Securing Smart Places: A Power-Aware Infrastructure

Tiago Diogo
Instituto Superior Técnico
Av. Rovisco Pais, 1
1049-001 Lisboa, Portugal
tiago.diogo@tecnico.ulisboa.pt

Miguel Pardal
Inesc ID
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
miguel.pardal@tecnico.ulisboa.pt

ABSTRACT

The Internet of Things (IoT) and its vision of connecting every device to one another presents an opportunity to create large information sharing networks. However, intruders can take advantage of the IoT devices constrained nature to disrupt these networks and launch a wide range of attacks on its nodes. In our work we address this issue from a power-aware perspective, trying to find the best relation between security and power consumption. To achieve this objective we do a thoroughly analysis of the existing protocols, attacks and mitigation strategies, combining that information into our proposed smart places network management system. Furthermore, energy consumption profiling was performed to endow future users with the knowledge of what kind of physical resources to deploy, based on the desired network security characteristics.

CCS Concepts

•**Computer systems organization** → **Sensor networks;**
•**Security and privacy** → *Mobile and wireless security;*

Keywords

Internet of Things; Power-Aware Security; Secure Bootstrapping; CoAP; 6LoWPAN; RPL; IEEE 802.15.4

1. INTRODUCTION

The Internet of Things can be seen as a web of interconnected devices that go from everyday wearable objects into fully deployed sensor networks. Despite the huge variety and characteristics of these devices, one thing that they all have in common is the constrained nature that they are built upon. In order to enable the massive deployment to be expected in the near future, IoT devices must be accessible and affordable, capable of operating under lossy wireless networks while being battery powered. This poses a challenge to current Internet protocols since the assumptions regarding the devices' capabilities and objectives do not hold true.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACSAC '16 Los Angeles, California USA

© 2016 ACM. ISBN 123-4567-24-567/08/06...$15.00

DOI: 10.475/123_4

To allow the IoT vision to come forward, several new protocols have been developed across the OSI layers, each addressing and tackling the challenges involved in trying to keep the quality and assurances of stronger, more expensive protocols, on constrained systems. After being thoroughly analysed, these protocols have been selected and grouped in a power-efficient stack, establishing a base line for power consumption. Additionally, major attention has been given to information security because for both corporations and individuals, the interconnection of the devices around us can provide information about our choices and whereabouts, therefore leaking corporate information or simply reducing our individual privacy [14]. Thus, the focus moved towards adding mechanism to ensure authentication, confidentiality and integrity of the transmitted information by securing the communication channel. This increased the infrastructure complexity and created the need for a management system capable of storing and flashing network credentials onto new nodes. In order to understand the cost of adding these mechanisms, additional experiments have been performed so that the added power consumption can be measured, profiled and documented, enabling the finding of the best parameters and requirements for a desired level of security.

2. SECURING SMART PLACES

2.1 Protocol Analysis and Selection

There are many alternatives and some proposed standards when it comes to choosing a protocol stack for IoT communications so a thorough analysis of the existing solutions is necessary to unveil the strong and weak points of each candidate. The following study is based on Al-Fuhaga survey[1] describing IoT enabling technologies, protocols and applications.

2.1.1 Data Link and Physical Layer

The first requirement for the physical layer of the IoT is the use of wireless radios. These should aim for simplicity, low-power and low-cost communications. While wireless communication is widespread and can be found from homes to airports, the type of radio commonly used, known as Wi-Fi, uses a high amount of power causing concerns for battery life. In the next paragraphs, an overview of Wi-Fi (IEEE 802.11) is given with the objective of comparing it with the IEEE 802.15.4, a protocol that aims to address these issues.

IEEE 802.11

IEEE 802.11 [5] is a set of standards for Wireless Local Area Networks (WLAN) communications. They are the basis for the so called Wi-Fi. IEEE 802.11 is concerned with high speed, long ranges, message forwarding and high data throughput. These concerns directly clash with the IoT objectives and account for the added power consumption of this protocol.

IEEE 802.15.4

IEEE 802.15.4 [6] on the other hand was created for Low-Rate Wireless Private Area Networks (LR-WPAN) and its specifications focus on low power consumption, low data rate, low cost and high message throughput make it a strong candidate for IoT applications. The IEEE 802.15.4 standard supports two types of network nodes, the Full Function Device (FFD) that acts as coordinator or normal node, and the Reduced Function Device (RFD) that is very simple, with very constrained resources and can only communicate with coordinators. The coordinators are responsible for controlling and maintaining the network. FFD are capable of storing a routing table in their memory and can implement a full Medium Access Control (MAC). IEEE 802.15.4 supports star, peer-to-peer (mesh) and cluster-tree topologies. Regarding performance, it would be unfair to directly compare the two, since IEEE 802.11 transmission power and receiver sensitivity are much greater than 802.15.4. Even if we limit both to a low power level, IEEE 802.11 still outperforms IEEE 802.15.4 in terms of packet delivery ratio, throughput, latency, jitter and average energy consumption. However this comes at the cost of a far lower transmission range [8]. We can conclude that for typical LR-WPAN network requirements, IEEE 802.15.4 is better designed to address the constrained environment issues, while IEEE 802.11 would still be a suitable option if a short transmission range is not a problem.

2.1.2 Network Layer

6LoWPAN

The IoT vision and its massive deployment can only be achieved through the use of IPv6. However, physical layers more suitable for communication over constrained networks pose some limitations to the use of the IPv6 messages. For example, the limited packet size in IEEE 802.15.4 based networks. To tackle these issues, the Internet Engineering Task Force (IETF) IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [12] working group developed a standard based on header compression to reduce the transmission overhead, fragmentation to meet the IPv6 Maximum Transmission Unit (MTU) requirements and forwarding to link-layer to support multi-hop delivery [4]. 6LoWPAN is able to remove a major share of IPv6 overheads, being able to compress its headers to two bytes, therefore allowing small IPv6 datagrams to be sent over IEEE 802.15.4 networks.

RPL

With the use of 6LoWPAN, upper layer routing protocols can now use the IPv6 addressing scheme. Given the possible frequent topology changes associated with the radio-link instability, successful solutions must take these requirements into account on their specification. Routing Protocol for Low-Power and Lossy Networks (RPL) [16] can support a wide variety of link-layers and is prepared for devices with very limited resources. It is able to build up network routes, distribute routing knowledge among nodes and adapt the topology in a very efficient way. More in depth, RPL creates a Destination Oriented Directed Acyclic Graph (DODAG) between the 6LoWPAN network nodes (Figure 1) that supports unidirectional traffic towards the DODAG root and bidirectional traffic between devices. Each node has a rank that indicates its position relative to other nodes and with respect to the root. This rank is used to create optimized network paths. In order to allow packets to propagate downwards in the topology, either source routing or stateful routing tables are used (More Information on this two types of routing are given in sections 2.2.1 and 2.2.2). For both modes, the DODAG root always maintains a complete list of the network nodes. RPL provides a set of control messages in order to exchange routing graph information. DODAG Information Objects (DIO) are used to advertise information needed to build the DODAG. Destination Advertisement Objects (DAO) are used to advertise information so that downwards traffic can go through the nodes towards the leafs. Nodes may also resort to DODAG Information Solicitation (DIS) messages to request graph information from neighbour nodes. Finally, RPL has a built in topology repair mechanism that acts in the case of a routing topology failure, link failure or node failure. In case the topology needs to be rebuilt, a link layer metric is used to calculate the new route. The new path is considered fit for work if the link layer acknowledgements are received on it.

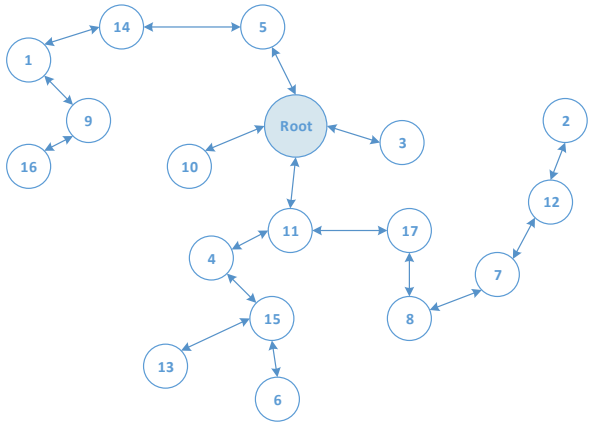

Figure 1: A Sample RPL DODAG.

2.1.3 Application Layer

Hypertext Transfer Protocol (HTTP)

HTTP is an application level protocol that uses a request-response model and is the foundation of data communication on the World Wide Web (WWW) It is primarily designed to run over Transmission Control Protocol (TCP) which is a problem in lossy and constrained environments due to the delivery assurances and congestion control algorithms it em-

ploys. Besides, HTTP is verbose, text-based, and not suited for compact message exchanges. Moreover, the header size required for a message exchange can leave too few payload space in constrained networks like the IEEE 802.15.4-based networks where the MTU size of the protocol is 127 bytes. These protocol specifications would not raise any issues in standard WWW communications, but when it comes to constrained environments it is clear that the protocol is not adequate to the necessities of IoT devices and networks.

Constrained Application Protocol (CoAP)

CoAP [13] is a document transfer protocol based on REpresentational State Transfer (REST) on top of HTTP functionalities. CoAP objective is to enable tiny constrained devices to use RESTful interactions, where clients and servers expose and consume web services using Universal Resource Identifiers (URIs) together with HTTP Get, Post, Put and Delete methods. Unlike REST, CoAP runs over User Datagram Protocol (UDP) instead of TCP which makes it suitable for full IP networking in small micro-controllers. Retries and reordering are implemented at the application stack using a messaging sub-layer that detects duplicated messages and provides reliable communication using different types of messages. Confirmable messages must be acknowledged by the receiver, nonconfirmable follow the fire-and-forget model. Despite being a lightweight protocol, CoAP still provides important features:

- Resource Observation - CoAP can extend the HTTP request model with the ability to observe a resource therefore monitoring resources of interest using a publish/subscribe mechanism;
- Resource Discovery - CoAP servers provide a list of resources using well-known URIs that allow clients to discover what resources are provided and their types;
- Interoperability - since CoAP is based on the REST architecture, a simple proxy enables CoAP to easily interoperate with HTTP.

A study that compared CoAP and HTTP using mobile networks concluded that there is no scenario where CoAP would consume more resources than HTTP [11].

Table 1: Protocol Stack Comparison Overview

Layer	Web	IoT
Application	HTTP	CoAP
Transport	TCP	UDP
Network	IPv6	6LoWPAN
Data-Link/Physical	802.11	802.15.4

2.2 Attack Analysis, Detection and Mitigation

Exploitation of existing solutions in the forms of malicious attacks can be found at all the studied OSI layers. They can go from a physical intruder replacing some node on a sensor field to the well-known Denial of Service (DoS) at the application layer. However, given the characteristics of the devices and networks used in IoT combined with the power consumption focus of this work, a specific kind of attacks performed at the network layer is of special interest and importance: battery depletion attacks, also known as, "vampire" attacks.

Battery depletion attacks aim at draining the battery, "life", of the network devices, working over time to entirely disable a network, hence being called "vampire" attacks. These attacks do not focus on flooding the network with many packages, instead they drain the node's life by delaying the packets transmission. Many of the existing attacks are not protocol specific [15], while others target specific protocols and implementations [10]. The following attacks aim at giving an overview of the existing attack possibilities on different routing solutions as well as existing mitigation strategies.

2.2.1 *Stateless Protocols*

In systems that use this type of routing protocols, the source node specifies the entire route to the destination in the packet header. This means that intermediaries do not make decisions regarding the next hop, they only forward to the next node as specified in the original path therefore reducing the amount of computation performed and used energy. However, the source node must ensure that the route is valid at the time of sending and that the neighbour relations among the devices allow the specified forwarding path. Using this transmission scheme, a malicious device can specify paths through the network that are far from optimal, wasting energy at the intermediate nodes who follow the included malicious source route. The Carousel Attack is an example of such attacks.

Carousel Attack

The objective of this attack is to send a packet along a route composed as a series of loops. This way, a single node may forward the malicious packet several times increasing the total energy consumption by a factor of the number of loops the attacker has introduced on the packet header path. It targets source routing protocols by exploiting the limited verification of the packet headers at the intermediary nodes. Figure 2 shows an example where a vampire node created a path composed of circles around the network when it could have exited after the first hop through the D node.

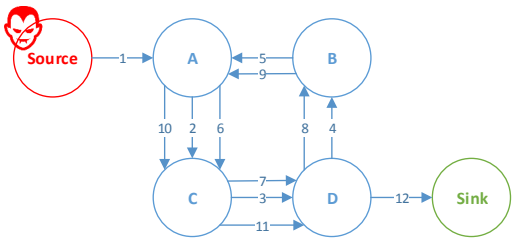

Figure 2: Carousel Attack.

Existing mitigation strategies rely on checking the source route for loops on intermediary nodes, either selecting an appropriate route for the packet or simply dropping it.

2.2.2 *Stateful Protocols*

In systems that use this type of routing protocols, network nodes are aware of the network topology and its state, being able to make local decisions on the node to whom they will forward the packet. The effect of the Vampires on this type of routing is limited since the route is built dynamically from many independent forwarding decisions. However, attackers can still cause damage by forcing packet forwarding through nodes that would not be on the optimal path, for example, by forwarding the packet back to the source. The Directional Antenna Attack is an example of such attacks.

Directional Antenna Attack

In this attack, the attacker takes the role of an intermediary and not the source of a packet. If the attacker has the resources to use a directional antenna, it can deposit a packet on arbitrary parts of the network while also forwarding the packet locally. This causes nodes that were not on the optimal path to also consume energy by forwarding a packet they would not normally receive, therefore increasing the total energy consumption by a factor of the directions the attacker can position the antenna and the distance between the receiver and the sink. Figure 4 shows an example where a "vampire" intermediary deposited a node on a distant location of the network, causing the packet to follow two different routes towards its destination

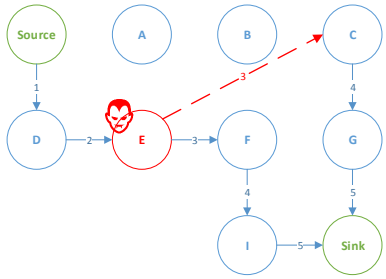

Figure 3: Directional Antenna Attack.

A mitigation strategy could be to analyse the route paths of a given packet that reached the sink more than once. The last node identifier to appear duplicated before the path started to diverge would be one who then directed the packet to multiple regions, therefore revealing the attacker.

2.2.3 *RPL Specific Attacks*

Selective Forwarding Attack

In a selective forwarding attack, a malicious node can launch a DoS attack by selectively forwarding packets. Its main goal is to disrupt routing paths but can be used to filter any protocol. Since RPL has built in topology repair mechanisms, a full packet filtering would trigger a healing phase and leave the malicious node out of the topology. For sustainability, an attacker could let the RPL control messages pass by and drop the remaining packets. Depending on the routing scheme being used (source routing or stateful tables) the source could first verify path availability or each node could dynamically decide to forward the packet through another path with similar quality. In any case, a good approach would be to report those failures to the underlying RPL system in order to trigger a preventive healing and improve the route quality.

Hello Flooding Attack

The Hello in the name of this attack comes from the initial message a node sends when joining a network. By broadcasting this message with a strong signal power, an attacker can try to introduce himself as neighbour to many nodes of the network, or at least force a large portion of the network to spend energy starting the message exchange for node insertion. A simple solution for this attack would be to test the bi-directionality of the link. If no acknowledgement is received, the path is discarded. Another approach, if the geographical locations of the nodes are known, would be to discard every hello message coming from a location beyond the transmission capabilities of ordinary nodes.

2.2.4 *Protocol Independent Attacks*

The last addressed category is not dependant on network topologies or protocol messages. It focuses on attacks that can be performed regardless of the used protocol and whose goal is to obtain information about a network device. With that information an attacker can, for example, try to include himself in the network as a legitimate device or spoof his identity to forward traffic towards him. The Clone and Sybil Attacks are examples of these attacks.

Clone ID and Sybil Attack

As the name suggests, in a clone ID attack, the attacker steals the identity of a legitimate network node by copying the information of that node onto another node. This way the attacker can gain access to the traffic that was destined to the legitimate node, prevent packets to reach their intended destination and can even influence voting schemes. The Sybil attack is similar to the Clone ID, with the difference that the attacker uses several stolen identities on the same physical node. This way, large parts of a network can be taken over without the need to deploy several physical nodes. Figure 6 shows an example of a clone ID attack where the cloned attacker received the packet that was originally destined to the legitimate node.

Proposed mitigation strategies for this type of attacks consist on keeping track of the number of instances of each identity. By using the node neighbours, either a centralized or distributed approach could be used to detect duplicate entries.

2.3 Secure Bootstrapping

The term bootstrapping is applied to the process in which a new device is connected to an existing network. To achieve a secure bootstrapping, a unique identity and security parameters are associated with the device during this phase. There are several ways to carry out the initial setup, either via a physical interface or wirelessly. In the case of wireless bootstrapping, attention must be given to eavesdropping so that the secure credentials cannot be intercepted.
Since many of the studied attacks are to be performed by a malicious intruder capable of interacting with the network, if we could assure a secure bootstrapping, meaning that the new node would be authenticated before becoming an ac-

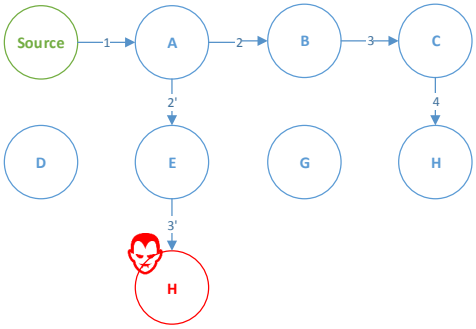

Figure 4: Clone ID Attack.

tive member of the network, a large portion of those attacks could no longer be performed. The following bootstrapping techniques were summarized in [3] and aim at providing secure bootstrapping for IoT devices.

2.3.1 *Token-Based*

In token-based distribution, device specific security credentials are generated and written to a token. That token can range from memory sticks or flash cards to Radio Frequency Identification (RFID) tags or smartcards. Is has the advantage that this initial credential generation can be performed on a physically controlled environment and only later, on the commissioning phase, is the token plugged into the device. After the successful insertion of the security credentials, the token can be removed and collected back into the secure environment. This process can be considered of high security since the credentials are generated on a closed environment and are transmitted through a physical link. To further increase the security level, a password could be used to encrypt the credentials, however, that would require the device to have some kind of interface that would allow to input the password. In the case of a large number of devices, this approach would be unsuitable due to the management effort of manually deploying the tokens to the devices [3].

2.3.2 *Identifier-Based Access Control List*

With an identifier based Access Control List (ACL), new devices are allowed or denied access to the network based on their unique ID. A commonly used identifier is the MAC address. This has some major drawbacks in security since, firstly, it provides no assurances on the first time the device connects to the network. An attacker can easily intercept the first messages and get access to the device information. And secondly, after the bootstrapping phase, MAC addresses can be spoofed by an attacker, allowing him access to the network by bypassing the ACL with the identifier of a legitimate node.

2.3.3 *One-Time-Passwords*

The use of one-time-passwords enhances the manual input of credentials on the device to be bootstrapped. The person responsible for the deployment of the new node should receive through a secure channel an one-time-password, that would then be used to authenticate the node, by authenticating its locally generated key material. This material can be either a certificate request to a Certificate Authority (CA) or a locally generated public/private key pair. The achieved security level is proportional to the security of the channel used to obtain the one time password, but assuming that channel is secure, so is this method. The drawback is that it forces devices to possess some kind of interface to insert the one time password.

2.3.4 *Manufacturer Installed Credentials*

So far, excluding the identifier based access control list, the intent of the studied techniques is to supply to the new device the security credentials needed to obtain access to the network, or at least provide an authentication method that allows fetching those credentials. In manufacturer installed credentials, those security credentials are deployed during the manufacturing process of the device vendor. Those credentials are typically a public/private key pair certificate bound to the identifier of the device. This certificate can be integrated into the initial loading of the firmware or stored in a separate integrated circuit designed for credential storing. In the second case, this method's security can be considered very high since those integrated circuits assure that the private key cannot be read from memory. This way, the new device comes shipped with the necessary security credentials not only for the bootstrapping phase but also for the normal operation phase since it does not need to fetch any additional credentials. The effort is on the root or management station that needs to import the vendor CA certificates to assure the new device credentials are trustworthy. Also the production costs increase, implying an increased device cost.

2.4 Existing Solutions

Secure bootstrapping and network admission solutions have already been proposed in past literature. However, the development and optimization of application layer protocols as well as network layer routing schemes allows for new approaches and solutions that can now fit the nature of IoT devices. Bergman et al. [2] proposed a three-phase secure bootstrapping technique for nodes in a CoAP network. Firstly the joining node broadcasts a request for a CoAP Service Discovery Server (CSDS). This server, once contacted by a new node takes the responsibility of key distribution. Then the system goes under a vulnerable phase where the secret is transmitted from the CSDS onto the new device. The author's proposes a short audible or visual feedback to the human installer when the secret is received and assumes that potential eavesdroppers can not intercept this transmission. Finally, this secret is used to setup the Datagram Transport Layer Security (DTLS) connection. This approach has major security drawbacks. On the secret transmission phase, so the authors propose limiting the radio power to a low level and disable data forwarding beyond the local network segment, but these techniques cannot assure that an attacker won't be able to intercept the transmission.
Oliveira et al. [9] proposed an admission control solution for 6LoWPAN networks based on administrative approval. Each joining node would broadcast its presence to the network, and that broadcast would be received by the administrator in the management server. Then, the administrator would grant access to that new device based on its address,

and that information would be transmitted to all the devices in the network. After this phase, the device would be allowed communication as a regular member of the network by its neighbours. This approach has the advantage of requiring no previous setup on the device before operation but is vulnerable to the attacks previously mentioned in identifier based ACL. The authors state that work still needs to be performed in order to validate the sensor identity and leave as possibility the pre-instalment of keys on the device.

2.5 Proposed Infrastructure

The IoT principle of connecting every device do one another in an automated way can create networks beneficial to a wide spectrum of applications, ranging from home environments to large enterprises. However these are domains that have different requirements. A home application should be easy to setup without complex configurations. An enterprise solution can benefit from additional administrative configurations as long as the deployment of the network nodes is done quickly due to their potential large number. In order to demonstrate the capacities and applicability domain of our system, we will use a Smart University Campus scenario since it is out believe that it can effectively demonstrate the needs targeted by our work. In the following sections we will apply the information gathered in terms of protocols, attacks and mitigation strategies do define our objectives and requirements. Then, a model of a university campus with our proposed power-efficient network architecture will be presented and their components role explained.

2.5.1 Objectives and Requirements

A major concern amongst IoT application is the communication model. It is out goal that the entire network is power-aware, using the minimum energy possible. Also, it is very important that the deployment of new nodes and system maintenance can be performed by regular staff members without knowledge of the inner workings of the system. This creates usability challenges and requires simple and automated interfaces and bootstrapping processes. Additionally, the following set of requirements is critical in order to allow secure communications to take place.

- Confidentiality: Without confidential message transmission, packets would flow in the network in plain text. Attackers could sniff the packets in order to obtain information, and depending on the application, this could be a security breach. Even if there is no critical data being sent, privacy is still compromised.
- Integrity: Assuring message integrity means that the message was not modified between the source and its destination. Without integrity we could not rely on the received data since it could have been, intentionally or not, modified on the fly, and be providing the system wrong information.
- Authentication: The studied type of networks relies on hop-to-hop communication, meaning several nodes will take place in forwarding a packet. If they are not authenticated they could perform a wide range of attacks and disrupt the network.

2.5.2 System Architecture

As previously stated, we will use a Smart Campus scenario. Being aware of the technological improvements on sensor networks and building management technologies, the Instituto Superior Técnico (IST) administration decided to improve the monitoring the overall conditions of the buildings and inside environments in order to better preserve its assets. To cope with the new requirements, we propose a solution for the monitoring of the campus sections by deploying a wireless sensor network on each building, connected to a central management station operated by the available staff. The scenario will be based on the IST campus model. An overview of the system and its components over the IST blueprints can be found in Figure 7. Regarding each individual component:

- Numeric Nodes: Represent the network sensor nodes, the most constrained element of the network. They cooperate to build the topology and route messages hop-by-hop until the root is reached. These are fully equipped with the previously presented energy efficient protocol stack
- Alphabetical Nodes: Represent the root node of each section network topology. They are equipped with the same stack of the numbered nodes but are more powerful, preferentially not battery powered and act as the bridge between the constrained 6LoWPAN environment and the central management station. These nodes must be more powerful than the numeric ones so that they can process all the requests between a group of sensors and the management station. Also, although the numeric nodes use low-power wireless radios, the alphabetical nodes must be capable of interfacing with more power hungry radios and protocols therefore requiring more resources. This differentiation allows numeric nodes (the large portion of the network devices) to keep their very constrained nature, consuming less energy, an still be able to communicate with external devices.
- Management Station: A black box model of the core components of the system. Each building reports to the central station and the staff monitors the status through it. A white box model will be shown in the following sections.
- Client: The system's clients can be any user with access credentials, but mostly the staff members. They can access the management station either from within the local network or from outside through the Internet.

Central Management Station.

The central management station is divided in five main components. A white box schematic of the core components and interactions can be found in Figure 8. Regarding each individual component:

- Key Store: This component is responsible for storing the shared network key for the RPL protocol;
- Bootstrapper: The bootstrapper acts as the interface between the management station and the network devices. It generates the device identifier and writes it together with the shared network key into the new device;

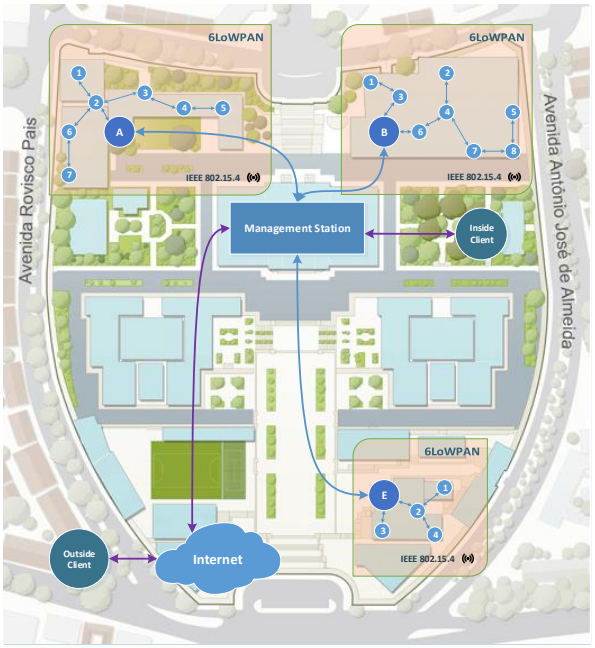

Figure 5: Global System Architecture

- CoAP Client Observer: The one and only client in the network. Instead of the user directly requesting the sensor readings, the client will observe each resource and be notified of the new value. Each time it receives an update, it stores the information on the Data Server for the clients to use;
- Data Server: A database with mappings of each node to the most up to date value reported. It's updated by the client observer and used on demand by the clients;
- Proxy: Responsible for bridging requests coming from the Internet to the Data Server. Responsible for authenticating the external clients and providing access to the Data Server information.

Although each user could access the system through a CoAP terminal and request the most up-to-date readings from the sensor nodes, this approach would cause unnecessary overheads in the system. Since many clients can connect from different locations, many requests would be performed to the sensor nodes for the same information, this would mean additional memory usage in the physical devices, and more requests to the already constrained battery operated network. With the single client approach acting as an observer, only one message needs to go through the network for each new reading.

3. EVALUATION

In this section we measure and profile the resources required for the system operation. This ranges from the memory required for the nodes operative system and protocol stack, to power consumptions and battery replacement expectations. Each following subsection both presents the collected data and explains the process and technologies involved in obtaining it.

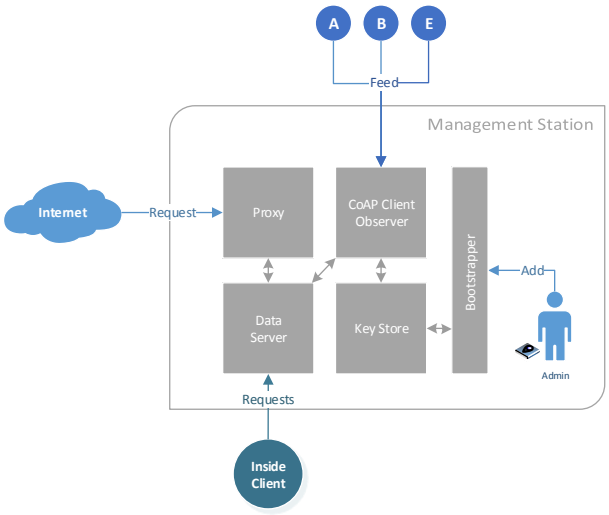

Figure 6: Central Management Station

Table 2: Memory Occupation

Security Mechanism	Flash(KB)	RAM(KB)
No-Sec	59.56	13.54
LLSec	61.36	13.80
DTLS	84.38	15.66
LLSec and DTLS	86.18	15.91

3.1 Hardware Requirements

In order to select hardware capable of supporting the stack of protocols used for communication it is necessary to measure the firmware size. For that task, the *msp430-size*[1] tool was used. This tool analyses the firmware file and outputs the amount of *text*, *data* and *bss*. *Text* corresponds to code and constants, *data* is for initialized variables and *bss* is for uninitialized data (which is initialized with zero in the startup code). The total amount of flash memory can be calculated from the sum of the text and data parameters. The total amount of ram memory can be calculated from the sum of the data and bss parameters. A comparison of the firmware size using the several security mechanisms can be found in Table 2 and a graphical display of the obtained values in Figure 9.

This study, allowed to select a hardware device capable of supporting the size of the protocol stack and additional security measures. Furthermore, the selected hardware should possess a 802.15.4 radio, be capable of being battery powered and provide development tools like sensors and actuators that would be mapped to the application layer endpoints. The board also needs to be compatible with the selected operating system. With all this in consideration, the Zolertia RE-Mote[2] board was selected since it fulfilled all the requirements and also provided an integrated cryptoprocessor while maintaining a low-power operation.

3.2 Power Consumption

[1]http://www.ti.com/tool/msp430-gcc-opensource
[2]http://zolertia.io/product/hardware/re-mote

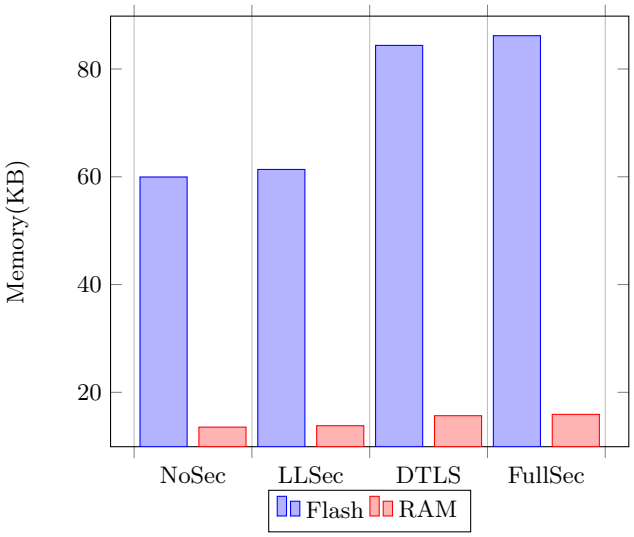

Figure 7: Memory Occupation

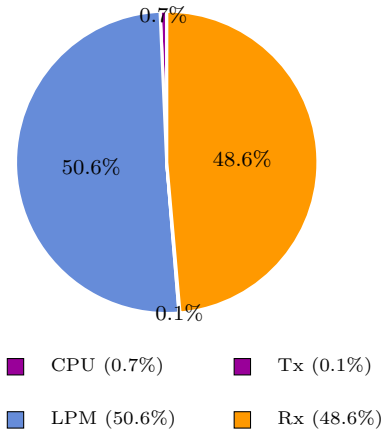

Figure 8: Resource Allocation

Table 3: Power Consumption

RDC Protocol	Voltage(V)	Current(mA)	Power(W)
nullRDC	9.0	30.0	0.27
contikiMAC	9.0	5.2 / 17.0	0.05 / 0.15

The introduction of security mechanisms in low-power networks is necessary and desirable, however due to the low-power characteristics of IoT sensor and actuator networks, a substantial increase in power consumption can disrupt the network by quickly draining the available resources. To this extent, a power consumption analysis was performed on our selected board in order to determine the most suitable battery powering solution.

Firstly, a software based approach was used to determine which hardware core components consumed most resources. An implementation of Energest[**?**] for Contiki was selected since the tool maintains a table with entries for all components, such as CPU, radio transceiver, and LEDs. When a component starts running, a counter starts measuring the clock cycles used for that operation. When the component is turned off, the timer is stopped indicating how much resources were spent during that time.

A sensor node from the selected hardware type, with a baseline of the selected stack without any additional security measurements or optimizations was left working for 60 minutes while producing resource update messages every 60 seconds. The resource division, as obtained from Energest is presented in Figure 10.

We can see from the tool output that nearly half of the node resources is spent on radio activity. This is due to the fact, that out of the box, Contiki-OS uses a Radio Duty Cycling (RDC) protocol that keeps the radio always turned on listening to inbound network transmissions. In the attempt to reduce this radio usage, the ContikiMAC[**?**] RDC protocol, also implemented in Contiki-OS was used. This protocol reduces power consumption by turning the radio off and regularly turning it on to listen to network activity and receive inbound packages. After this phase, the radio is turned off again. With ContikiMAC, nodes can take part in network communication and keep their radios turned off for roughly 99% of the time[**?**]. The drawback are increased latency and decreased throughput due to the cost of keep retransmitting packets until the receiver turns its radio on. A comparison of the measured power consumptions for both protocols can be found in Table 3. The power consumption in Watts(W) has been calculated from the power equation, where I is the current in Amps(A) and V is the voltage in Volts(V):

$$P = IV \tag{1}$$

Due to the enormous variety of batteries in terms of size and capacity, which needs to be selected based on the application necessities and requirements, there is not a fixed amount of days that the system can run without changing or recharging batteries. However, application architects can use these power models to design the battery powering necessities of their systems.

4. CONCLUSIONS

Due to the limitations of IoT devices, achieving secure communications is not an easy task. In order to allow the deployment of battery powered nodes, their communication model must be very efficient and consume the minimum amount of power required for operation. To achieve those requirements we started by analysing the existing protocols across the OSI layers, trying to find the best suited solutions for this type of environments. After a thorough comparison we achieved a working stack of protocols but soon discovered possible breaches and attacks, especially on the network layer. Those attacks were further investigated and catalogued. Given the common principle on the majority of the attacks, the introduction of rogue nodes to the network, we presented some possible solutions based on secure bootstrapping, the secure authentication of new nodes when joining a network.

Once the energy efficient stack, possible attacks and mitigation strategies were defined, we proposed our solution based on a Smart Campus scenario. This solution is focused on providing the joining devices all the secure credentials required for a secure bootstrapping before the deploy on the field, so that when they start the operation phase no

additional credentials need to be fetched, implying that no additional energy is spent on configuration.

Always maintaining a power-aware perspective, the system has been evaluated by measuring its energy consumption with different configurations and core battery usage components. This charting allows future users of the system to decide the type of resources they need to allocate in order to achieve a desired level of security for their application.

As future work, currently out of the scope of this project, memory access protection should be addressed in order to prevent the stealing of secure credentials from deployed devices.

5. REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. *IEEE Communications Surveys & Tutorials*, PP(99):1–1, 2015.

[2] O. Bergmann, S. Gerdes, S. Schafer, F. Junge, and C. Bormann. Secure bootstrapping of nodes in a CoAP network. *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 220–225, 2012.

[3] K. Fischer, J. Geßner, and S. Fries. Secure Identifiers and Initial Credential Bootstrapping for IoT@Work. *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 781–786, 2012.

[4] J. Hui and D. Culler. Extending IP to low-power, wireless personal area networks. *IEEE Internet Computing*, 12(4):37–45, 2008.

[5] IEEE. *IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, volume 2012. 2012.

[6] IEEE Computer Society. *Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, volume 2011. 2011.

[7] F. I. Khan. Wormhole attack prevention mechanism for RPL based LLN network. *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 149–154, 2013.

[8] N. K. N. Kok Seng Ting, Gee Keng Ee, Chee Kyun Ng and B. M. Ali. The Performance Evaluation of IEEE 802 . 11 against. (October):850–855, 2011.

[9] L. M. Oliveira, J. J. Rodrigues, C. Neto, and A. F. de Sousa. Network Admission Control Solution for 6LoWPAN Networks. *Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 472–477, 2013.

[10] P. Pongle and G. Chavan. A survey: Attacks on RPL and 6LoWPAN in IoT. *2015 International Conference on Pervasive Computing (ICPC)*, 00(c):1–6, 2015.

[11] T. Savolainen, N. Javed, and B. Silverajan. Measuring Energy Consumption for RESTful Interactions in 3GPP IoT Nodes. pages 1–8, 2014.

[12] Z. Shelby, S. Chakrabarti, E. Nordmark, C. Systems, C. Bormann, and Ericsson. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). https://tools.ietf.org/html/rfc6775, 2012.

[13] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). https://tools.ietf.org/html/rfc7252, 2014.

[14] A. Ukil, S. Bandyopadhyay, and A. Pal. Privacy for IoT: Involuntary privacy enablement for smart energy systems. *2015 IEEE International Conference on Communications (ICC)*, pages 536–541, 2015.

[15] E. Y. Vasserman and N. Hopper. Vampire attacks: Draining life from wireless ad Hoc sensor networks. *IEEE Transactions on Mobile Computing*, 12(2):318–332, 2013.

[16] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. https://tools.ietf.org/html/rfc6775, 2012.