Power-Aware Security Protocols for the Internet of Things

Tiago Miguel Correia Diogo

Instituto Superior Técnico, Universidade de Lisboa
Avenida Rovisco Pais 1, Lisboa,
tiago.diogo@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 the 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 network management system to be evaluated on a Smart Campus scenario. Furthermore, we will perform energy consumption profiling to endow future users with the knowledge of what kind of physical resources to deploy, based on the desired application security level.

Keywords: Internet of Things, Power-Aware Security, Secure Bootstrapping, CoAP, MQTT, 6LoWPAN, RPL, IEEE 802.15.4

Table of Contents

1 Introduction

The Internet of Things IoT 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,[1] 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.
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. Additionally, security is also a very important due to the fact that the interconnection of the devices around us can provide information about our choices and whereabouts, therefore reducing our privacy [1]. This document will address these issues from a power-aware perspective, meaning that the battery consumption will be of major importance.

1.1 Main Goals

Given the constraints and limitations of IoT devices described in the previous section and the recent protocols, the first objective of this work is to identify a working stack of protocols that takes into account the IoT constraints and focuses on allowing a power-aware communication model.

After the analysis of the existing solutions, a baseline of power consumption will be established. Then, the focus will move towards adding authentication, confidentiality and integrity to the transmitted information by securing the channel. Once both the application level protocols and proper security solutions are defined, experiments will be performed so that the added power consumption cost of adding the security layers can be measured, profiled and documented therefore enabling the finding of the best parameters for a desired level of security.

The work will then proceed towards finding effective counter-measures against a specific group of attacks that targets the IoT devices by intensifying the use of its resources, therefore draining the available power and placing the device offline. The ultimate goal is to propose an energy-efficient administration system that can provide the tools to resist those attacks and assure a proper working network.

[1] http://blogs.wsj.com/cio/2015/06/02/internet-of-things-market-to-reach-1-7-trillion-by-2020-idc/

1.2 Document Roadmap

In this document we start by analysing the state-of-the-art in Section 2. This includes the selection of the most adequate protocol stack for our necessities in Section 2.1, an overview of the existing attacks and mitigation strategies in Section 2.2 and a summary of the existing solutions regarding secure insertion of new nodes in an existing network in Section 2.3. All this knowledge will be integrated into our proposed solution defined in Section 3. Section 4 defines how our work will be tested and evaluated so that a power-aware perspective can be achieved. Section 5 states how the development of our solution will unwind over the next months and finally, Section 6 presents the conclusion of this document.

2 Related Work

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. The decision must be based on the particularities of the devices to be used and the objective of the application itself, however a thorough analysis of the existing solutions is a proper way to unveil the strong and weak points of each protocol providing a good basis for an informed decision. A recent survey (January 2015) [2] of IoT enabling technologies, protocols and applications was the starting point for the analysis to follow. The presentation of the available protocols and solutions will follow a bottom-up approach, starting from the data link and physical layer all the way up until the application layer. In particular, the session layer will be left to the end since securing the channel is an optional feature and will be addressed after the application level protocols are properly examined.

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 [3] 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 [4] 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 [5]. 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) [6] 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 [7]. 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) [8] 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.

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 employs. 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 [9] 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

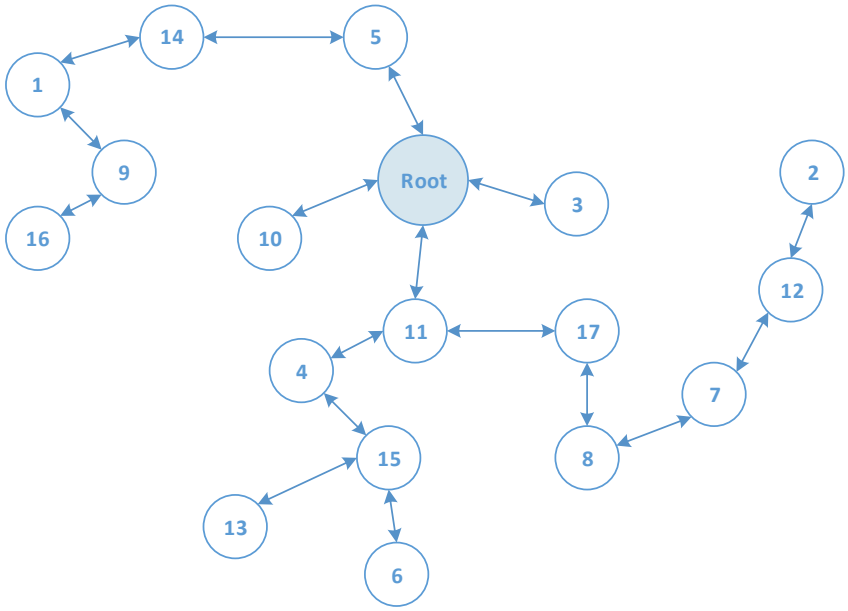

Fig. 1: A Sample RPL DODAG.

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 [10].

Message Queue Telemetry Transport (MQTT)

MQTT [11] is a publish/subscribe messaging protocol designed for lightweight Machine to Machine (M2M) communications. It employs a client/server model and consists of three components: the publisher, the subscriber and a broker. Subscribers register their interest for a specific topic and then get informed by the broker when a publisher generates data regarding that topic. Every message is a discrete chunk of data, opaque to the broker. The broker, on his side, checks authorization of the publishers and subscribers. MQTT supports three Application Level Quality of Service (QoS) levels:

– At Most Once (Fire and Forget): A message will not be acknowledged by the receiver or stored and redelivered by the sender;

– At Least Once: It is guaranteed that the message will be delivered to the receiver, but more that one copy can reach the destination due to message resending. The sender stores the message until it gets an acknowledgement from the receiver;

– Exactly Once: A four-way handshake mechanism is used to guarantee that the message will be received exactly once by the counterpart.

MQTT has support for persistent messages stored on the broker, where the most recent message will be sent to a client that subscribes that topic. Clients can register a custom message to be sent to the broker on disconnect enabling other subscribers to know when a device disconnects. MQTT runs on TCP which in some cases has drawbacks in performance. A performance evaluation of MQTT and CoAP [12] provides comparisons on several protocol facets:

– Influence of Packet Loss on Delay: With low values of packet loss, MQTT experienced lower delays, but as the packet loss increased CoAP performed better. This is due to the greater TCP overheads involved in the retransmissions of messages when compared to UDP;

– Influence of Packet Loss on Data Transfer: CoAP generated less data for each packet loss versus all the MQTT QoS levels;

– Overheads for Message Sizes: When packet loss rate is low, CoAP generates less overhead than MQTT for all message sizes, but as message size grows, the reverse is true. This happens because when the message size is is large, the probability that UDP loses the message is higher than TCP which causes CoAP to retransmit the whole message more often than MQTT.

In order to address the drawbacks on constrained devices, Message Queue Telemetry Transport for Sensor Networks (MQTT-SN) protocol [13] was created. Among the improvements and new features, MQTT-SN runs on UDP, adds broker support for indexing topic names, provides a discovery procedure to help clients without a pre-configured server address and supports devices in sleep state. With this approach, an extra gateway is necessary to convert from MQTT-SN to MQTT so the communications can be understood by the broker.

2.1.4 Session Layer

So far, security issues have not been addressed in any of the previous layers. This is because security is an expensive, optional feature. The application layer protocols rely on underneath layers to achieve secure communications, and network layer protocols assume that if security is necessary then it has already been handled in upper layer protocols. In fact, the session layer is where the security mechanisms are implemented and provides an abstraction layer to application layer protocols. These mechanisms work on top of the transport layer and aim to provide authentication, confidentiality and message integrity.

Transport Layer Security (TLS)

TLS is a well-known security protocol that is used to provide a secure transport layer for TCP communications, allowing the upper layer protocols to be left untouched. TLS operation consists of two phases: the handshake and then the data encryption. During the handshake, both parties negotiate which algorithms will be used during the session, authenticate themselves, and prepare the shared secret for the data encryption. Both HTTP and MQTT work over TCP and use TLS as the adopted security protocol.

Datagram Transport Layer Security (DTLS)

DTLS aims to be the equivalent of TLS over UDP transport layer. DTLS works over datagrams that can be lost, duplicated, or received in the wrong order, therefore needing some extra mechanisms (application layer protocols QoS) to cope with that. Although both CoAP and MQTT-SN work over UDP and use DTLS as the adopted security, some authors argue that DTLS is not a suitable option [14] and defend the need of a new integrated security solution. Some of the presented drawbacks are:

- There is no multicast support, which is a key feature in IoT (topology discovery and update for example);
- Handshake phase is prone to exhaustion attacks on the device resources;
- The loss of a message in-flight requires the retransmission of all the messages in-flight.

A final overview of the analysed protocols and security solutions is given in Table 1. And a comparison of the protocol stack is shown in Table 2.

Table 1: IoT Application Protocols Comparison

Application Protocol	RESTful	Request/ Response	Publish/ Subscribe	Adjustable QoS	Transport	Security
HTTP	✓	✓	✗	✗	TCP	TLS
CoAP	✓	✓	✓	✓	UDP	DTLS
MQTT	✗	✗	✓	✓	TCP	TLS
MQTT-SN	✗	✗	✓	✓	UDP	DTLS

Table 2: Protocol Stack Comparison Overview

Layer	Web	IoT
Application	HTTP	CoAP
Session	TLS	DTLS
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 [16]. The following attacks aim at giving an overview of the existing attack possibilities on different routing solutions as well as existing mitigation strategies. Additionally, a range of attacks that target the RPL routing protocol is also analysed. Since RPL is the selected protocol of our energy efficient stack, it is of special importance to consider and assure the mitigation of attacks that would drain the device's batteries by exploiting this light weight protocol's inner workings.

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 and Stretch Attacks are examples of these 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.

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.

Stretch Attack

The objective of this attack is to create a source route around the network, longer than the one that would be required to transverse the network from the source to the sink. The number of elements in the path would be greater than the optimal path, therefore increasing the total energy consumption by a factor of the number of additional hops. Its success rests on intermediary nodes not

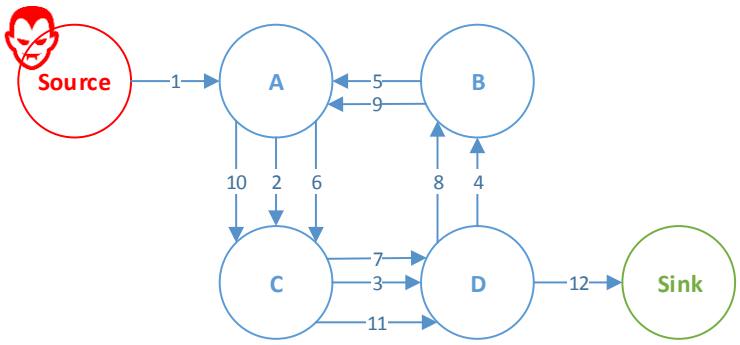

Fig. 2: Carousel Attack.

checking for better paths. Figure 3 shows an example where a vampire node created a path that goes through a greater number of nodes than required to reach the sink.

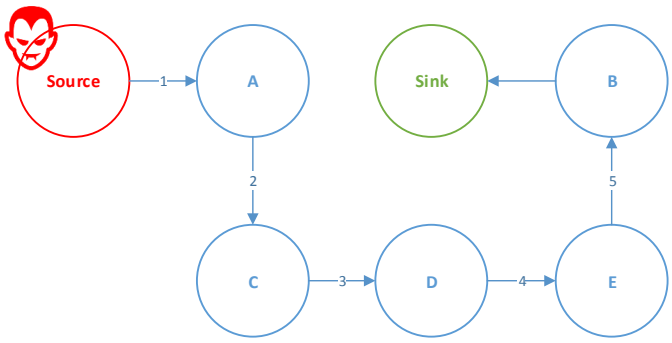

Fig. 3: Stretch Attack.

A limited way of mitigating this attack would be to ensure that path routes have less than the total number of devices on the network. Vasserman and Hoper proposed a property called “no-backtracking” that assures the packet is always moving closer to the sink on every hop [15].

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 and Wormhole Attacks are examples of these 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

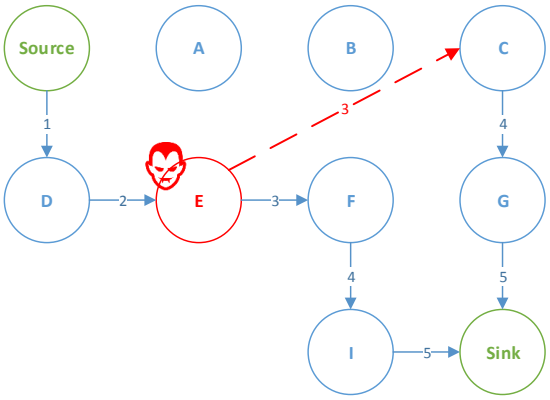

Fig. 4: Directional Antenna Attack.

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 du-

plicated before the path started to diverge would be one who then directed the packet to multiple regions, therefore revealing the attacker.

Wormhole Attack

This attack can be seen as variation of the Directional Antenna Attack but with the collaboration of two or more attackers. Instead of simply forwarding the packets to arbitrary parts of the network, the attacker emulates a link between them and advertises to the network that recently formed connection. This disrupts the topology and has severe impact on routing paths since attackers can indicate that the link cost between them is very low, and therefore influence the forwarding decisions of neighbour nodes. By using these malicious routes, the energy consumption is increased because either this channel does not exist at all (packets are dropped and need to be resent), or the transmission cost between the attackers is greater than the normal message propagation through the network. Figure 5 shows an example where two vampires emulate a connection between them influencing the routing decisions of their neighbours. The hops numbered with prime numbers represent the path taken by a packet after the wormhole is constructed. Although the packet still reached the destination, the cost of the wormhole path is greater than the previous regular path.

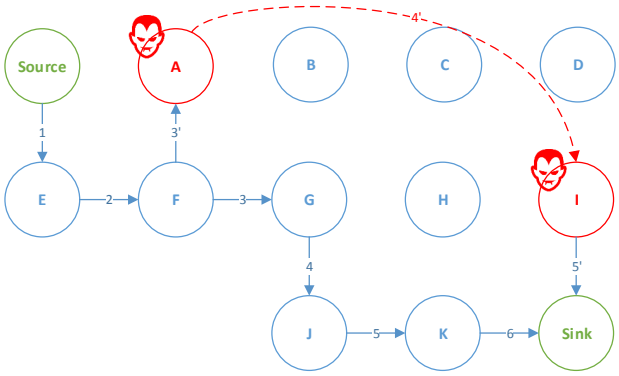

Fig. 5: Wormhole Attack.

Wormhole attacks can be prevented using the Merkle tree authentication [17]. This tree is organized from the leafs towards the root where every parent knows their children and asks them for authentication based on their ID and public key.

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 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.

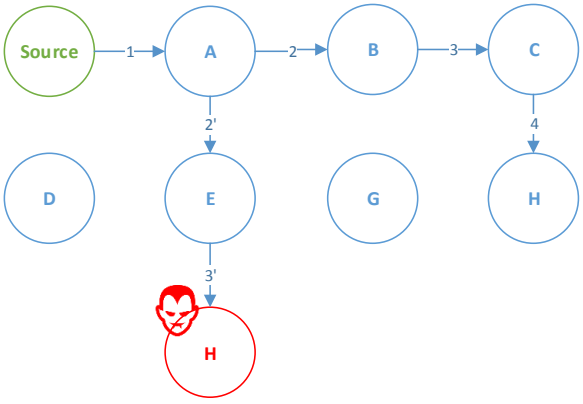

Fig. 6: Clone ID Attack.

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

active member of the network, a large portion of those attacks could no longer be performed. The following bootstrapping techniques were summarized in [18] 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 [18].

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.

2.3.5 Recently Proposed 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. [19] 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 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. [20] 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.

3 Proposed Solution

Before attempting to develop a protocol, or solution, that meets our goals and properly addresses the problems and attacks described in the previous chapters, it is important to define the scope of our work. A common appropriate way to do so is by presenting both a scenario and architecture of the solution. The Internet of Things is currently a hot trend and this work could potentially apply and benefit a wide spectrum of applications, ranging from home environments to large enterprise networks. However, these are domains that have different requirements. A home application should be easy to setup and not require complex configurations to the user. An enterprise solution can benefit from additional administrative configurations as long as the deployment of the devices can be done quickly and easily due to their potential large number. Our work will position itself in the middle of these two domains. We are looking at a complexity and number of devices greater than a home environment but it is not our focus to provide solutions for enterprise networks and their deployment restrictions. It is our belief that a Smart University Campus is an adequate scenario and can effectively demonstrate the needs targeted by our work. In the following sections we will apply the information gathered in the related work sections to formally define what we are trying to achieve, what are the difficulties in achieving those goals and how can we overcome them. Then, a model of a campus with the proposed energy-efficient network architecture will be presented and their component roles explained. Finally, we will discuss additional improvements that are currently out of the scope of this work but can be left for future work.

3.1 Objectives and Requirements

One of the major concerns regarding IoT application is the communication model. For our work, we pose as requirement that the system is power-aware and uses the minimum energy possible. Additionally, the following set of objectives is desirable to build trust and 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.

3.2 System Architecture and Message Flow

As stated in the beginning of the chapter, 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 energy efficient protocol stack defined in Section 4.1

– 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.

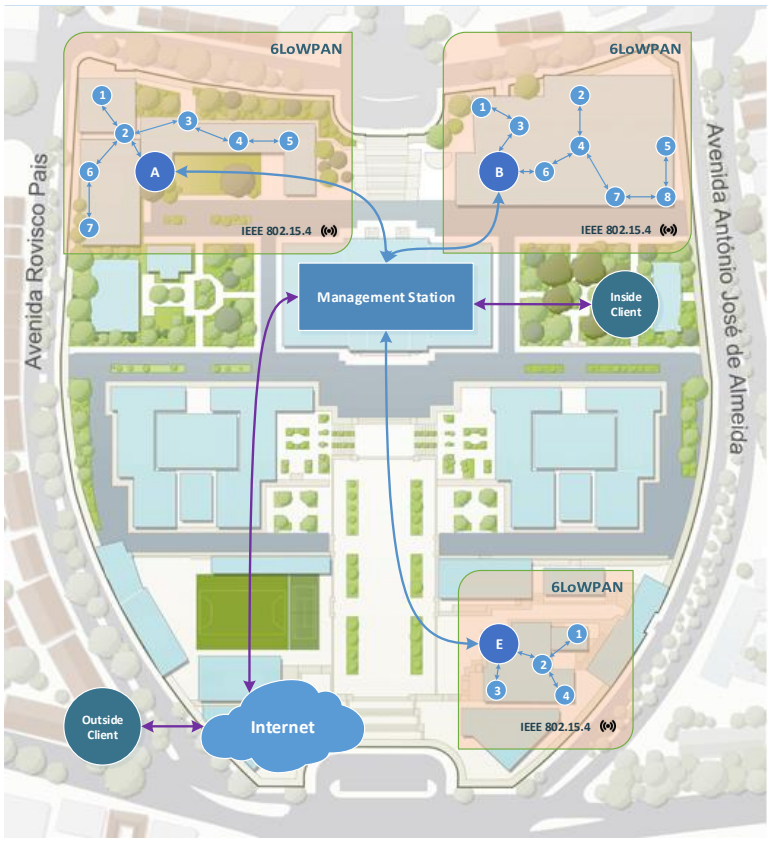

Fig. 7: Global System Architecture

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 and a mapping between each network device and its key pair. This information is facilitated to the Client Observer for creating a secure connection to each sensor node;

- Bootstrapper: The bootstrapper acts as the interface between the management station and the network devices. It generates the device key pair and

writes it together with the shared network key and the Client Observer public key into the new device;

- 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 many overheads in the system. Firstly, and since many clients can connect from different locations, many requests would be performed to the sensor nodes for the same information. Additionally each sensor node would need to be pre-installed with the public keys of all the different user terminals. 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.

Credentials Configuration

In order to achieve secure communications, the new node must connect to the RPL network in a secure way, that is by using a pre-shared group key. After that network setup, it will also need to make a DTLS handshake with the client observer, for that needing a key pair and the public key of the client observer. That information is written into the device during the configuration phase, done by the staff members. Figure 9 shows a sequence diagram of the initial configuration phase. This process will be fully automated without requiring the staff administrator any knowledge of the inner workings of the network and authentication procedures.

As shown in the sequence diagram, the process is initiated by a staff member by connecting a new device to the bootstrapper. The bootstrapper automatically requests the network group key from the key store and the Client Observer public key. Then a new key pair is generated and stored in the key store for that device. Finally the bootstrapper writes the group key, the key pair and the Client Observer public key into the device.

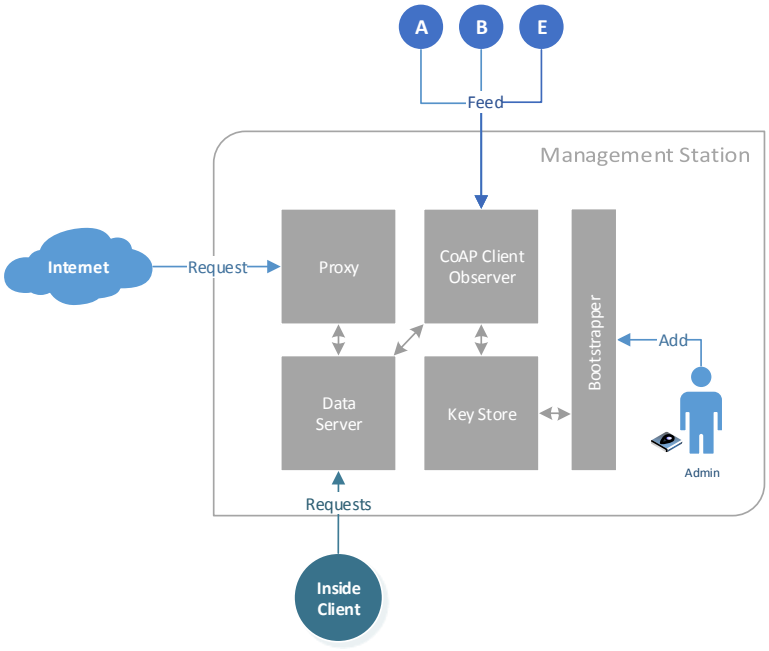

Fig. 8: Central Management Station

Network Layer Bootstrapping

After the credentials configuration phase, the new device is fully equipped with the security credentials required for joining the RPL network. Figure 10 shows a sequence diagram of process started by the new node to join the network topology. The vocabulary used to represent the message exchange was previously presented in Section 2.1.2. All the message exchange is done with the secure versions of the RPL control messages, meaning the data is cyphered with the shared group key.

As shown in the sequence diagram, the process is initiated by the joining device by broadcasting DAO messages to any available neighbour devices in range. The receiving nodes will reply to the new device with a DIO message that provides graph routing information. With this information the new device is able to compute its rank (the distance towards the root) and define its parents based on that metric. After that process is complete, the new device tells his neighbours about its position in the graph using a DIO message and the receiving nodes update their routing tables so that downwards traffic can now reach the new node. This information is further propagated up the network topology until the root is reached.

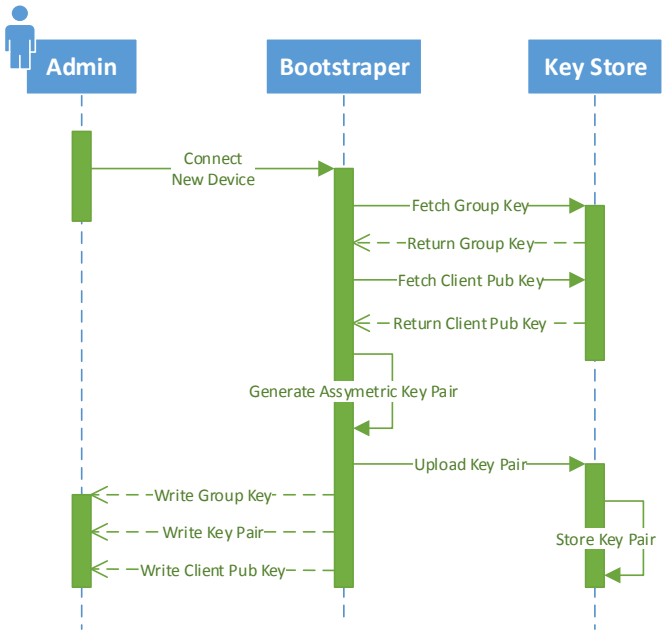

Fig. 9: New Device Initial Configuration

Application Layer Bootstrapping

Although the device is now bootstrapped at the network layer, it still needs to discover and be discovered at the application layer. This means contacting the Client Observer, securing the channel through DTLS and then send new readings as they occur. Figure 11 shows a sequence diagram of process started by the new node to join the CoAP network.

As shown in the sequence diagram the process is started by the new device that advertises its services to the network. That message is eventually received by the Client Observer who uses the new device public key, stored in the key store to start a DTLS secured channel. After the handshake is completed, the Client Observer requests the latest readings to the sensor and sets the observe option meaning each time there is a change in the sensor reading the client will be notified.

3.3 Limitations and Future Work

The use of secure RPL messages with the pre-shared group key together with the raw public/private key pairs assures that a new device is properly authen-

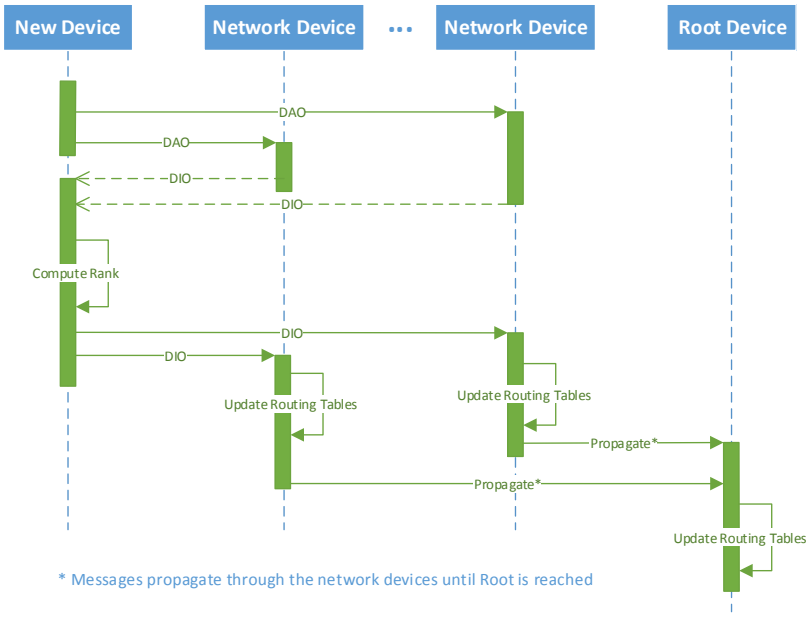

Fig. 10: Network Layer Bootstrapping

ticated when joining the network as well as confidentiality and integrity of the propagated packets. However, public key cryptography is based on computationally intensive mathematical functions that are not very efficient on constrained devices. In fact, asymmetric encryption techniques are almost 1000 times slower than symmetric techniques because they require more computational processing power [21]. In the event that our work reveals the impossibility to use public key cryptography on the most constrained sensing nodes, symmetric cryptography is hereby posed as an alternative.
Also, as discussed in Section 2.2, an attacker could try to introduce himself in the network by stealing the keys from a deployed device. The solutions for that attack can be either software based, assuring that secure memory areas cannot be copied to external locations. Or hardware based, certain integrated circuits assure the stored information cannot be read from them [22]. This attack will not be mitigated in our system as we believe the mitigation strategies are outside the scope of this project and more related to other engineering fields of research.

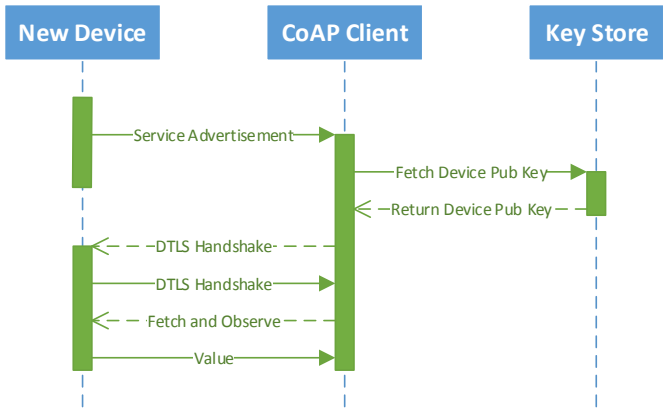

Fig. 11: Application Layer Bootstrapping

4 Work Evaluation

Following the power-aware perspective of this work, our solution will evaluate the power consumption of the system in several scenarios with different network configurations:

- No Security: The system does not provide any type of security credentials. All messages are exchanged in plain text and no node authentication is performed. This will be the baseline.

- Shared Key: The system provides to new nodes a shared group key that enables them to join a secure instance of the network layer protocol RPL therefore assuring node authentication at the network layer.

- Asymmetric Cryptography : The system provides to new nodes an asymmetric key pair and the client observer public key. This enables the new nodes to join a secure instance of the application layer protocol CoAP therefore assuring node authentication at the application layer. Moreover, this enables the DTLS handshake to be performed using raw public keys assuring message confidentiality and integrity.

- Full Security Credentials: The system provides to new nodes both the shared group key, the asymmetric key pair and the client observer public key.

After the data is collected, it will be analysed and charted so that the added power consumption of inserting security measures can be traced to the increasing

power consumption. This will allow a network administrator to consider the type of device and powering mechanisms to deploy based on the security level he desires for a given application.

5 Work Planning

The work on the proposed solution is guided by the following schedule:

- January 9th 2016 - January 31st: Search for the most suitable protocol and operating system implementations for constrained devices.
- February 1st 2016 - February 15th: Implement, Test and Debug the proposed solution on a Network Simulator (first version).
- February 16st 2016 - February 29th: Implement, Test and Debug the proposed solution on a Network Simulator (improved version)
- March 1st - March 15th: Implement, Test and Debug on a Physically Constructed Network (first version).
- March 16th - March 31th: Implement, Test and Debug on a Physically Constructed Network (improved version).
- April 1st - April 11th: Measure, profile and document energy consumptions for the previously described test cases.
- April 12th - May 12th: MSc Thesis Writing
- May 13th: MSc Thesis Delivery

6 Conclusion

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 will be evaluated by measuring its energy consumption with different configurations. These range

from “no security” where messages are sent in plain text and no node authentication is performed, to “full security”, where the node is authenticated at network and application layers and messages are sent cyphered. 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 the prevent the stealing of secure credentials from deployed devices.

References

1. Ukil, A., Bandyopadhyay, S., Pal, A.: Privacy for IoT: Involuntary privacy enablement for smart energy systems. 2015 IEEE International Conference on Communications (ICC) (2015) 536–541
2. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials **PP**(99) (2015) 1–1
3. 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)
4. IEEE Computer Society: Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). Volume 2011. (2011)
5. Kok Seng Ting, Gee Keng Ee, Chee Kyun Ng, N.K.N., Ali, B.M.: The Performance Evaluation of IEEE 802 . 11 against. (October) (2011) 850–855
6. Shelby, Z., Chakrabarti, S., Nordmark, E., Systems, C., Bormann, C., Ericsson: Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). https://tools.ietf.org/html/rfc6775 (2012)
7. Hui, J., Culler, D.: Extending IP to low-power, wireless personal area networks. IEEE Internet Computing **12**(4) (2008) 37–45
8. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., Alexander, R.: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. https://tools.ietf.org/html/rfc6775 (2012)
9. Shelby, Z., Hartke, K., Bormann, C.: The Constrained Application Protocol (CoAP). https://tools.ietf.org/html/rfc7252 (2014)
10. Savolainen, T., Javed, N., Silverajan, B.: Measuring Energy Consumption for RESTful Interactions in 3GPP IoT Nodes. (2014) 1–8
11. OASIS: MQTT Version 3.1.1. OASIS Standard (October) (2014) 81
12. Ma, X., Valera, A., Tan, H.x., Tan, C.K.y.: Performance Evaluation of MQTT and CoAP via a Common Middleware. (April) (2014) 21–24
13. Ibm: MQTT For Sensor Networks (MQTT-SN) Protocol Specification. (2013) 28
14. Alghamdi, T.a., Lasebae, A., Aiash, M.: Security analysis of the constrained application protocol in the Internet of Things. 2nd International Conference on Future Generation Communication Technologies, FGCT 2013 (2013) 163–168
15. Vasserman, E.Y., Hopper, N.: Vampire attacks: Draining life from wireless ad Hoc sensor networks. IEEE Transactions on Mobile Computing **12**(2) (2013) 318–332
16. Pongle, P., Chavan, G.: A survey: Attacks on RPL and 6LoWPAN in IoT. 2015 International Conference on Pervasive Computing (ICPC) **00**(c) (2015) 1–6

17. Khan, F.I.: Wormhole attack prevention mechanism for RPL based LLN network. 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN) (2013) 149–154
18. Fischer, K., Geßner, J., Fries, S.: Secure Identifiers and Initial Credential Bootstrapping for IoT@Work. 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (2012) 781–786
19. Bergmann, O., Gerdes, S., Schafer, S., Junge, F., Bormann, C.: Secure bootstrapping of nodes in a CoAP network. 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) (2012) 220–225
20. Oliveira, L.M., Rodrigues, J.J., Neto, C., de Sousa, A.F.: Network Admission Control Solution for 6LoWPAN Networks. Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (2013) 472–477
21. Kumar, Y., Munjal, R., Sharma, H.: Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures. IJCSMS International Journal of Computer Science and Management Studies **11**(03) (2011) 60–63
22. Lesjak, C., Ruprechter, T., Haid, J., Bock, H., Brenner, E.: A Secure Hardware Module and System Concept for Local and Remote Industrial Embedded System Identification Design Center Graz , Austria. Emerging Technology and Factory Automation (2014)