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. le pretty abstact...

Keywords: power-aware, security, protocols, internet, things, coap, rpl, bootstrapping

Table of Contents

1 Introduction

The Internet of Things (IoT) can be seen as 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 in the constrained nature they're 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 big topic of interest due to the fact that the interconnection of the devices around us can provide information about our choices and whereabouts, therefore reducing our privacy. This document will address these issues from a power-aware perspective, meaning the battery consumption will be of major importance.

2 Main Goals

Given the constrains and limitations of IoT devices described in the previous chapter and the recent protocols, the first objective of this work is to identify a working stack of protocols that takes into account these 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 confidentiality 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 security mechanism that can resist these power-drain (a.k.a vampire) attacks.

3 Document Roadmap

In this document we start by analysing the state of the art in Section 4. This includes the selection of the most adequate protocol stack for our necessities

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

in Section 4.1, an overview of the existing attacks and mitigation strategies in Section 4.2 and a summary of the existing solutions regarding secure insertion of new nodes in an existing network in Section 4.3. All this knowledge will be integrated into our proposed solution defined in Section 5. Section 6 defines how our work will be tested and evaluated so that a power-aware perspective can be achieved. Section 7 states how the development of your solution will unwind over the next months and finally, Section 8, presents the conclusion of the document.

4 Related Work

4.1 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 thoroughly 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) [1] of the IoT enabling technologies, protocols and applications will be 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.

4.1.1 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 are far spread and can be found from homes to airports, the type of radio commonly used, known as Wi-Fi, use 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 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 Ethernet matching 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 on the other hand was created for Low-Rate Wireless Private Area Networks (LR-WPAN) and its specifications 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 act as coordinator or normal nodes. And the Reduced Function Device (RFD) that are very simple, with very restricted 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 and much greater than 802.15.4. But 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 and average energy consumption. However this comes at the cost of a far lower transmission range[2]. 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 transmission range is not a problem.

4.1.2 Network Layer

6LoWPAN

The IoT vision, as presented in the introduction, 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) 6LoWPAN 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. [3] 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. RPL 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 it's 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 the the topology, either source routing or stateful routing tables are used. (More Information on this two types of routing are given in sections 4.2.1 and 4.2.2). 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 Object (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 the 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.

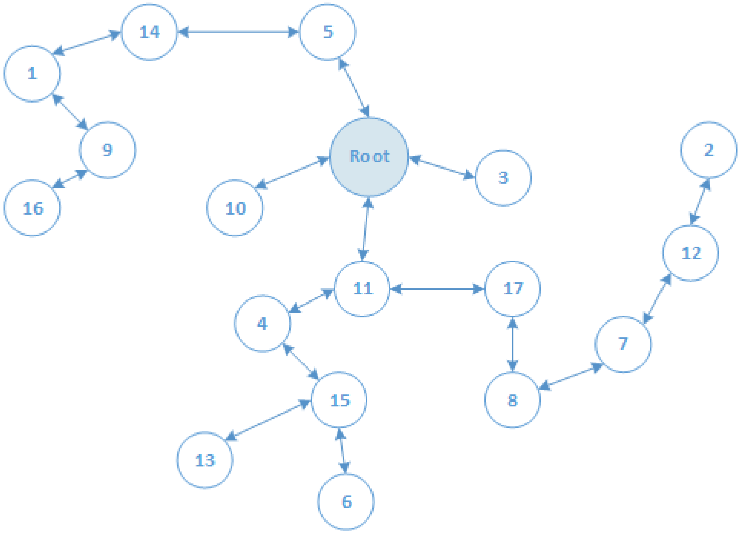

Fig. 1. A Sample RPL DODAG

4.1.3 Application Layer

Hypertext Transfer Protocol (HTTP)

HTTP is an application level protocol that works in the 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 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, non-confirmable follow the fire and forget model. While 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 situation where CoAP would consume more resources than HTTP [4]

Message Queue Telemetry Transport (MQTT)

MQTT 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 is 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 can reach the destination due to message resending. The sender stores the message until it gets an acknowledge 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 causes drawbacks in performance. A performance evaluation of MQTT and CoAP [5] 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[6] 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 understand by the broker.

4.1.4 Session Layer

So far security issues have not been address 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 in 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 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 [7] 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.

4.2 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 the well-known Denial of Service (DoS) at the application layer to a physical intruder replacing some node on a sensor field. 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:

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/Phy	802.11	802.15.4

Battery Depletion Attacks aka 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 [8], while others target specific protocols and implementations [9]. 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 devices batteries by exploiting this light weight protocol inner workings.

4.2.1 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. A couple examples of these attacks are the Carousel and Stretch 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 packets 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 exit after the first hop through the D node.

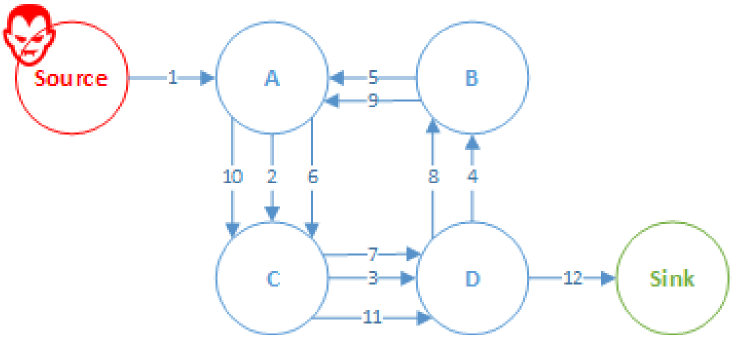

Fig. 2. Carousel Attack

Existing mitigations 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 longer source route around the network than the one who 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. It's success rests on intermediary nodes not 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.

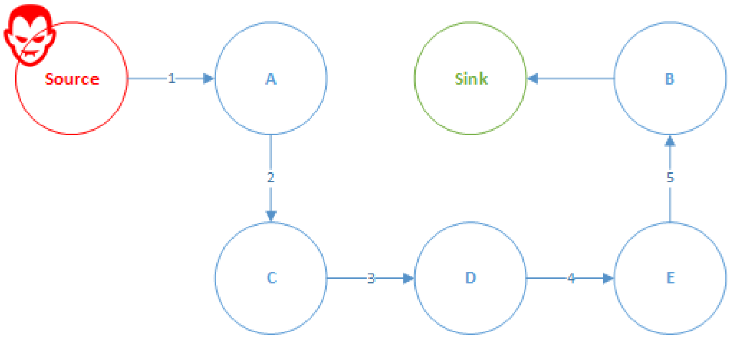

Fig. 3. Stretch Attack

A limited way of mitigating this attack would be to ensure that path routes have less than the total number on 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. [8]

4.2.2 Stateful Protocols

In systems that use this type of routing protocols, network nodes are aware of the network topology and it’s 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. A couple examples of these attacks are the Directional Antenna and Wormhole 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 2 different routes towards it's destination

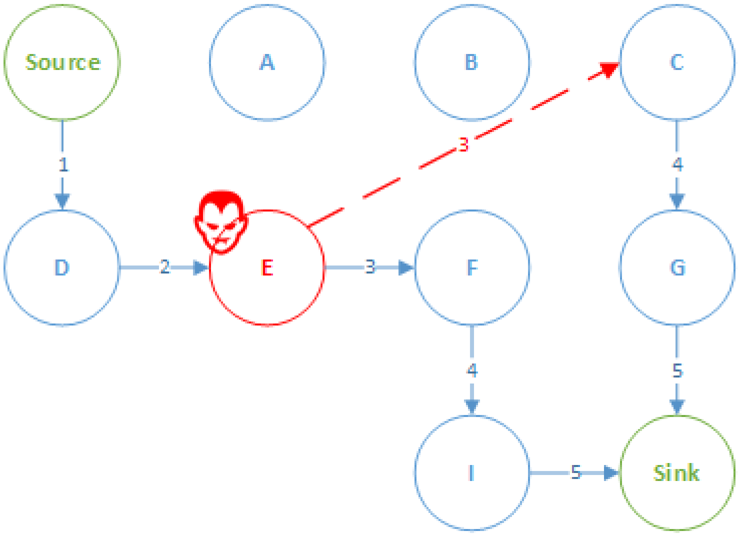

Fig. 4. Directional Antenna Attack

A mitigation strategy could be to analyse the route paths of a given packet that reached the sink more that one time. The last node identifier to appear duplicated before the path started to diverge would be one who then directed the packet to multiple regions, 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 advertise 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 therefore influence the forwarding decisions of neighbour nodes. By using these malicious routes, the energy consumption is increased because either this channel doesn't 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.

Wormhole attacks can be prevented using the Markle tree authentication. This tree is organized from the leafs towards the root where every parent knows

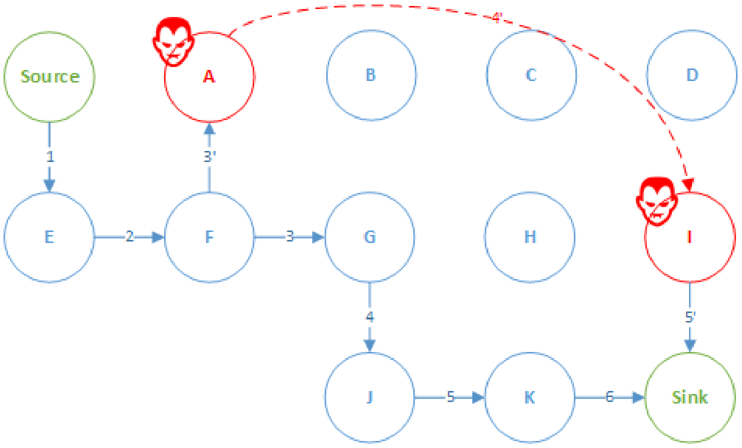

Fig. 5. Wormhole Attack

their children and asks them for authentication based on their ID and public key.

4.2.3 RPL Specific Attacks

Selective Forwarding Attack

In a selective forwarding attack, a malicious node can launch a DoS attack by selectively forwarding packets. It's main goal is to disrupt routing paths but can be used to filter any protocol. 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 improve the path quality.

Hello Flooding Attack

The Hello in the name of this attack comes from the initial message a node send 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 so 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.

4.2.4 Protocol Independent

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 it 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. A couple examples of these attacks are the Clone and Sybil 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 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.

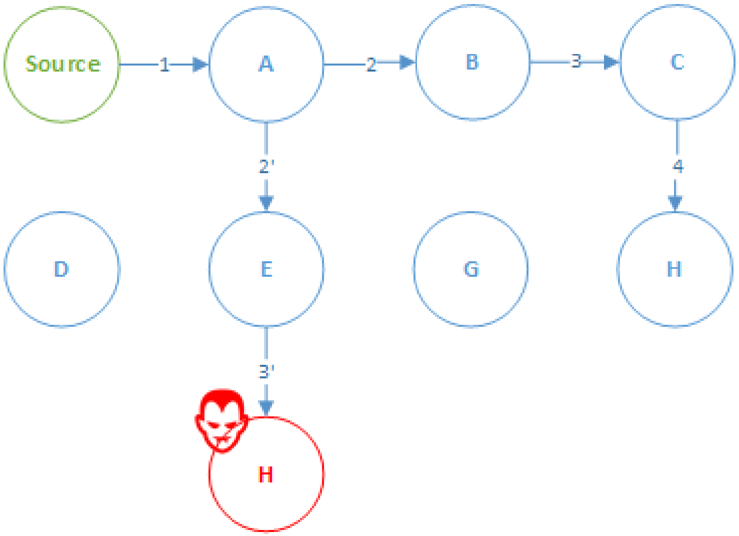

Fig. 6. Clone ID Attack

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.

4.3 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 [10] and aim at providing secure bootstrapping for IoT devices.

4.3.1 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 to insert the password. If the case of large number of devices, this approach would be unsuitable due to the management effort of manually deploying the tokens to the devices [10].

4.3.2 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, 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.

4.3.3 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 deploy of the new node should receive through a secure channel an one time password, that would then use to authenticate the node. This one time password could be used to authenticate locally generated key material (either certificate requests or generated key pair) towards the management station or root node. 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.

4.3.4 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 load of the firmware or stored in a separate integrated circuit designed for credential storing. In the second case, this method 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.

4.3.5 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 in the nature of IoT devices. Bergman et al. [11] 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 propose short audible or visual feedback to the human installer when the secret is received and assume that potential eavesdroppers can not intercept this transmission. Finally, this secret is used to setup de DTLS connection. This approach has major security drawbacks on the secret transmission phase, the authors propose limiting the radio power to a low level and disable data forwarding beyond the local network segment, but this techniques cannot assure that an attacker wont be able to intercept the transmission.
Oliveira et al. [12] proposed an admission control solution for 6LoWPAN networks based on administrative approval. Each joining node would broadcast it's 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 it's 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.

5 Proposed Solution

vou falar de: ter esta stack bonitinha que os outros ainda não têm preocupar em resolver este ataques ao nível do routing por meter lá as chaves logo falar que o custo de meter lá as chaves conpensa na segurança do sistema depois clone ficam resolvidos garantindo que um atacante não consegue extrair a informação dos nós.

5.1 Architecture

5.2 Protocol Analysis and Selection

6 Work Evaluation

7 Work Planning

8 Conclusion

References

1. 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

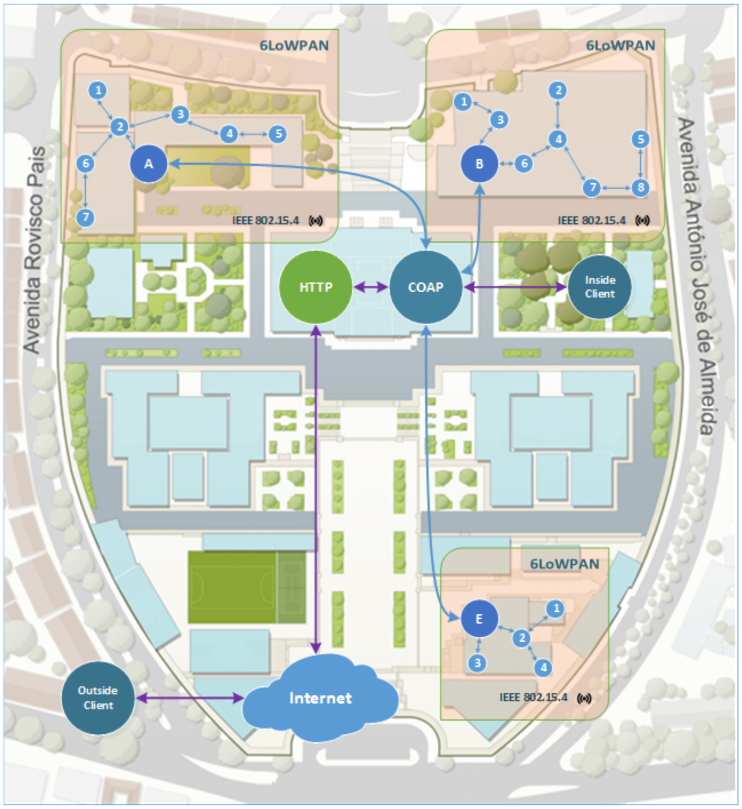

Fig. 7. Global System Architecture

2. 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
3. Hui, J., Culler, D.: Extending IP to low-power, wireless personal area networks. IEEE Internet Computing **12**(4) (2008) 37–45
4. Savolainen, T., Javed, N., Silverajan, B.: Measuring Energy Consumption for RESTful Interactions in 3GPP IoT Nodes. (2014) 1–8
5. 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
6. Ibm: MQTT For Sensor Networks (MQTT-SN) Protocol Specification. (2013) 28
7. 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
8. 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
9. 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

10. 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
11. 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
12. 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