A Middleware Approach to Provide Security in IEEE 802.15.4 Wireless Sensor Networks

Stefano Marchesani, Luigi Pomante, Marco Pugliese, Fortunato Santucci
Center of Excellence DEWS
University of L'Aquila
L'Aquila, Italy
stefano.marchesani@graduate.univaq.it {luigi.pomante, fortunato.santucci}@univaq.it, marco.pugliese@ieee.org

Abstract— **Last years have seen the growth of interest for middleware exploitation in distributed resource-constrained systems as Wireless Sensor Networks (WSNs) are. A WSN is a versatile smart sensing system to support pervasive monitoring in a variety of applications. In this context available middleware platforms usually provide the Application Layer with different basic services, as shared memory or addressing repository, but do not usually provide security services such as secure links management protocol or intrusion detection. Nevertheless, since WSN applications normally require the collection and the aggregation of reliable measurements and data from the sensing units, secure communications should be guaranteed even in the presence of resource constraints. In this paper we then present a novel middleware approach that is directly tailored to an IEEE 802.15.4-based WSN. The security-related components of the proposed middleware include a light yet powerful cryptographic scheme (TAKS) and an Intrusion Detection System (WIDS): the former module exploits the topological properties of a WSN, while the latter one is based on a Weak Process Model approach.**

Keywords- security; middleware; wireless sensor networks; cryptographic scheme; intrusion detection system

I. Introduction

In recent years, a new wave of networks labeled Wireless Sensor Networks (WSNs) has attracted a lot of attentions from researchers in both academic and industrial communities. WSNs can be used to form the underlying sensing and network infrastructure in pervasive computing environments. A WSN consists of a collection of sensor nodes and typically a sink node connected through wireless channels, and can be used for building up distributed systems for data collection and processing, that encompass functionalities of on field signal sensing and processing, in-network data aggregation, and self-organized wireless communication. WSNs have found many applications in different areas, including environmental surveillance, intelligent building, health monitoring, intelligent transportations, and so on [28].

In the depicted context, that is typically resource-constrained, particular attention has been devoted to development of middleware platforms. A middleware is a software platform used to hide complexity and heterogeneity of the underlying physical platform and network and to offer several services to the *Application Layer*, eventually providing an application execution environment [29]. When the underlying physical network is a WSN, considering typical monitoring oriented applications, data and system reliability are also required. Although security is not usually included in the services portfolio by middleware platforms for WSNs, reliability involves security issues: so, a middleware for WSNs should not ignore aspects such as secure data transmission and intrusion detection.

This paper deals with the definition and development of a new middleware framework to provide security in WSNs: in particular, an architecture for the middleware is proposed and main design choices are discussed. Moving from our previous work, we focus on a *hybrid cryptographic scheme* called TAKS and an *Intrusion Detection System* (IDS) based on a simplified version of *Hidden Markov Models* (HMMs) called *Weak Process Models* (WPMs). The relevant feature of the presented work is related to the fact that the proposed architecture is tailored to real-world IEEE 802.15.4-based WSNs.

The remainder of this paper is organized as follow: Section 2 deals with background and motivations that have led us to propose a new middleware and also state-of-art about middleware platforms for WSNs that handle security is reported. Section 3 deals with the provision of security services for IEEE 802.15.4 networks and Section 4 with the proposed middleware architecture. Section 5 and 6 are focused on the secure transmission service (which refers to a WSN-oriented cryptographic scheme), the intrusion detection service and, specifically, the issues related to implementation on the protocol stack. In Section 7 some conclusive comments and future works are reported as well.

II. Background and Motivations

Usually, WSNs are used in monitoring and control applications wherein energy consumption is very constrained. Often, nodes are battery supplied and not accessible, so the energy consumption should be carefully considered. Moreover, a WSN should be flexible especially with respect to node heterogeneity. In fact, the burden of computation may vary from node to node and the exploitation of a heterogeneous network setup could be beneficial.

IEEE 802.15.4 has been designed to achieve these goals. It is a standard which specifies the *Physical Layer* and *Media Access Control* (MAC) for *low-rate wireless personal area networks* (LR-WPANs) [5]. A LR-WPAN is a simple, low-cost communication network that allows wireless connectivity in

MOBILWARE 2013, November 11-12, Bologna, Italy

DOI 10.4108/icst.mobilware.2013.254275

applications with limited power and relaxed throughput requirements. The main objectives of a LR-WPAN are easy installation, reliable data transfer, short-range operation, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol.

By defining the *Physical Layer* of the protocol stack, the standard allows to different IEEE 802.15.4 compliant nodes (e.g. from different manufacturers) to correctly communicate. Therefore, the exploitation of this standard implicitly gives the capability to manage heterogeneous WSNs.

Moreover, IEEE 802.15.4 provides to the higher layer two types of services: the *MAC data service* and the *MAC management service*. The former one provides services to exchange data in the network, while the latter one allows to handle network management issues such as synchronization, network formation and maintenance (e.g. *scan* and *association/disassociation*), etc.

Generally, these services are exploited by the *Network Layer* to provide multi-hop through routing table or smart address managing in association/disassociation. There are a lot of *Network Layer* suitable for IEEE 802.15.4, with some examples reported in [34] and [38].

It is worth noting that by providing a *Network Layer* on the top of the IEEE 802.15.4 *MAC Layer*, we have just provided a simple middleware suitable for typical monitoring and control WSNs applications. A software package of this type helps to hide the complexity and heterogeneity of the underlying hardware and network platforms and simplify the management of system resources: in other words, it could be considered as a middleware. This remark is also compliant with the classification of middlewares for WSNs provided in [36] and [37]. Nevertheless, the middleware discussed in this paper is more complex with respect to the approach devised above. In addition to providing methods to route and control the medium access through IEEE 802.15.4, our framework embeds a method to ensure reliability of the network based on the exploitation of a hybrid cryptographic scheme and an intrusion detection system. It is worth to note that in [1][2][3], we have proposed an architecture for a middleware where security services are embedded in the mobile agent-based middleware *Agilla* [4]. Here, part of the same considerations are moved in a different context for a different purpose: in [1][2][3], the middleware is unaware of the underlying physical network, while in our current proposal security services are tailored to a system prototype that explicitly relies on the IEEE 802.15.4 standard.

This approach is quite different when compared to other related works. For example, *Zigbee* [34] aims to standardize the application execution environment covering the largest number of WSN application domains and providing some basic services. In fact, although Zigbee is famous for its definition of *Network* and *Application Layers* to put on top of IEEE 802.15.4 ones, its specification includes a number of basic security provisions and options. In particular, Zigbee provides facilities to manage secure communications (for instance, link setup and key management), ciphering frames and controlling devices. Instead, the middleware proposed in this paper aims to provide advanced security services focusing on the most popular WSN application domain, i.e. monitor and control applications. In literature, other than ZigBee, there are several proposals of middleware platforms that provide security through cryptography: for instance, *SM-Sens* [30] uses symmetric and asymmetric cryptography along with message authentication code to ensure security requirements on data flows. It also provides a method to distribute keys by exploiting hierarchical routing and a mechanism to exclude compromised nodes of the network. *STaR* [31] is a modular, reconfigurable and transparent software component for secure communications in WSNs. STaR guarantees confidentiality, integrity, and authenticity by means of encryption and/or authentication and it is totally transparent to the application, i.e. no changes to the original application or the communication protocol are required. *SpartanRPC* [32] extends nesC programming language to provide a *link-layer remote procedure call* (RPC) mechanism. All the RPC resources are protected via language-level policy specification. *SMEPP Light* [33] features group management, group-level security policies, mechanisms for query injection and data collection based on a subscribe/event mechanism, and adaptable energy efficiency mechanisms. Other middleware proposals provide security by deploying specific defense mechanisms for a set of predefined attacks. *Di-Sec* [35], for example, provides a framework to model defense strategies. Through a training phase, nodes are able to learn the behavior to adopt in case of attacks.

The middleware proposed in this paper exploits benefits of these two approaches (i.e. cryptography and attack defense mechanisms) by providing a light but powerful cryptographic scheme to protect data and an intrusion detection system to guarantee the availability of the network. Moreover, since it explicitly refers to IEEE 802.15.4-based WSN, the middleware is suitable for network composed by heterogeneous nodes. Usually, this property is not considered in middlewares that handle security but it turns out to become very interesting to deploy a real-world WSN.

III. PROVIDING SECURITY IN IEEE 802.15.4 NETWORKS

To discuss the security facilities provided by IEEE 802.15.4 it is important to highlight that the standard does not provide only a method to access the medium, but also several mechanisms to create and detect a WPAN, associate or disassociate with it and so on. This means that in a 802.15.4 network, there are several types of messages exchanged, such as *beacon packets* (used to synchronize the network), *acknowledgments packets* (used to notice the message reception), *data packets* and other *control packets*.

As specified by the standard [5], the *MAC Layer* provides security services on each incoming and outgoing frame (with the exception of acknowledgement packets). The services supported by the standard are as follows:

- *data confidentiality*;
- *data authenticity*;
- *replay protection*.

Data confidentiality is ensured by using encryption and decryption algorithms: the standard defines to use AES (*Advanced Encryption Standard*) with 128 bit keys. Data authenticity is guaranteed using cryptographic hash functions that associate to each message a *Message Authentication Code*

(MAC). The receiver can check it to authenticate the message. Finally, the usage of a monotonically increasing sequence number to each packet ensures the protection from replay attacks. It is worth noting that the standard can also work with no security, encryption only (AES-CTR), authentication only (AES-CBC-MAC), and encryption and authentication (AES-CCM). Each category that supports authentication comes in three variants depending on the size of the *Message Authentication Code* that it offers. Each variant is considered as a different security suite and has its own name. The *Message Authentication Code* can be either 4, 8, or 16 bytes long.

The IEEE 802.15.4 specification provides basic security mechanisms but these security features cannot work on their own: since the standard does not suggest any key management approach, in applications that require security a method to generate *symmetric* keys is needed. Symmetric key generation is one of the most addressed problem in the literature [6]. Pair-wise key pre-distribution solutions are based on deterministic pre-distribution of keys for each pair of nodes. Random pair-wise key schemes are based on storing only a subset of all possible keys in each node. To communicate with each other, every node needs to negotiate a key with its peer, randomly selecting one key in its subset [7]. The master key pre-distribution scheme requires that a master key is distributed in the entire network and that nodes use a combination of it and previous exchanged nonces [8]. Other schemes can be found in [6]. In [2][9][10], we have proposed a family of novel schemes called TAKS (*Topology Authenticated Key Scheme*) to generate topology authenticated keys for handling cryptographic aspects in resource constrained deployments of WSN. TAKS cryptographic schemes allow to authenticate each message exchanged in the network referring to a certified topology of the network. Since TAKS provides good results from both performance and security points of view [10], its usage in IEEE 802.15.4 networks is very encouraged.

The complexity and distributed nature of a WSN makes cryptography not sufficient to ensure network security. In addition, to provide confidentiality, authenticity and integrity of messages, network security aims to make the system always (or mostly) available. Ensuring availability is more complex than other issues. Typically, this is done by auditing network activities, detecting potential threats and reacting opportunely through an *Intrusion Detection System* (IDS). IDS denotes a system that is capable of supporting mechanisms to detect and appropriately manage (through reaction functions and proper countermeasures) intrusions and attacks in the form of malicious control and data messages [12]. An IDS is typically formed by three components: *Intrusion Detection* (ID) that deals with the detection of network intrusions by sensing suspect phenomena, *Intrusion Reaction Logic* (IRL) that schedules the priorities for actions on all compromised nodes according to a specific defensive strategy and *Intrusion Reaction Application* by performing the appropriate countermeasures (IRL Application).

In this paper, we focus on Intrusion Detection and we do not deal with Intrusion Reaction. For what concerns Intrusion Detection, an IDS can be classified into three frameworks: *anomaly based* intrusion detection, *misuse based* intrusion detection and *specification-based* intrusion detection [11][13]. Anomaly based intrusion detection relies on the assumption that intruders will demonstrate abnormal behavior relative to the legitimate nodes: anomaly has to be detected by knowing the normal system behavior. Instead, the misuse intrusion detection relies on the assumption of an up-to-date database of intrusion signatures. Using them, the system can easily detect intrusions on the network. Specification-based detection systems work by defining rules for attacks. Sensor node's behavior is checked against each rule sequentially. There is a failure counter associated with each node. If the sensor node violates any rule, failure counter is incremented. If number of failures of a particular node increase over a threshold after a time interval *t* an alert about that node is generated.

Another typical IDS classification is done on the distribution of code in the network [13]. There are the following type of IDS: in *purely distributed* IDS intrusion detection, algorithm is installed in every node; in *centralized* IDS, intrusion detection is performed only by the sink node or base station upon the reception of processed information from the network; in mixed *distributed-centralized* IDS, suitable only for particular types of networks, such as clustered WSN, the detection is delegated to a particular subset of nodes of the network. Examples of these types of IDS are [14][15][16].

Although many of these approaches can be applied in IEEE 802.15.4 WSNs, we cannot provide IDS examples focused on these networks: many works survey attacks and propose methods to detect them, such as [40], but, at the best of our knowledge, there are no papers proposing IDS frameworks specifically focused on these kind of networks.

IV. The IEEE 802.15.4-based Middleware Architecture

This section deals with the main functional blocks of the proposed IEEE 802.15.4-based middleware. A high-level representation is given in Fig. 1.

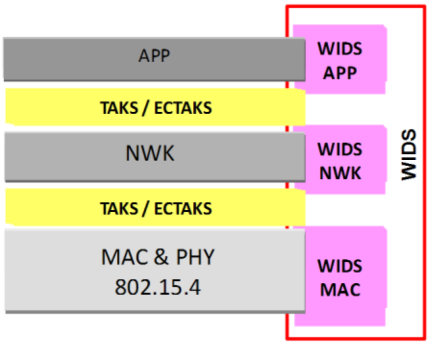

Fig. 1 The middleware architecture

The proposed architecture refers to a typical WSN protocol stack, where security facilities are now considered and embedded in the framework. From a protocol point of view, the proposed architecture specify only *MAC* and *Physical Layers* to IEEE 802.15.4 and provides flexibility of choices on both *Network* and *Application Layer*. Usage of IEEE 802.15.4 is not limitative because it is the de-facto standard in industrial applications while *Network Layer* is not standardized as well, although Zigbee exploitation is quite diffuse.

In the following, the embedding of the security services into the reference middleware architecture is discussed. Typically, data security is accounted at the *Application Layer*: a large number of protocols, such as ones used in Internet, provide security directly at *Application Layer*. The proposed architecture is compliant to this view because it provides TAKS [9] facilities (ECTAKS if elliptic cryptography is exploited [10]) to *Application Layer*. From a *Network Layer* point of view, this means that only the effective payload (i.e. the Application packet) is encrypted and only the intended legal receiver can decrypt it. However, IEEE 802.15.4 has some interesting properties that we can exploit to improve the security level. As we have seen in previous sections, 802.15.4 *MAC Layer* provides a security suite accessible by means of APIs provided by the *MAC Layer* itself. Using this service we can ensure the encryption of each MAC data packets (i.e. the entire MAC payload composed by Network and Application packets). We have designed the middleware so that encryption can be selected at one or both layers or depending on user security requirements.

The proposed architecture motivates further remarks about IDS. Our consolidated line of research is oriented towards a misuse based purely distributed IDS which exploits the Weak Process Models (WPM) over WSN, denoted here as WIDS (WPM-based Intrusion Detection System) [1][2][3]. First of all, WIDS is purely distributed. Most literature contributions propose to put intelligence (usually more consuming both in computational resources and in memory) outside the WSN [18][19]: however, if the algorithms are designed by considering the very constrained environment of WSNs, these systems can operate as functionally “autonomous entities” and not only for pure sensing operations. This choice implies two types of benefits: the former one consists in the distributed architecture which avoids the typical drawbacks presented by centralized solutions; the latter one is the reduction in energy consumption since distributed solutions do not need sharing information with a centralized entity (i.e. sending them via radio and wasting energy). However, the drawback is that distributed IDSs need a fine configuration.

Looking at the architecture, it is straightforward to remark the cross-layer nature of the Intrusion Detection System that concerns all active layers of the stack (i.e. Application, Network and MAC). Each active layer implements protocols which are characterized by a set of constraints and rules and, for this reason, exposed to attacks by intruders: constraints and rules in a protocol represent points of weakness which can be exploited by intruders to induce altered behaviors on network nodes (e.g. a denial of service). For example, we can refer to a kind of attack known in literature as *HELLO flooding*. This kind of attack relies on the fact that wireless protocols often require that nodes execute an association procedure by sending the so-called HELLO messages. HELLO Flooding is when the attacker continuously issues malformed HELLO messages to WSN nodes, which waste computational and memory resources that can later result in a denial of service. Now it is easy to understand how IDS is strictly based on the kind of protocol that it monitors. Therefore, the WIDS component of the middleware is conceptually the same one that we have proposed in [1][2][3] but it is instantiated for the different protocols provided by the actual architecture. It is important to remember that, such an architecture, do not define any *Network* and *Application Layer* so, in this work, we do not provide any fixed approach to handle Intrusion Detection at these layers.

In next sections we will give a complete overview of both WIDS design, focusing on the *MAC Layer* of the stack, and the cryptographic scheme.

V. Weak Process-based Intrusion Detection (WIDS)

A. *Motivations and Logic*

WIDS main function is to identify abnormal network activity that differs from the expected behavior. In [1] and [2] we have shown how a light state-based anomaly-based detection logic can be suited to be implemented over WSNs. An *Hidden Markov Model* (HMM) [20] is a doubly stochastic finite state machine with an underlying stochastic process that represents the real state of the system: the real state of the system is hidden but indirectly observable through another stochastic process that produces a sequence of observable events. The relationships between hidden states and observable data are stochastic as well as the transitions among states. While detailed description of models and proofs can be found in [1] and [2], we would like to emphasize here that our consolidated line of research is oriented towards an anomaly detection logic which exploits *Weak Process Models* (WPM) [1] over WSN: WPMs are non-parametric version of HMM, wherein state transition probabilities are reduced to rules of reachability in a graph representing the abnormal behaviors. The estimation of a threat in the case of weak processes is greatly simplified and less demanding for resources. The most probable state sequence generated by the Viterbi algorithm [21] for HMM becomes the possible state sequence generated by simplified estimation algorithms for WPMs.

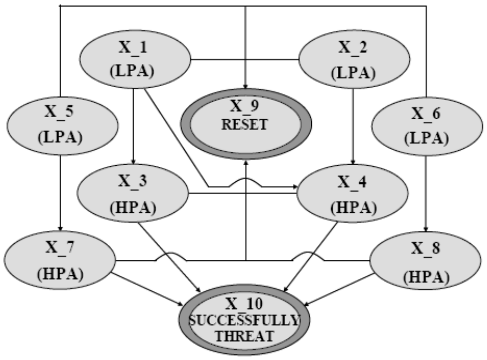

Fig. 2 An example of WPM-based Threat Model

The intensity of the attack is evaluated by introducing a threat score, a likelihood criterion based on weighting states and transitions. In [1] we introduced two classes: LPA (Low Potential Attack) and HPA (High Potential Attack). An attack is defined in a “low potentially dangerous” state (or in a LPA state) if the threat is estimated to be in an early stage, otherwise it is defined in a “high potentially dangerous” state (or in a HPA state) if the threat is estimated to be close to its completion. WIDS module identifies any observable event correlated to a threat by applying a set of anomaly rules to the incoming traffic. Attacks are classified into LPA or HPA according to specific states in the corresponding WPM-based threat model (Fig. 2). Alarms are issued as soon as one or more high potential attacks are detected.

The idea underlying the logic of our approach consists in auditing network activities and, at the occurrence of anomalies, generating state transitions on WPM. Generally, multiple traces can coexist, so it is very important to be able to handle all them. To avoid storing each trace in memory completely, we have defined a scoring mechanism of state transitions (not present in Fig. 2 to simplify the image). This allows to recognize the achievement of LPA or HPA studying the exceeding of threshold. It is worth noting that this is done without save in memory the entire trace on WPM with consequent memory saving.

Moreover, next sub-section shows that its core allows to WIDS to be dynamically configured, eventually at runtime. This property is also provided by traditional IDS, but in this context it has a different mean. In traditional network, IDS dynamicity means the capability of the system to perform configurable detection rules on a single type of input, typically IP packets. In such a context, since only logical attacks are of interest, the alarm is raised knowing only information stored in the packet and the IDS state. In WSN domain, the very unreliable environment suggests to log any anomaly, physical or logical. In fact, in WIDS, anomalies are different nature data such as packet not authenticated, negative clear channel assessment, corrupted packet reception, correct packet reception, etc. Therefore, a WPM is a representation of a threat and also the data structure on which we run the algorithm to track all possible anomaly sequences to raise alarm opportunely. To make WIDS sensitive to another threat, following operations are needed:

- definition of threat observables;
- design of the WPM modeling the threat;
- insertion of the new WPM in WIDS;
- insertion of WIDS notification points in the code.

B. *Implementation Issues of IDS*

This sub-section deals with definition of algorithms and data structure used to implement IDS. Previous sub-section suggests that needed elements are:

- a double Finite State Machine (FSM) which represent WPM;
- a set of current states and corresponding scores.

It is important to note that, for each layer, a single WPM is needed which eventually represents an aggregation of threats. From an implementation point of view, the key issue is how to represent a WPM. State space representation is computational expensive. Computational complexity to generate next state is $O(n^2 + nq)$, where n is the state space size and q is input space size. Moreover, state space representation require a lot of multiplications that are computationally expensive on basic microcontrollers. Some empirical results, that we have made on *MicaZ* motes, shows that is possible to calculate the dynamic of a WPM modeling three threats in ~10 ms. So, an efficient representation of WPM becomes very important. It is important highlight that a WPM extends FSM (somewhere WPM is said to be a "double FSM") in the sense that observables are in many-to-many relationship with states and therefore states result to be not directly observable, hence hidden. From an implementation viewpoint, the "physical" observable, i.e. the set of specific values assumed by the information elements contained in signaling messages, is mapped into the "logical" observables considered in the WPM though some state-less algorithm. According to the observable sequence, a set of possible state traces can be estimated and therefore system behavior. Generally a WPM represents a good analytic technique for IDS when behavior to be modeled is rather complex and many states have to be introduced, otherwise conventional FSM can be employed. Now, it should be clear that WPM can be implemented using FSM implementation techniques. It is possible to implement a FSM coding its transitions and actions directly in the code or coding them in a data structure [22]. To explain differences between these solutions we can think a software FSM as a function that upgrades the state on the base of current state and input. This can be done by directly coding in the function each state transition and corresponding action in a great *switch-case* or coding this information in a data structure. The first solution leaves a lot degrees of freedom to the programmer that is a pro and a cons at the same time. Moreover, this solution implies a greater memory occupancy due to repetition of pieces of code. For these reasons, generally, data structure coding representation is preferred. This implementation technique requires the representation of the state transition graph in a data structure and for this is more compact, regular and structured than the previous one. The basic idea is to code the state transition graph with standard graph data structure (i.e. an adjacency matrix or a list if the matrix is sparse). In the following, we consider the most general version of the data structure, i.e. suitable also for extended FSM version such as *statechart* [22]. In this scenario, each element of the data structure can contain:

- activities to do when reach in the state;
- activities to do until next state transition;
- activities to do when leaving the state;
- a description of the outgoing transitions from the state.

Using an adjacency matrix or an adjacency list the cost to generate next state is decreased a lot. In fact, it is proportional to the outgoing degree of current state. In worst case, outgoing degree is n, where n indicates the size of WPM state space, so the complexity is reduced to $O(n)$. Exploiting empirical results on *MicaZ* motes, we can say that, by means of an efficient implementation, it is possible to calculate dynamic of a WPM modeling three threats in ~50 us in worst case.

C. *Implications on stack layers*

This sub-section deals with IDS usage on stack layers: previously we have seen that, following a misuse-based approach, one of the main issue is to define a set of threats to model.

For what concerns IEEE 802.15.4 threats models, we have surveyed on typical attacks that can be executed on any *Physical* and *MAC Layer* [23], on wireless MAC layers [24] and on specific IEEE 802.15.4 networks [25][26]. In fact, some of the attacks (e.g. *radio jamming* and *link layer jamming*) are common to all *MAC Layer* definitions. Others, like *backoff manipulation* may also occur in IEEE 802.11 wireless networks due to some common properties in the *MAC Layer* implementations. Finally, we have planned to provide facilities to handle specific variants of some general attacks applied on IEEE 802.15.4 *MAC Layer* security mechanisms such as *replay-protection attack* and *ACK attack* [25].

For what concerns *Network Layer* threats models, since we do not have defined a true *Network Layer* in the architecture, it is not possible to design specific threats to model. Therefore, we have planned to handle typical routing attacks such as *HELLO flooding*, *Sinkhole* and *Wormhole* [27].

Application Layer ID, such as *Network Layer* ID, cannot be defined with respect to a specific protocol since the architecture does not define it. Moreover, the great variety of WSN applications does not allow to identify a meaningful set of application independent attacks such as in *Network Layer*. Nevertheless, if following a anomaly-based approach for *MAC* and *Network Layer* is unfeasible due to their complexity, this could not be true for *Application Layer*. In fact, since *Application Layer* protocols are typical simpler than the other ones, a different approach can be evaluated.

D. Considerations on IEEE 802.15.4 Threat Models and Detection Strategies

In this sub-section, some considerations on threat models and detection strategies are given. Supposing to have a beacon-enabled network [5] monitoring slow physical quantities, it is first analyzed how to detect radio and link layer jamming, replay-protection and ACK attacks, i.e. the main attacks suffered by IEEE 802.15.4 networks. Then, an aggregate WPM (Fig. 3) modeling these attacks is proposed and motivated.

Radio jamming. Jamming is basically creating radio interference that causes a denial of service on receiver or transceiver nodes. It is possible to perform different radio jamming attack strategies [39]: *constant* jammer, *deceptive* jammer, *random* jammer, *reactive* jammer. First two strategies continuously send out a radio signal while the remaining two ones alternate between sleeping and jamming. Therefore, approaches to detect them are different. IEEE 802.15.4 *MAC Layer* Specification [5] defines that some packets (i.e. beacon and ACK packet) do not require CSMA/CA for their transmission. Some timing constraints ensure the reception of these packets. Therefore, continuous jamming can be easily detected performing the clear channel assessment before sending beacon or ACK packets. To avoid false positive, the alarm should be raised only if the previous anomaly is detected for a certain number of times. In Fig. 3, a LPA alarm is raised when observable O_1 (i.e. the impossibility to transmit beacon or ACK messages) is detected while an HPA alarm is raised if it occurs again. To detect not continuous jamming, anomalies discussed above are evaluated with respect to the elapsed time. If anomalies are too frequent, an alarm is raised. In the WPM proposed, it can be observed that the LPA state (i.e. X_4 state) is reached when observable O_4 occurs. This happens when the number of medium access failures (observable O_2) in the timing window exceeds the fixed threshold. In Fig. 3, timing window elapsed is represented by observable O_3 that implies a step back on the WPM while the threshold exceeding is represented by observable O_4. An HPA alarm is raised if the anomaly is detected again.

Link layer jamming. Link layer jamming is the most complicated type among the jamming attacks. An intelligent adversary, who wisely uses the link layer protocol logics, can be as defective as a blind radio jammer. An example of link layer jamming is the *backoff manipulation attack*. This attack exploits the fact that a sender listens to the channel before transmitting its packet. If the channel is found busy the sender will defer its access by an amount of time which is called *backoff period*. The recent channel access is given to the contending node with the smallest backoff value. This value is randomly chosen from the range of the contention window which is enlarged exponentially for a node that finds the channel busy each time. An adversary node can take the advantage for channel access over legitimate nodes by not applying the protocol rules and constantly choosing a small backoff interval. Since the legitimate nodes select the rule-based backoff intervals, their chance of channel access would reduce exponentially. In beacon-enabled network, the start of the first backoff period of each device is aligned with the start of the beacon transmission. Moreover, at each beacon transmission, radio should be in receive mode. Therefore, greater backoff exponents means more energy consumed since radio duty cycle is increased. Detection of this attack is straightforward: it is sufficient to assess channel clarity of the frame (i.e. radio activities between two beacons) to evaluate busy anomalies about its first portions. The best moment to declare that a threat has been detected is dependent on transmission rate and size of the network. In the WPM proposed, backoff manipulation detection is performed each time a medium access returns failure since we refer to a scenario where this occurs rarely. In other scenarios, backoff manipulation detection can be performed, for example, at the detection of radio jamming LPA and so on. Fig. 3 WPM defines that, at the occurrence of medium access failure, the packet retransmission has to be performed at the beginning of the frame (state X_6) and, moving ahead on it, if transmission does not occur (i.e. observable O_2 is detected). Retransmissions have to be performed until HPA state (i.e. X_9 state) is reached. Vice-versa, if a transmission is correctly completed (i.e. observable O_5 is detected) the threat WPM is reset.

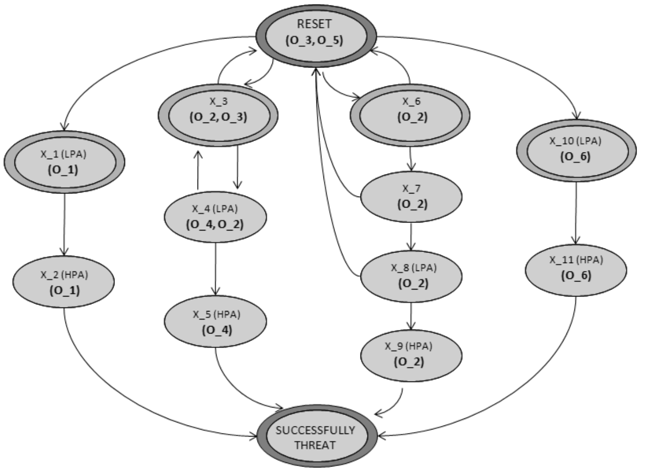

Fig. 3 IEEE 802.15.4 MAC Layer WPM threat model

Replay-protection attack. Replay protection mechanism is done by checking the counter of a recent message with the previous obtained counter. If the recent counter is equal to or less than the previous one, then the frame would be rejected. In the IEEE 802.15.4 specification, the replay protection mechanism is provided, but it is subjected to replay-protection attack which can be accomplished by an adversary via sending many frames containing large counters to a legitimate receiver. When another legitimate sender transmits a frame with a lower counter, it will be rejected according to replay protection procedure. Detection of this attack is very straightforward: it is sufficient to monitor frame counters and raise an alarm when anomalies are detected. This can be observed also by WPM of Fig. 3: if a frame counter anomaly (i.e. observable O_6) is detected a LPA alarm is raised while if it occurs again an HPA is delivered.

ACK attack. In the middle of a transmission between two legitimate users, an eavesdropper can listen to the un-encrypted sequence numbers of the frames. When the eavesdropper wants to prevent the legitimate receiver from getting a frame, it corrupts the frame by interferencing at receive time. Then, the eavesdropper sends a fake ACK frame with the related sequence number to the sender in order to fool the sender as if the ACK was coming from the receiver. Therefore, a sender cannot be sure if the received frame is coming from the receiver or another node even if the receiver correctly received the ACK frame. Since there is no integrity protection provided on ACK frames, this weakness should be addressed in higher layers, e.g. making able the receiver to send an authenticated acknowledgement to the sender.

VI. THE CRYPTOGRAPHIC SCHEME TAKS

A. *Motivations and Logic*

This sub-section gives a brief overview on TAKS. We have already seen TAKS as a scheme to generate symmetric keys in WSN. Its main property is the capability to generate keys based on the topology of the WSN. To generate keys based on topology of the network and consequently to authenticate messages exchanged among nodes respect to its topology is a very strong property that increases the security level in the network itself.

While a more detailed description can be found in [9][10], here the essential TAKS design principles are summarized. TAKS requires the offline setup of some topology-related parameters to allow their pre-distribution in the entire network: *Local Key Component*, *Transmitted Key Component*, *Local Planned Topology* i.e. a set of vectors (denoted as *Topology Vectors*) in one-to-one relationship with admissible neighbor nodes. To communicate with another one, a node needs to own the destination Topology Vector. In fact, TAKS allows to a node to generate keys combining its own Local Key Component and the destination Topology Vector. Supposing to distribute Topology Vectors on the network according to a planned network topology (i.e. the graph of allowed communication among nodes of the entire network defined by a certified authority), we give to each key the topology-based property. Therefore, the Topology Vector is the tool used by TAKS to distribute the knowledge of the planned/certified topology in the network: each node knows his local and certified planned topology knowing his Topology Vectors.

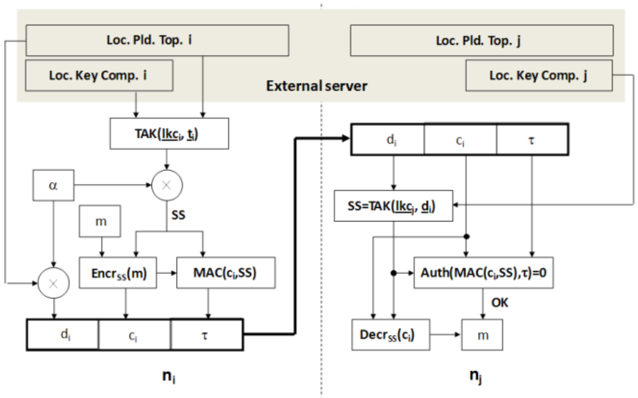

Fig. 4 General TAKS Scheme

The scheme defines that, if a node wants to communicate with another one, it has to generate a random value α and to build a message as concatenation of (Fig. 4):

- the cipher text (c) produced by the symmetric encryption algorithm Encr() with key equals to the product of the random alpha and TAK() applied to his Local Key Component and the destination Topology Vector;
- the deciphering information (d) computed as product between the random alpha and the sender Topology Vector changed of sign;
- the message authentication code (τ) associated to the cipher text using the cryptographic hash function MAC() with key equals to the one used to produce the cipher text.

From the receiver point of view the cipher text can be decrypted using the symmetric decryption algorithm Decr() with TAK() applied to his Local Key Component and the deciphering information vector as key. The cipher text can be considered correctly decrypted if and only if actual computed MAC equals to the tag provided in the message (τ), i.e. if Auth() function equals zero. Therefore, if Auth() function returns zero, encryption key equals to the decryption one and the message is handled, otherwise it is discarded.

B. *Implications on stack layers*

This sub-section deals with implications of adopting TAKS in proposed architecture. The usage of TAKS requires different considerations if it is applied on the top of *Network Layer* or on the top of IEEE 802.15.4 *MAC Layer* due to different API provided by the layers.

Typically, API provided by *Network Layer* is similar to interface provided by *Physical Layer*: by an application point of view, only the procedures needed to send and receive messages are of interest. Therefore, considerations that we have made in [10] and reported in previous sub-section are completely applicable.

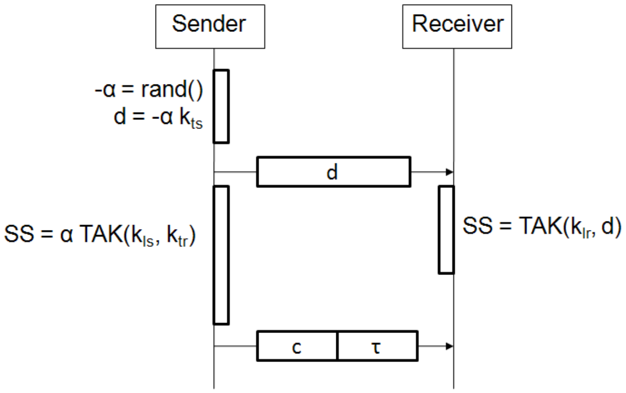

Fig. 5 TAKS Scheme for IEEE 802.15.4

If we apply TAKS facility on the top of IEEE 802.15.4 *MAC Layer*, we need to do some adaptations. In the following, we consider as a reference scenario a network composed of two nodes. With reference to IEEE 802.15.4 *MAC Layer*, this means that we have a *device* and a *coordinator*. To simplify, we suppose to transmit from the device to the coordinator (we have the same conditions in the opposite case but the communications are indirect). Primitives provided by the standard implies the splitting of the transmission procedure between the key agreement and truly transmission phases. When a secured packet is received, the receiver IEEE 802.15.4 *MAC Layer* needs to be already set with security parameters. In this manner, it can correctly notify higher layer. However, since original TAKS scheme assigns key generation phase contextually to the packet reception, it is not applicable without modifications.

To solve the problem, a new scheme is proposed. In Fig. 5, the scheme is presented, where k_{ts} and k_{tr} represent Topology Vectors of sender and receiver respectively and k_{ls} and k_{lr} are their Local Key Components. The new scheme requires two phases: the former is the transmission of the deciphering information (d), that is necessary to the receiver to calculate key as in "normal" scenario; the latter is the transmission of cipher text (c) and authentication tag (τ). In this scheme, the actual receiver can correctly decrypt the message if and only if, at the reception of message, it has already computed and set the key. This happens if and only if the time elapsed between two phases at the sender is greater than the one at the receiver. This is always true because time to compute αTAK(k_{ls}, k_{tr}) is greater than time to compute TAK(k_{lr}, d), due to the execution of two extra multiplication. However, also the backoff procedure to access the medium could contribute to separate the two phases and therefore, to facilitate the correctness condition checking.

VII. Conclusions and Future Works

The middleware proposed in this paper aims to provide advanced security services to applications that rely on a real-world WSN protocol stack where lower layers are compliant to the IEEE 802.15.4 standard. Our approach makes this middleware suitable for a wide set of applications since IEEE 802.15.4 is the de-facto standard used for realistic WSN deployment. Our work proposes TAKS adaption to be implemented on the top of the IEEE 802.15.4 MAC Layer (i.e. topology-based key agreement protocol for 802.15.4 networks) and a set of design choices that we have identified to implement WIDS while guaranteeing availability of the network.

While actual implementation of the proposed stack is in an advanced stage, near future objectives consist in performance assessment through field trials even in large test-beds and implementation of models for a larger set of threats.

Acknowledgment

The research leading to these results has received funding from the *European Union Seventh Framework Programme* [FP7/2007-2013] under grant agreements n. 257462 *HYCON2 Network of excellence* and n. 240555 *ERC SG VISION*. Moreover, it has been motivated and supported by the *ESF-COST Action IntelliCIS* (Prof. Fortunato Santucci has participated to this Action). The development of the middleware platform also fits in the frame of the projects *RICOSTRUIRE* and *SMILING* supported by the *Ministry of Economic Development* to enhance technology transfer in the *RIDITT* framework.

References

[1] M. Pugliese, L. Pomante, F. Santucci, "Agent-based Scalable Design of a Cross-Layer Security Framework for Wireless Sensor Networks Monitoring Applications", Proceedings of the International Workshop on Scalable Ad Hoc and Sensor Networks (SASN2009), Saint Petersburg, 2009.

[2] M. Pugliese, L. Pomante, F. Santucci, "Secure Platform over Wireless Sensor Networks", INTECH Publishers. 2012. ISBN 978-953-51-0218-2.

[3] L. Pomante, M. Pugliese, S. Marchesani, F. Santucci, "WINSOME: A Middleware Platform for the Provision of Secure Monitoring Services over Wireless Sensor Networks", 9th International Wireless Communications & Mobile Computing Conference (IWCMC 2013), Cagliari, 2013.

[4] Agilla Home Page, http://mobilab.wustl.edu/projects/agilla/

[5] IEEE 802.15.4-2006 standard, http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf

[6] S. A. Camtepe, B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey", Techical Report TR-05-07, Troy, 2005.

[7] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks", IEEE Symposium on Research in Security and Privacy. 2003.

[8] B. Lai, S. Kim, I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks", IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES). 2002.

[9] M. Pugliese, F. Santucci, "Pair-wise Network Topology Authenticated Hybrid Cryptographic Keys for Wireless Sensor Networks using Vector Algebra", Proceedings 4th IEEE Intern. Workshop on Wireless Sensor Networks Security (WSNS08), Atlanta, 2008

[10] S. Marchesani, L. Pomante, M. Pugliese, F. Santucci, "Definition and Development of a Topology-based Cryptographic Scheme for Wireless Sensor Networks", 4th International Conference on Sensor Systems and Software (S-Cube 2013), Lucca, 2013.

[11] S. Kaplantzis, "Classification techniques for network intrusion detection" , Technical Report, Monash University, 2004.

[12] D.E. Denning, "An Intrusion-Detection Model", IEEE Transactions on Software Engineering, vol. SE-13, no. 2, February 1987.

[13] A. H. Farooqi, F. A. Khan, "Intrusion Detection Systems for Wireless Sensor Networks: A Survey", Communication and Networking, pp. 234-241, 2009.

[14] R. Roman, J. Zhou, J., Lopez, “Applying Intrusion Detection Systems to WSNs”, IEEE Consumer Communications and Networking Conference, vol. 1, pp. 640–644 (2006)

[15] I. Krontiris, T. Dimitriou, “Towards Intrusion Detection in Wireless Sensor Networks”, Proceedings of 13th European Wireless Conference, Paris, France (2007)

[16] C.E. Loo, M.Y. Ng, C. Leckie, M. Palaniswami, “Intrusion Detection for Routing Attacks in Sensor Networks”, International Journal of Distributed Sensor Networks 2(4), 313–332 (2006)

[17] D. Hankerson, A. Menezes, S. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer-Verlag, New York, 2004. ISBN 0-387-95273-X.

[18] S. Gupta, R. Zheng, A,M.K. Cheng, “An Anomaly Detection System for Wireless Sensor Networks”, Proceedings of IEEE International Conference on Mobile Ad hoc and Sensor Systems, pp. 1–9 (2007)

[19] Q. Zhang, T. Yu, P. Ning, “A Framework for Identifying Compromised Nodes in WSNs”, ACM Transaction Information System Security 11(12) (2008)

[20] Y. Ephraim, N. Merhav, “Hidden Markov Processes”, IEEE Trans. Information Theory, vol. 48, no. 6, 2002

[21] G. Forney, “The Viterbi Algorithm,” Proceedings IEEE, vol. 61, pp. 263–278, 1973

[22] A. Gill, “Introduction to the Theory of Finite-State Machines”, McGraw-Hill. 1962.

[23] W. Xu, K. Ma, W. Trappe, Y. Zhang, “Jamming sensor networks: attack and defense strategies”, IEEE Network, vol.20, no.3, 2006, pp.41-47.

[24] Y.W. Law, P. Hartel, J. den Hartog and P. Havinga, “Link-layer jamming attacks on S-MAC”, Proceedings of IEEE WSN’05, 2005, pp.217-225.

[25] R. Sokullu, I. Korkmaz, O. Dagdeviren, A. Mitseva, N. R. Prasad, “An Investigation on IEEE 802.15.4 MAC Layer Attacks”, Proceedings of the International Symposium on Wireless Personal Media Communications (WPMC'07), Jaipur, India, 2007.

[26] S.S. Jung, M. Valero, A. Bourgeois, and R. Beyah, “Attacking Beacon-Enabled 802.15.4 Networks”, Proceedings SECURECOMM, 2010, pp.253-271.

[27] C. Karlof , D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, Ad Hoc Networks, vol. 1, pp.293 -315 2003

[28] I. F. Akyildiz IF, W. Su,Y. Sankarasubramaniam, E. Cayirci, “A survey on sensor networks”, IEEE Communications Magazine, 2002.

[29] C. Mascolo, S. Hailes, “Survey of middleware for networked embedded systems”, Technical Report for Project: Reconfigurable Ubiquitous Networked Embedded Systems,University College London, 2005.

[30] L.H. Freitas, K.A. Bispo, N.S. Rosa and P.R.F. Cunha, “SM-Sens: Security middleware for Wireless Sensor Networks”, Proceedings of the Information Infrastructure Symposium, 2009.

[31] R. Daidone, G. Dini, M. Tiloca, “STaR: a Reconfigurable and Transparent middleware for WSNs security”, Proceedings of the 2nd International Conference on Sensor Networks (SENSORNETS 2013), 2013.

[32] P. Chapin, C. Skalka, “SpartanRPC: Secure WSN middleware for cooperating domains”, Proceedings of the Seventh IEEE International Conference on Mobile Ad-hoc and Sensor Systems. 2010.

[33] C. Vairo, M. Albano and S. Chessa, “A secure middleware for wireless sensor networks”, Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services. 2008.

[34] ZigBee Alliance: ZigBee Document 053474r17, ZigBee Specification. ZigBee Alliance (January 2008)

[35] M. Valero, S. S. Jungy, A. S. Uluagacy, Y. Li, R. Beyahy, “Di-Sec: A Distributed Security Framework for Heterogeneous Wireless Sensor Networks”, Proceedings of the IEEE INFOCOM Conference.2012.

[36] S. Hadim, N. Mohamed, “Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks”, IEEE Distrib. Syst. Online 7, 3. 2006.

[37] M. Wang, J. Cao, J. Li, S. K. Das, “Middleware for Wireless Sensor Networks: A Survey”, J. Comput. Sci. Technol. 2008.

[38] A. Cunha, M. Alves, A. Koubaa, “Implementation Details of the Time Division Frame Scheduling Approach for Zigbee Cluster-Tree Networks”, IPP-HURRAY Technical Report, HURRAY-TR-070102, Jan 2007

[39] W. Xu, K. Ma, W. Trappe and Y. Zhang, ”Jamming sensor networks: attack and defense strategies”, IEEE Network, vol.20, no.3, 2006.

[40] Y. Xiao, S. Sethi, H.-H. Chen and B. Sun, ”Security services and enhancements in the IEEE 802.15.4 wireless sensor networks”, in Proceedings of IEEE GLOBECOM’05, vol.3, 2005.