *Association Permission*: eavesdropping those fields allow an adversary to recover the role of the devices in the network (both at 1 for a coordinator). Then, the association is initiated with a MAC Command frame *Association Request* that informs on the node capabilities thanks to the field *Capability Info*. If the association is allowed, the 16-bits short address is sent to the mote in the payload of the MAC Command message *Association Response*.

At this stage, several leaks are observed:

- The mote can be provided with the PAN ID of the WSN it should join. When it holds no PAN ID, it selects the one of the neighbor ZC or ZR with the best LQI that allows the join. In presence of several WSNs, such a mote has no way to recognize its legitimate WSN.
- A sniffer located near the ZC or ZR knows the PAN ID of the targeted WSN, the short and extended addresses of all the devices that have joined the WSN via this ZC or ZR, their type (ZR or ZED), their power source, their sleeping capability, the LQI value of the one-hop link, the parent/child relationship and their available resources.

[3] OSGi Alliance, http://www.osgi.org/Main/HomePage
[4] Wireshark, https://www.wireshark.org/

B. Data exchange phase

Let us consider an application in which a remote end-user sends a restful request to get a resource from a ZigBee object through an URI such as described in paragraph III-B.

The Route Discovery is launched when the source or a router, doesn't know the path until the destination. At the NTW layer, *Route Request* and *Route Reply* commands are included into a MAC Data frame to build the routing path. *Link status* commands are then periodically sent by ZC and ZR to maintain the routing tables up to date. Listening to the *Link Status* frames enables the deduction of the short addresses and the roles of the motes located in the sniffer area. A request sent by an end-user enters in the WSN through the Gateway and the ZC that builds the frames to route the request to the resource owner object. The response containing the resource value takes a back path.

The eavesdropping of the exchanges gives the ability to deduce the network topology, the mote activities and capabilities, the valid routes. By this way, the traffic can be dynamically monitored.

V. Data leaks in a secure ZigBee WSN

We analyzed the data leaks of a secure ZigBee WSN.

A. Join & Association phase

The messages exchange during the join and association protocols are MAC frames. They do not include NTW information, and so are neither ciphered nor authenticated. The NTW key is sent in clear from the ZC (or an intermediate ZR) to each joining device. If the key is eavesdropped, the whole security is compromised (confidentiality, integrity and authentication). This is a serious threat to privacy which brings back to the case of an insecure network.

Our analysis of the data captured during our third experiment shows that the use of a pre-installed Link Key avoids the leak of the NTW key because it is sent encrypted. When the NTW payload is ciphered by the NTW key, the results are similar to our second experiment. Unfortunately, sending the NTW key encrypted with pre-installed Link key is only available with ZigBee PRO version.

B. Data exchange phase

As the Route Discovery protocol and *Link Status* control frames are launched at the NTW layer, their content is ciphered. The routing path will be deduced by traffic analysis thanks to the metadata of the messages exchanged each time a resource is requested. The NTW header includes the source and destination short address and the *Radius* that is decremented at each hop from the maximum number of hops. The maximum number of hops is the same for all the devices of the WSN. It is easily known by listening at a packet issued from the ZC. The ZC may be identified thanks to its NTW address (0x0000) or by the analysis of the common source of requests. The value of the *Sequence Number* included in the NTW header remains the same for a given message from its source to its finale destination. By observing the traffic of a ZR device, this field allows to associate the incoming and the outgoing packets. Closer is a mote from the ZC, higher is its activity. Moreover, while the next hop extended address is updated at each hop, the listener is able to follow the message into the WSN and to deduce the routes to access the resources, by the observation of the traffic of each ZR.

When the resources are located out of the range of the sniffer, as for the WSN described Fig. 3, the outgoing and incoming packets on the ZR element located on the border of the listening area will bring sufficiently information thanks to their metadata to deduce quite accurately the topology of the remote devices and to monitor the activity. For a request frame routed outside the listening area, the expected response frame will swap NTW source and destination addresses. Analyze the *Radius* may indicate how many ZR it has crossed.

The topology of the WSN reconstructed after the traffic analysis of the metadata is detailed on Fig. 5. The resource request for the mote ZEDu crosses the router ZRa on the border of the listening area. The incoming response listened on the router ZRa indicates that the packet has already jumped two hops. So, ZEDu is necessarily located beyond ZRc. The resource request for the mote ZEDv leaves the listening area through ZRa. The associated response indicates that one hop has been jumped. This allows to reconstruct the back path through ZRb, but the path taken by the request is not necessary the same and remains unknown.

This example highlights that enabling the security both at NTW and APS layers does not preserve the privacy in the ZigBee WSN because many privacy leaks come from the NTW header content. The devices and their activity remain traceable. Moreover, any device may join the WSN when a Link key is not pre-installed, that is most often the case, because the use of the ZigBee Link keys is not user-friendly!

VI. How to preserve privacy?

The ZigBee PRO XBee front-end of the Waspmote devices does not provide the capability to enable the security at the IEEE 802.15.4 MAC layer. However, it should be the best way to preserve information regarding the privacy. But this implies to develop a new radio chip including the new IEEE 802.15.4e specifications for security purpose and compatible with the ZigBee standard for the higher layers.

By this way, the MAC Command frames exchanged during the association protocol could be ciphered and authenticated, allowing only legitimate devices to join the WSN, and masking the *Capability Info* field and the short address of the joining devices. These cryptographic functions could be performed in hardware in the radio front-end and be transparent for the ZigBee higher layers. Another benefit is that the transmission of the NTW key would be ciphered

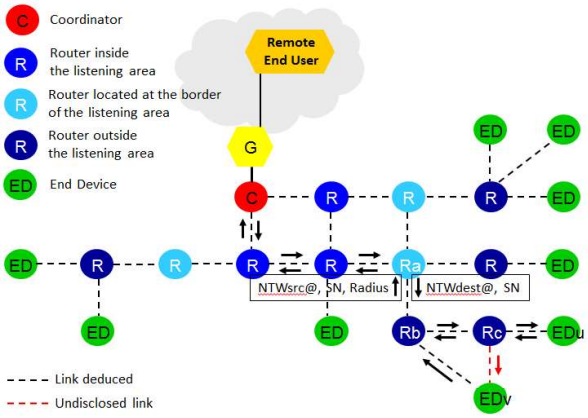

Figure 5: Reconstructed Topology of a secured ZigBee WSN

even without provisioned Link key or trust center.

The route discovery and the path disclosure would be protected at the cost of hardware deciphering and ciphering again at each hop by the ZR elements. The traceability of the devices, the monitoring of their activity and the reconstruction of the topology of the WSN based on the use of NTW header information would be avoided. Nevertheless, the MAC header content remains in clear and includes private information on the mote neighborhood, notably the MAC addresses of the neighbor motes. Whatever the layer where the security is enabled, the one hop MAC addresses cannot be hidden. To preserve the privacy at this layer, one solution consists in the use of pseudonyms for the MAC addresses. The Incoming/Outgoing rate provides information on the mote role and activity. A ZR should route all the incoming frames. A ZED may send only outgoing frames as application resources. To mask the activity, several techniques can be used. Random delay can be introduced between the frame reception and its routing. The data could be aggregated. Motes may emit periodically even if they have no relevant data to disclose.

For SmartHome applications, or more generally static WSN, another precaution consists in making easily configurable by the owner the register *Join Permission*. Thus, this header field can be set to the value 1 when new motes are deployed, then reset to 0. The transmission of this command to the motes ZC and ZR located in the WSN must imperatively be secured by encryption.

A user-friendly ZigBee trust center should be designed, enabling securely the secret keys distribution, the key management and renewal, the detection of anomaly or intrusion, the repudiation of an element, the resilience to attacks.

VII. Conclusion

We have deployed an IoT platform including a secured ZigBee WSN. We have used `killerbee` to eavesdrop the traffic exchanged between the motes and the network coordinator. Using Wireshark and targeted regular expressions, we have analyzed in-depth the datasets collected during three different experiments. Then, we succeed to reconstruct the network topology from the metadata exchanged both in clear and ciphered messages and to identify the role of the motes, their capabilities and activity. These severe privacy weaknesses are a critical obstacle to the deployment of ZigBee WSN at large scale, especially for SmartHome applications.

To counter these issues, we propose simple and pragmatic solutions to preserve the privacy into ZigBee WSNs. The future work will be dedicated to assign dynamic pseudonyms.

Acknowledgment

This research work was supported by the FP7 European projects SocIoTal under contract no. 609112 and by the Celtic TILAS project (C2012/1-9), partially funded by the French National Authorities.

References

[1] Christine Hennebert, Jessye Dos Santos, Security Protocols and Privacy Issues into 6LoWPAN stack: A synthesis, IEEE Internet of Things Journal Issue, Vol 1 Issue 5, october 2014, pp. 384-398 , DOI: 10.1109/JIOT.2014.235953.

[2] ZigBee Specification, document 053474r17.

[3] Abror Abduvaliyev, Al-Sakib Khan Pathan, Jianying Zhou, Rodrigo Roman and Wai-Choong Wong, On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks, IEEE Communications Surveys & Tutorials, vol.15, no.3, pp.1223-1237, 2013, DOI: 10.1109/SURV.2012.121912.00006.

[4] Miao Xie, Song Han, Biming Tian and Sazia Parvin, Anomaly detection in Wireless Sensor Networks: A survey, Journal of Network and Computer Applications, vol.34, pp.1302-1325, Elseiver, 2011, DOI: 10.1016/j.jnca.2011.03.004.

[5] Wassim Znaidi, Marine Minier, Jean-Philippe Babau, An Ontology for Attacks in Wireless Sensor Networks, [Research Report] RR-6704, 2008, https://hal.inria.fr/inria-00333591.

[6] Björn Stelte and Gabi Dreo Rodosek, Thwarting Attacks on ZigBee - Removal of the KillerBee Stinger, 9th International Conference on Network and Service Management, 2013, pp.219-226, DOI: 10.1109/CNSM.2013.6727840.

[7] IEEE Standard for Local and Metropolitan Area Networks 802.15.4: Low Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard 802.15.4-2011, Sep. 2011.

[8] IEEE Standard for Local and Metropolitan Area Networks 802.15.4: Low Rate Wireless Personal Area Networks, MAC Sub-Layer, IEEE Standard 802.15.4e-2012, Amendment to IEEE Standard 802.15.4-2011, Apr. 2012.

[9] Mohammed Abdul Qadeer, Mohammad Zahir, Arshad Iqbal and Misbahur Siddiqui, Network Traffic Analysis and Intrusion Detection using Packet Sniffer, IEEE 2nd International Conference on Communication Software and Networks, 2010, pp.313-317, DOI: 10.1109/ICCSN.2010.104.