Key-Pre Distribution for the Internet of Things Challenges, Threats and Recommendations

Ayman El Hajjar

Abstract The Internet of Things is the next evolution of the Internet which will substantially affect human life. IoT is important because it is the first of its kind that is propelling an evolution of the Internet and smart environment; It is clear that secure communication between IoT devices is essential and the threats and risks for having an insecure IoT are a lot bigger than for conventional Internet connected devices. The motivation behind this chapter is to set variables needed to investigate the performance of both the probabilistic scheme or the deterministic scheme approaches and to find a reliable and efficient mechanism for nodes within the IoT and to establish trust by securing end-to-end communication by having a certain pre distributed key scheme that will enable such communication by the use of a Key Pre-distribution scheme KPS.

Keywords IoT · Distributed Sensor Networks · Key Pre-distribution scheme · Threat Model for IoT · SCADA attack · Man in the middle MiTM

1 Introduction

In this chapter we will introduce the challenges that comes with implementing a key distribution algorithm in the context of the Internet of Things. We look at the threats on key distribution in IoT environment and how key distribution can be performed between devices using the routing protocol for 6LoWPAN, RPL routing protocol.

The motivation behind this chapter is to find a reliable and efficient mechanism for nodes within the IoT and to establish trust by securing end-to-end communication by having a certain pre distributed key scheme that will enable such communication by the use of a Key Pre-distribution scheme KPS. Many KPS were proposed for Distributed Sensor Networks (DSN). DSN shares a lot of the IoT characteristics and can be used as a starting point for this research.

A. El Hajjar (✉)
University of Westminster, London, UK
e-mail: A.ElHajjar@westminster.ac.uk

H. Jahankhani and A. El Hajjar (eds.), *Wireless Networks*, Advanced Sciences and Technologies for Security Applications,
https://doi.org/10.1007/978-3-031-33631-7_1

2 Chapter Question

The question this chapter is looking to investigate is whether the probabilistic key distribution scheme (KPS) proposed in [1] and the deterministic Key Pre-distribution proposed in [2] can be used in an Internet of Things (IoT) environment similarly to how they are in used in the context Wireless Sensor Networks (WSN).

While looking at the research question, we can deduce several sub questions that need to be answered in order to identify the effectiveness of KPS for the IoT. We first need to establish the differences between DSN and IoT in order to assess whether different KPS schemes used in DSN are suitable for the IoT. This will be done by investigating whether the identified schemes can provide the same security measure without any modification of the parameters used. We will then evaluate the impact of those KPS schemes use on the IoT devices and networks without any modification. Based on the answer of the previous question, we will be able to identify the required modifications that are needed to achieve the necessary security measures in the context of the IoT with acceptable security performance and an affordable resource usage on its devices. After identifying the required modifications needed, if any, we should look at what can be optimized in the IoT in order to determine the most effective security measure with the least cost in term of resources.

The main objective in this research is to establish a reliable and efficient mechanism for nodes within the IoT to establish trust by a mean of establishing a secure end-to-end communication by having certain pre distributed key scheme that will enable such a communication. A Pre-distribution Key (KPS) is therefore needed. Not a lot of research was done in this field. Many PKS were proposed for WSN and ZigBee. Both network technologies share a lot of the IoT characteristics and can be used as a starting point for this research. Some of the research was done on securing the communication of between the nodes in the IoT network but not in securing the routing topology formation. To my knowledge, using a Key Pre-Distribution Scheme in the context of the IoT is something that was not looked at before to secure the routing formation. Future research needs to find the answers for the following questions in order to develop/identify the most suitable (KPS) for the IoT are: To achieve our main objective the research needs to find the answer for the following questions:

1. Determine the advantage and disadvantages of using the probabilistic or deterministic key pre distribution schemes for distributed sensor networks in the context of the Internet of Things.
2. Evaluate the performance of the simulated key management schemes for distributed sensor networks on the Internet of Things using the same variables used in the distributed sensor networks to achieve full connectivity and assess if they are enough to achieve full connectivity in the Internet of Things network.
3. Evaluate the overhead of experiments to determine the quality of service obtained from implementing the key management scheme for distributed sensor networks on the Internet of Things.

2.1 Differences Between DSN and IoT

Many KPS were proposed for Distributed Sensor Networks (DSN). DSN shares a lot of the IoT characteristics and can be used as a starting point for this research.

Although both DSN and IoT are considered infrastructure-less networks and operate on an Ad-Hoc basis, many essential characteristics (by definition) between them are not shared. Those characteristics change the whole environment of IoT in comparison with DSN. Distributed Sensor Networks are not able to use classical IP based protocols simply because it is very difficult to allocate a universal identifier scheme for a large DSN network and proprietary protocols are usually used to identify unique devices. A distributed sensor network can operate by itself sending data to a centralized entity in order to monitor the physical conditions of an environment. An IoT network requires one or more devices to act as a sink and to connect the network to other types of networks such as the Internet in order to send data collected. The devices in an IoT network do not need to be the same and all can communicate to complete a specific task.

For that reason, DSN nodes cannot inter-operate and communication between various nodes only exist for routing purposes and to allow data to reach the centralized location. Since IoT nodes are able to inter-operate with the existing Internet infrastructure, each of them needs its own unique identifiable Internet protocol (IP) address rather than a proprietary protocol.

The challenge of the addressing and identifying nodes in DSN present us with a complete set of challenges that differs in the scenario of an IoT network. The flow of data in a DSN network is most of the time in one direction towards the sink connected directly to the centralized location. The flow of data in an IoT network is bi-directional as a mote can either send data to the Internet or receive instructions from another entity. This difference means that the routing protocols used for a DSN network cannot be used in an IoT network. In most applications of DSN networks, route discovery base routing protocols are used; Ad Hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and Optimized Link State Routing (OLSR). Each of those protocols have their own characteristics however they all share two important features. All are proprietary protocols and are not IP based protocols but proprietary classless protocols and they only allow route discovery and route establishment messages to be exchanged between nodes in both directions in comparison with the IoT where data can only travel through one direction at a time.

There are some challenges that need to be taken into consideration when implementing the KPS in the context of the Internet of Thing. The use of a suitable symmetric encryption protocol is also essential. Different encryption protocols require different time to decrypt as each will present different limitations in terms of computation and processing speed.

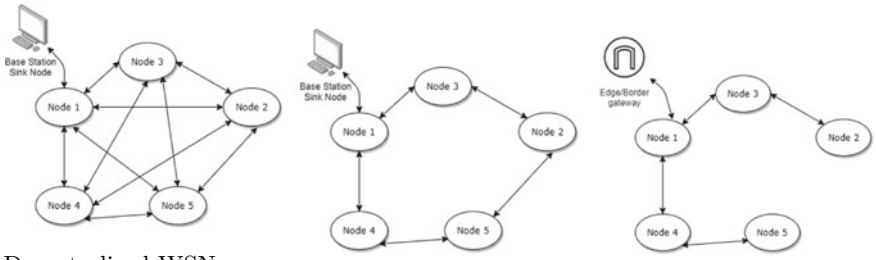
DSN network nodes were assumed to have proprietary unique identifiers simply because they were never intended to be used as part of a large network such as the Internet. This is not a practical solution for the IoT as data is needed to travel between two directions and sometimes directly to the internet. For that reason, it

requires an IP based routing protocol. Most of the conventional devices on the Internet uses the Transmission Control Protocol/Internet Protocol TCP/IP communication suite to identify how data should travel between devices, in which format and using which route. This suite however was not intended to be used with the IoT and it is not suitable for the IoT as the devices that participate in this type of network are considered lightweight resource constraints devices. Some attempts were made to develop a unique addressing scheme for the IoT until most researchers and IoT device manufacturers agreed that devices should use the same addressing scheme as the Internet to make it easier for devices to communicate with the Internet. Using IP protocols in sensor networks simplify the connectivity model as the hierarchy of the devices in the network can be flattened. This also removes the complexity of having devices to translate between proprietary protocols and standard Internet protocols as explained in [3].

However, the TCP/IP suite was still considered heavy and IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) which is a new communication suite designed specifically for the IoT was created. 6LoWPAN defines how to layer, transmit and deal with data using IPv6 over low data rate, low power, and small footprint radio networks as identified by IEEE 802.15.4 in [4] radio. Routing is a fundamental piece of the overall IPV6 architecture for the Internet of Things. The networks in these environments can be described as Low Power and Lossy Networks (LLN), meaning they often operate with significant constraints on processing power, memory and energy translating into high data loss rates and low data transfer rates and instability. Routing protocol for Low Power and Lossy Networks (RPL) was developed to translate the potential of Internet of Things into reality. The objective of RPL is to target networks which comprise of thousands of RPL DODAG where the majority of the nodes have very constrained resources. RPL solves the unique challenges that IoT brings to the exchange of messages between nodes in a conventional DSN network.

The physical nature of the IoT devices makes it difficult to implement security schemes to secure communication between nodes. In an IoT device, limited resources are available such as the limitation of storage capacity and processing power. A KPS used to secure communication between DSN devices assumes the presence of several routes to a mote and if a shared key between two nodes does not exist an alternative secured route can be found. This is not the case in the IoT and therefore a large number of keys is needed to ensure that all links between nodes is secure. This will require a large storage space for a large scale IoT network. This solution will present a problem for IoT devices.

The architecture of the IoT, similarly to the DSN is of Ad-Hoc mode (also known as peer to peer). It means that there is no centralized entity that organizes the distribution of the keys between nodes. It also means that all links between any two nodes needs to contain a shared key. This will naturally result in an increase of the number of keys that each node should have to make sure that all links between two nodes are secured by the use of the shared link. This presents us with another challenge as the implementation of any suggested solution will be limited by the storage capacity of devices used regardless of which KPS scheme is used. The difference in how



Decentralized WSN representation DSN representation IoT representation

Fig. 1 Five nodes physical topology comparison for WSN, DSN and IoT networks. Each node in Wireless and Distributed Sensor networks can have one or more links. Distributed sensor networks establish enough links to have a route to the sink node. IoT nodes establish nodes with preferred parent to reach root node

devices communicate in an IoT in comparison with DSN as explained in Sect. 2.1 means that devices that do not share a secure key cannot communicate indirectly if a secure route between them cannot be identified when the routing table is formed using RPL. This will lead to several devices in the network not being included in the routing table and thus will not be allowed to join the network.

Secure communication between end to end IoT devices is essential. IoT devices are meant to exchange data from critical infrastructure such as devices in smart cities, smart houses, SCADA systems and other important infrastructure. Those devices will not only be exchanging important data but also participating in automated decision making and this makes the security of the communication between those devices more important. An attacker listening to the communication between those devices, if the devices are communicating in plain text, can simply intercept the message and understand it. For example, a camera device sending a message to a heating source in a smart home, informing the heater that there is no one at home in order for the heating to automatically go off, will give clues to any attacker who is listening to this communication and thus be able to deduct that the house is empty and a theft can take place.

The differences between WSN, DSN and IoT can be summarized by two main differences, first how the nodes connect with each other and report to the sink node or gateway and the number of connections between the different nodes in the network. An example of how five nodes form a network in the different networks is shown in Fig. 1. Wireless and Distributed Sensor networks can take the form of different physical topologies outlined in [5] and can be summarized by three topologies decentralized self organizing, centralized architecture and grid networking techniques.

In order to transform WSN into a viable technology to make the IoT vision cost-effective and deployable, authors in [6] claim the need for middleware-layer solutions fully compliant with accepted standards (or largely adopted specifications). This in fact is essential to allow sensor nodes in IoT to communicate with the Internet to process its data.

2.2 Threat Model for IoT and the Research Problem

In this section we will look at the threats on Internet of Things and identify where the research problem that this research is attempting to solve fits. Authors in [7–9] categorized the attacks in different categories as shown in Fig. 2. As we can see from Fig. 2, several attacks can be mitigated if nodes in an IoT network communicate in a secure way. The motivation to mitigate those threats all at once is because by ensuring that only nodes that share one or more secret key can communicate we ensure that all nodes that have joined that network are genuine and trusted.

In Sect. 6.6 we present the attack surfaces that exploits threats identified in this section as relevant to this research problem such as attack surfaces on key distribution, key storage or the process of routing formation and maintenance.

The threats that this research is attempting to solve and shown in Fig. 2 highlighted in blue or gray can be summarized below.

Generalized category threats on IoT identifies threats that do not only exist in IoT environments or multi-layer threats. Security and user privacy are essential to maintain in any network and protecting the confidentiality and integrity of data from violation will prevent devices from leaking private user data and confidential data. Researcher in [10, 11] have identified that IoT devices have higher chances of leaking private and confidential data due to the lack of reliable authentication, the lack of data encryption and the lack of network access control measures.

Cryptanalytic attacks explained in [12, 13] exploits the weaknesses in the cryptographic algorithm and can result if successful in the attacker discovering the original message. There are several cryptanalytic attacks that all networks can be vulnerable to depending on the cryptographic algorithm used. Cryptanalytic attacks will result in the violation of confidentiality, integrity and availability of data transmitted in such networks. The type of encryption used to encrypt data will be essential to ensure that the IoT secure DODAG is not vulnerable to cryptanalytic attacks. Ensuring that no malicious node can compromise the network will also prevent this type of attacks as devices will not be able to participate in the network in order to carry such attacks. The solution proposed in this research will have a direct impact on mitigating this attack.

Denial of Service (DoS) attacks on IoT devices explained in [14] result in resources exhaustion due to the physical features of the internet of Things devices such as low processing power and low battery consumption. Resources exhaustion attacks include jamming of communication channels, extensive unauthorized access and malicious utilization of critical IoT resources and those attacks result in operational functionality of IoT devices or non availability which result in disruption of services. 96% of the devices involved in Distributed Denial of Service DDoS attacks were IoT devices and participated in Botnets as discussed in [14, 15]. Although this attack is out of context of the research and having encrypted data between nodes do not prevent it directly, however some DoS attacks are carried out by malicious nodes that exhausts the resources of other nodes until they crash. Securing the routing formation will prevent malicious nodes from joining the network and hence protecting networks against DoS attacks. DDoS on the other hand cannot be prevented if it is the result of nodes participating in a botnet.

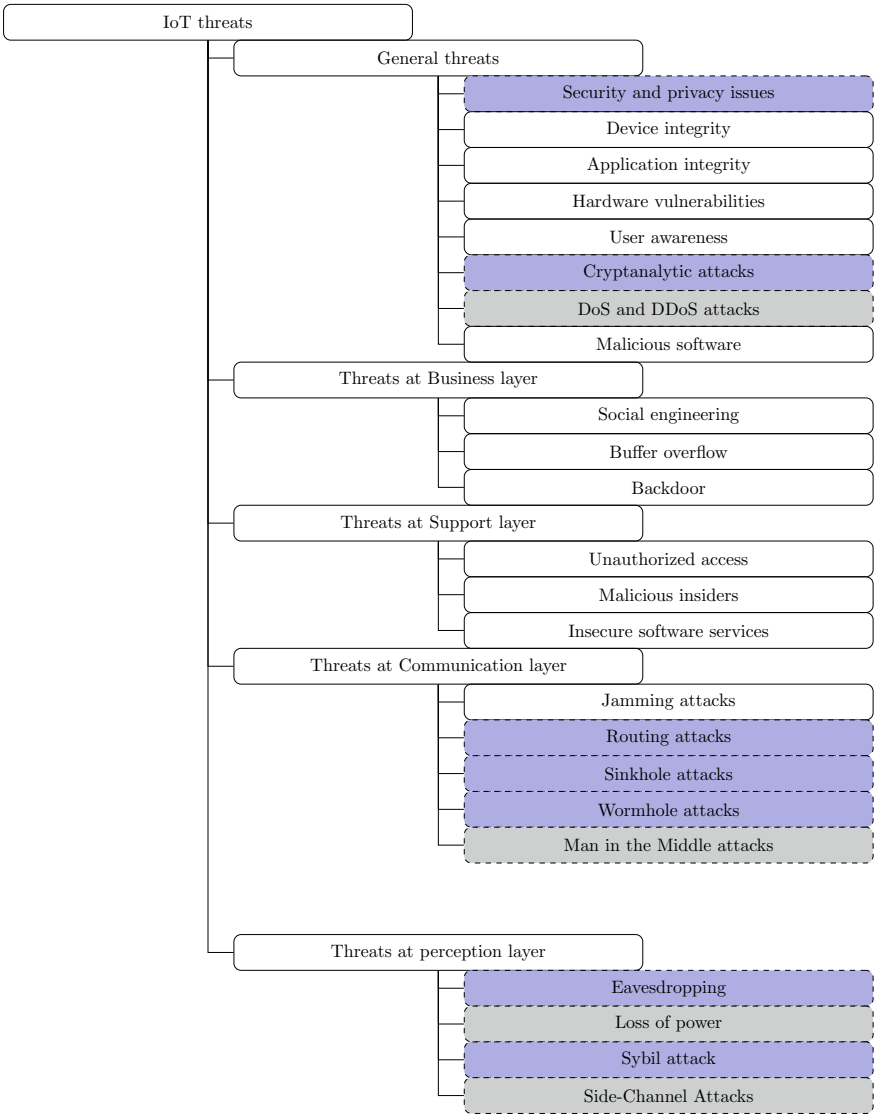


Fig. 2 IoT threats categorized based on the IoT layers that is affected. For each category, the threats are either considered not related to the chapter topic (in white background), directly related (blue background), indirectly related (gray background)

Various attacks threaten the Internet of Things routing formation and routing process as investigated in [16–19]. IoT RPL DODAG is vulnerable to a selective forwarding attack. In this attack malicious nodes do not participate in transmitting the packets received by it and destroys the routing path of the network by doing so as explained in [16, 20]. The Blackhole attack explained in [21] is an example of a

selective forwarding attack in which a malicious node do not forward any packet and breaks the DAG in the routing table. HELLO flood attacks threaten the RPL DODAG formation process. In this attack when a genuine node utilizes HELLO messages to join a network a malicious node can capture this packet and use it to declare itself a neighbour. In this case, the DODAG Information Object DIO messages can be utilized with strong routing metrics in order to start such an attack as in [20] and leads to the malicious node joining the RPL DODAG. Rank attacks in RPL are other type of attacks in which malicious nodes advertise falsely their rank as discussed in [22, 23]. Increased rank attack and decreased rank attack are two examples of rank attacks examples in which a malicious node falsely advertise its rank either lower or higher and repeatedly does this in a way that it disrupts the routing topology as nodes will have to regularly update their preferred parent based on the new rank that the malicious node is advertising.

Routing attacks are at the core of the motivation of this research since preventing routing attacks will mitigate several other threats such as preventing malicious nodes from joining the network. Other type of routing attacks discussed in [20, 24, 25] are the sinkhole attack and the wormhole attack. In the sink node attack, malicious nodes redirect the traffic of a network to a specific node that acts as a sink node. Several malicious nodes participate in this attack by advertising a particular route that leads to the malicious node that is acting as a sink node. In the wormhole attack investigated in [20, 26, 27], the malicious nodes create direct links with each other and force the network traffic data through those links rather than links with intermediate nodes. Sinkhole attack and wormhole attack can be prevented by securing the routing formation process and encrypting the traffic between nodes as it will prevent malicious nodes from joining the network.

Other Man in the middle MiTM attacks discussed in [11, 28, 29] are defined as a form of eavesdropping in which malicious actors can intercept the traffic exchanged between two nodes and tamper with the exchanged node or use the captured packets to carry on further attacks. Different examples of MiTM can threaten the confidentiality and authenticity of the Internet of Things network such as Neighbor Discovery Protocol NDP poisoning explained in [30, 31], Address Resolution Protocol (ARP) poisoning identified in [32], replay attacks in [33, 34] and session hijacking in [35, 36]. Man in the Middle attacks can be prevented indirectly since encrypted traffic will prevent malicious node from carrying on such attacks and they are unable to decrypt the traffic to get the parameters and values needed to tamper the data in session in hijacking or to replay the traffic.

Threats at perception/physical layer consists of sensors, actuators, computational hardware, identification and addressing of the things. Securing data sensing and data collection in this layer is essential as they are done at this layer as explained in [8]. Threats in this layer are related to the physical aspects of the device such as resources exhaustion that causes battery drainage and loss of power by preventing a node from sleeping or going into saving mode. Malicious actors investigated in [11, 37] can physically install unauthorized devices in order to sniff the traffic and extract valuable information. Eavesdropping and traffic analysis can go together as the sniffed traffic can be captured and analysed by a network packets analyser to gather information

about the nodes and their environment in the network. The solution protect against the threat of eavesdropping since malicious nodes cannot decrypt or understand the context of the captured or sniffed traffic. Loss of power if it is caused by the threat of DoS attacks can be indirectly protected by the proposed solution as it prevents malicious nodes from joining the network in order to generate large amount of traffic and exhausts nodes until the battery is drained. If the loss of power is the result of physical tampering of the devices then this solution will not prevent it.

Sybil Attack investigated in [9, 38, 39] is a form of attack that the IoT networks can be subject to. In this attack a malicious node impersonate one or more genuine nodes in the network and generate fake data and thus violating the trust and confidentiality between the nodes in the network. This attack can be prevented by this solution as the malicious nodes will be prevented from joining the network.

Side channel attacks as defined by [40] is based on side-channel information about the encryption device that are found on the physical device when data is being processed in the perception and physical layers of the device such as information about data processing time or power consumption of the device when encrypting/decrypting various messages and during the computation of different security protocols. This threat can be mitigated indirectly if a strong encryption algorithm is used to prevent malicious actors from data information leaked generated when the encryption and decryption process of the keys takes place.

3 Internet of Things

Internet of Things (IoT) is the next evolution of the Internet which will substantially affect human life. IoT is important because it is the first of its kind that is propelling an evolution of the Internet and smart environment—an evolution that will lead to innovative applications that have the ability to revolutionize our lives and our surroundings.

The vision of having a variety of physical elements “Objects” and “things” connected to the Internet is what forms the IoT. In the conventional Internet, most of the devices connected to the Internet were used directly by humans and needed a direct interaction from a human being to be able to generate data. The IoT vision enabled objects and things to interact with an external entity and send data without the interference of a human. No human participation is needed and objects are able to take decisions based on data received, sent or generated.

Thus the term of the Internet of Things explained in [41] is now considered as a global network which allows the communication between human-to-human, human-to-things and things-to-things that is anything in the world by providing a unique digital identity to each and every object.

The idea is that all objects connected to the IoT will contain embedded technology, allowing them to interact with internal states or an external environment. Those objects will be able to sense and communicate thus changing how and where decisions are made and who makes them [42].

The IoT is an emerging technology closely related to other research areas like Peer to Peer Networking, Mobile computing, Pervasive or Ubiquitous computing, Wireless Sensor Networks, Cyber Physical Systems, Real Time Analytics, etc. Technologies like ZigBee and Wi-Fi Direct can be widely deployed to achieve the notion of smart cities, eventually achieving a globally integrated smart world. However, there are ongoing issues like architecture design, hardware design, cost accountability, identity, privacy, and security issues for building new ices and solutions in IoT [43].

The applications and usage of the Internet are multifaceted and expanding on a daily basis. The Internet of Things (loT), Internet of Everything (loE) and Internet of Nano Things are new approaches for incorporating the Internet into the generality of personal, professional and societal life [44].

Applications of IoT encompasses medical implants, alarm clocks, wearable systems, automotives, washing machines, traffic lights, and the energy grid. It is expected that 50 billion devices will be interconnected by 2030. Having this huge Global Network will result in the generation of a huge unprecedented amount of data.

Internet protocols have always been considered too heavy for sensor networks and thus the 6LoWPAN protocol stacks were created [45]. 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices" and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things [4].

4 6LoWPAN

To achieve the vision of the Internet of Things, a review of the currently used Internet protocols and standards was needed. The Internet Protocol (IP) was always considered a protocol for Local Area Networks, Wide Area Networks, PCs and servers. The IP protocol was not intended to be used with Wireless sensor networks, Personal Area Networks and the sensor itself. The main reason why it was not intended to be used is that the IP is too heavy for those applications. Sensor networks are meant to be lightweight resource constraints devices.

However, recently there has been a rethinking of the many misconceptions about the IP. The main discussion was to answer this question "why invent a new protocol when we already have IP" thus the development and standardization of 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) was carried out. A simple 6LoWPAN architecture is shown in Fig. 3 and outlines the basic concept of connecting low power devices in a 6LoWPAN network with a conventional IPv4/v6 network by using an edge router.

6LoWPAN technology realizes the IPv6 packet transmission in the IEEE 802.15.4 based WSN. And 6LoWPAN is regarded as one of the ideal technologies to realize the interconnection between WSN and Internet which is the key to build the IoT [46].

6LoWPAN defines how to layer, transmit and deal with data using IPv6 over low data rate, low power, and small footprint radio networks 6LoWPAN as identified by IEEE 802.15.4 radio. 6LoWPAN protocols resides between the data link layer

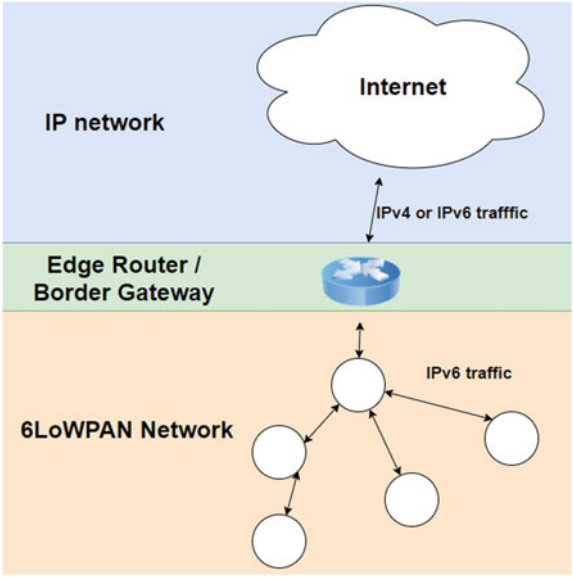


Fig. 3 The 6LoWPAN simple architecture comprises the IoT network layer, the edger router and the connection to the Internet where the data collected from lower layers are analysed and processed [45]

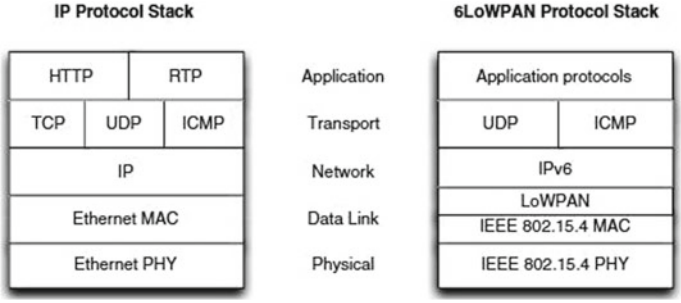


Fig. 4 IP and 6LoWPAN protocol stacks as presented in 6LoWPAN the wireless embedded Internet by Shelby and Bormann [45]. The representation of each layer in the 6LoWPAN shows how the logical communication between the layers at the same level can be interpreted, i.e. communication between the IP network layer and IPv6

and the network layer. The adaptation of the full IP format and the 6LoWPAN in performed by the edge router that translates conventional IP traffic to 6LoWPAN traffic as is shown in Fig. 4 in relation to an IPv6 stack.

Using IP protocols in Sensor networks simplify the connectivity model, as the hierarchy of the devices in the network can be flattened. This also removes the complexity of having devices to translate between proprietary protocols and standard Internet protocols [3].

IoT applications are implemented using a wide range of proprietary technologies which are difficult to integrate with larger networks and Internet-based services. Where as the 6LoWPAN approach is an IP based one, these devices can be connected easily to other IP networks which doesn't require any translation gateways or proxies, and which can use the existing network infrastructures [47].

It is normal to assume that using IP is too heavy in terms of code size, protocol complexity, required configuration infrastructure or head and protocol overhead. Implementation of 6LoWPAN can easily fit into 32 kb flash memory parts which is suitable for the Internet of Things devices and wireless Networks. 6LoWPAN uses the IPv6 thus the need for configuration servers such as DHCP and NAT is not available as the IPv6 has the Zero Configure and Neighbour Discovery capabilities. The use of IPv6 also allowed the protocol to define a unique stateless header compression mechanism for the transmission of IPv6 packets in as few as 4 bytes.

A key attribute to 6LoWPAN is the IPv6 (Internet Protocol version 6) stack, which has been a very important introduction in recent years to enable the IoT. IPv6 provides a basic transport mechanism to produce complex control systems and to communicate with devices in a cost-effective manner via a low-power wireless network.

The challenges to develop Internet of Things applications using 6LoWPAN stack similarly but with more complexity and can be identified specifically to routing and security of all nodes on the network.

5 Routing

Routing is a fundamental piece of the overall IPv6 architecture for the Internet of Things. It became clear as intelligent devices were proliferating into all aspects of life, that a new routing protocol would be required for devices on the smart grid as well as other smart devices operating in harsh environments such as smart grids, manufacturing plants, commercial buildings, and on transportation networks. The networks in these environments can be described as Low Power and Lossy Networks LLN, meaning they often operate with significant constraints on processing power, memory and energy translating into high data loss rates, low data transfer rates and instability. Routing Protocol for Low-Power and Lossy Networks RPL is a routing protocol on IPv6 that will translate the potential of Internet of Things into reality.

As of 2011, RPL has been deemed ready by the IETF as a proposed standard RFC. The objective of RPL is to target networks which comprise of thousands of nodes where the majority of the nodes have very constrained resources. RPL protocol consists of routing techniques that organize networks in units called Directed Acyclic Graphs DAG. DAG is structure where all nodes are connected but there is no available round trip path from one node to another [48].

Each DAG structure is called Destination Oriented Directed Acyclic Graph (DODAG). The DODAG starts at the root node or sink. The root node is initially the only node that is a part of the DODAG, until it spreads gradually to cover the

whole IoT as DODAG Information Object DIOs are received down in the network. In a converged IoT network, each RPL router has identified a stable set of parents, each of which is a potential next hop on a path towards the root of the DODAG as well as the calculated rank for each preferred parent for each node.

When a router needs to decide on the preferred route to use and on the preferred parent, it will emit DODAG Information Object (DIO) messages using link local multicast thus indicating its respective rank in the DODAG (usually the distance to the root is considered the metric "hop count"). All routers will do the same and each router will receive several DIO messages. Once it receives all DIO messages, it will calculate its own rank and select its preferred parent and then itself start emitting DIO messages.

Since RPL is a Distance Vector routing protocol, it restricts the ability for a router to change rank. A router can freely assume a lower rank but it can assume a higher rank, it is restricted to avoid count to infinity problem. For a router to assume a greater rank, it has to ask the root to trigger global recalculation of the DODAG by increasing a sequence number DODAG version in DIO messages. The protocol tries to avoid routing loops by computing a node's position relative to other nodes with respect to the DODAG root. RPL is mostly communication between multipoint to point routes from the sensors inside the LLN and towards the root. RPL by way of the DIO generation provides this as upward routers.

Downward routes are only used by parents to issue Destination Advertisement Object (DAO) messages, propagating as unicast via parents towards the DODAG root. In RPL routers two modes exist one that is non storing mode, where an RPL router originates DAO messages, advertising one or more of its parents and unicast it to the DODAG root. The root once it receives all DAOs from all routers, it can use source routing for reaching advertised destinations inside the LLN. The second mode, the storing mode, where each RPL router on the path and the root records a route to the prefixes advertised in the DAO and the next hop.

A routing metric is a quantitative value used to find the cost of a path and helps in making the routing decision in case there are different routes available. In Low power Lossy Networks a metric is a scalar used to find the best path according to the objective function.

Another important fact about the protocol's design is the maintenance of the topology. Since most of the devices in LLN and 6LoWPAN networks are typically battery powered, it is crucial to limit the amount of sent control messages over the network. To do that, a trickle timer algorithm is used since the time for each router to send a DIO message is relevant to how the network topology is changing. If the network topology keeps on changing, which means if routers keep on finding in DIO message out dated messages, it means the trickle timer for DIO messages needs to be smaller. If routers keep on finding messages and information stored up to date (similar) it means no need for DIO messages at this rate, the timer is made bigger.

5.1 RPL Messages

To understand the messages of RPL and how they propagate over a RPL DODAG, we need to first look at how the messages of RPL are sent. RPL messages typically exist in an IEEE 802.15.4 network. The data frame of the IEEE 802.15.4 encapsulates a compressed header of the IPv6 as shown in Table 1 and the payload shown in Fig. 5. The compressed header of IPv6 is used since a full IPv6 packet does not fit in an IEEE 802.15.4 frame [49]. The IEEE 802.15.4 standard specifies a maximum transmission size (MTU) of 127 bytes, yielding about 122 bytes of actual Media Access Control (MAC) payload [50]. The payload also contains the ICMPv6 control message contained with the IP datagram, also shown in Fig. 5. The type of messages in ICMPv6 is set to 155 when RPL control messages are being sent [51]. Thus an IPv6 header compression is used, encapsulated in the IEEE 802.15.4 header as per IEEE 802.15.4 specifications in [52]. The IPv6 compressed header of IEEE 802.15.4 header is of 5 bytes in size and shown in Table 1.

RPL messages are considered part of the data frame message and they are sent in the payload of an 802.15.4 packet. Control of RPL and the order for a root to form a DODAG and for a node to join a DODAG are shown below:

1. DODAG Information Solicitation message (DIS) (Sect. 5.1.1)
2. DODAG Information Object (DIO) (Sect. 5.1.2)

Table 1 Size of the different fields of the IEEE 802.15.4 frames

Name of field	Size in bytes
LOWPAN_IPHC base encoding	2
Context identifier extension	1
Next header	1
Group ID to identify all-RPL-nodes multicast address	1

This is encapsulated in the IPv6 compressed header

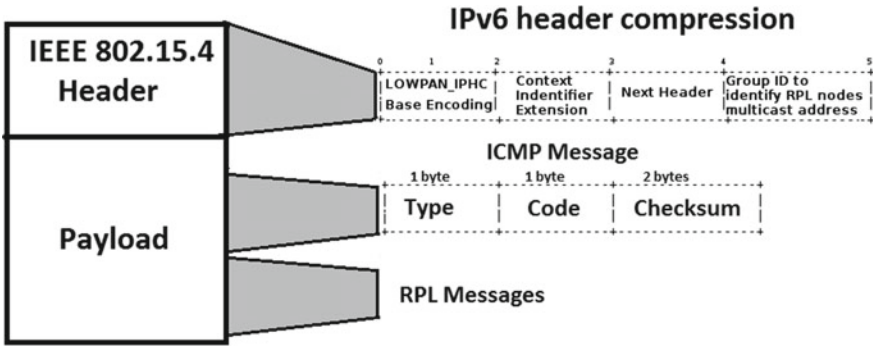


Fig. 5 IEEE 802.15.4 frame with the header and the payload sizes as defined by the 802.15.4 specifications

3. Destination Advertisement Object (DAO) (Sect. 5.1.3)
4. Destination Advertisement Object Acknowledgement (DAO-ACK) (Sect. 5.1.4)—Optional.

5.1.1 DODAG Information Solicitation (DIS)

The DODAG Information Solicitation (DIS) message shown in Fig. 6 as per the definition of RPL messages in [53] may be used to solicit a DODAG Information Object from a RPL node. Its use is analogous to that of a Router Solicitation as specified in IPv6 Neighbour Discovery. A node may use DIS to probe its neighbourhood for nearby DODAGs.

5.1.2 DODAG Information Object (DIO)

A DIO base object structure shown in Fig. 7, as per the definition of RPL messages in [53] consists of 24 bytes. This is followed by the route information bytes and metric container bytes.

The RPLInstanceID is an 8 bits field set by the DODAG root that indicates which RPL instance the DODAG is part of. The version number is set by the DODAG root

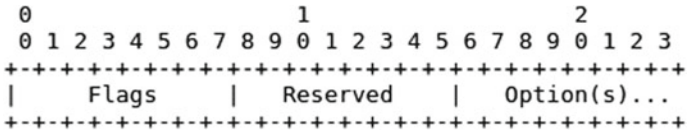


Fig. 6 DIS base object frame with the 8 bits unused field reserved for flags. This field is ignored by the receiver and set to zero by the sender. The reserved and the option fields are ignored by the receiver

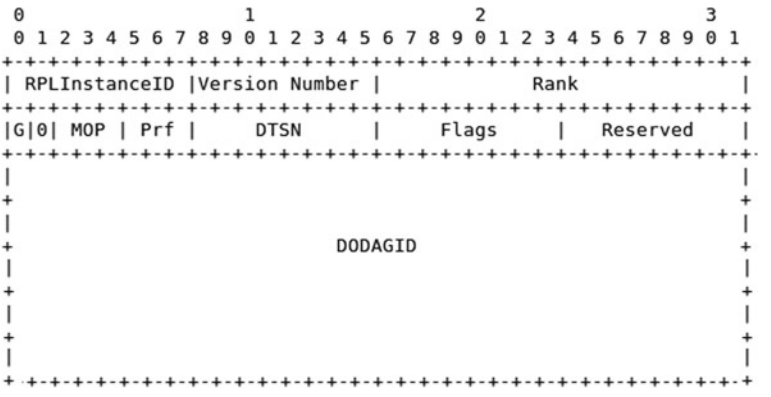


Fig. 7 DIO message embedded in a 6LoWPAN frame

Table 2 DIO message fields

Name of field	Size in bytes
DIO base object (Fig. 7)	24
DODAG configuration option	16
Route information option	24
Metric container	16

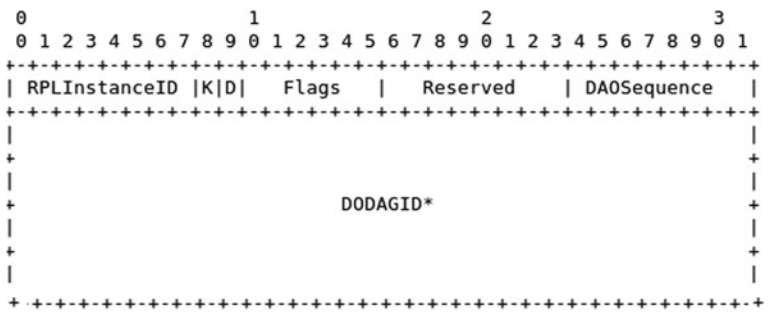


Fig. 8 Destination Advertisement Object (DAO) base object

and the rank is a 16 bit unsigned integer indicating the DODAG Rank of the node sending the DIO message. This defines how the nod receiving the DAO will decide how it will respond to the DIS message. The DODAGID is a 128 bit IPv6 address set by the DODAG root that uniquely identifies a DODAG. The DODAGID must be a rootable IPv6 address belonging to the DODAG root as defined in [53].

The DIO message shown in Fig. 7 is embedded in the payload of the IEEE 802.15.4 data frame and takes 80 bytes as defined by Routing Over Low power and Lossy networks (ROLL) in ROLL and shown in Table 2.

The metric container shown in Table 2 takes 16 bytes from the IEEE 802.15.4 message. This consists of 2 bytes for "type and option length", 6 bytes for "ETX metric object" and 6 bytes "ETX constraint object".

5.1.3 Destination Advertisement Object (DAO)

A DAO base object format shown in Fig. 8 as per the definition of RPL messages in [53] consists of 24 bytes. This is followed by the route information bytes, metric containers bytes and other IPv6 bytes.

The structure of a DAO message shown in Table 3 is 60 bytes.

Table 3 DAO message fields

Name of field	Size in bytes
DAO base object (Fig. 8)	20
DODAG configuration option	16
Route information option	24

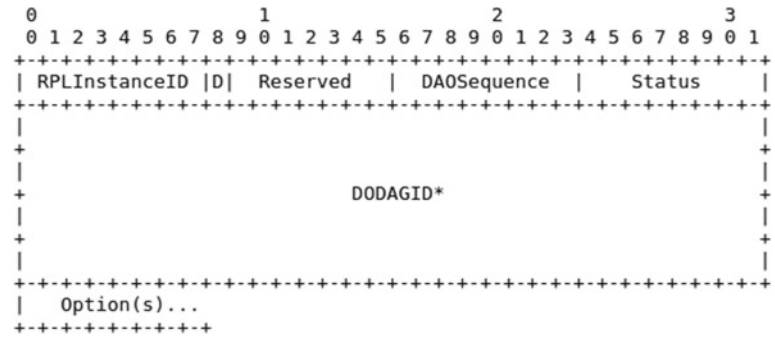


Fig. 9 Destination Advertisement Object Acknowledgement (DAO) base object

Table 4 DAO-ACK message fields

Name of field	Size in bytes
DAO-ACK base object	20
DODAG configuration option	16
Route information option	24

5.1.4 Destination Advertisement Object Acknowledgement (DAO-ACK)

The DAO-ACK message shown in Fig. 9 as per the definition of RPL messages in [53] is sent as a unicast packet by a DAO recipient (a DAO parent or DODAG root) in response to a unicast DAO message. It consists of 20 bytes. This is followed by route information bytes, metric containers bytes and other IPv6 bytes.

The 69 bytes of the DAO-ACK message are shown in Table 4.

5.2 RPL Routing Metrics and Constraints

For a DODAG to be constructed, the root will need to first broadcast a DODAG Information Object (DIO) message, discussed in details in Sect. 5.1.2 to all its neighbours. This DIO message will propagate through the network. Each node that receives a

DIO message will consider the sender node a preferred parent to reach the root node until it receives another DIO message with better metrics to reach the root from another node [53]. The DIO message contains the DAG Metric Container option that is used to report metrics along the DODAG. The DAG metric Container may contain one specific metric or various numbers of metrics and constraints as chosen by the implementer [53]. Should multiple metrics and/or constraints be present in the DAG Metric Container, their use to determine the "best" path can be defined by an Objective Function (OF). Directed Acyclic Graph (DAG) that attempts to minimise path costs to the DAG root according to a set of metrics and Objective Functions. There are circumstances where loops may occur and RPL is designed to use a data-path loop detection method. This is one of the known requirements of RPL, and other data-path usage might be defined in the future. The graph is constructed by the use of an Objective Function (OF) which defines how the routing metric is computed. In other words, the OF specifies how routing constraints and other functions are taken into account during topology construction.

The Routing Metrics and Constraints for RPL are defined in [53]. Those metrics and constraints are used in addition to other variables together and identified as OCP 0 for Objective Function Zero (OF0). When the DAG Metric container contains a single metric, called an aggregated metric, that adjusts its value as the DIO message travels along the DAG. A node decides on its preferred parent and thus its rank based on this single rank only [54]. For example if the node Energy metric is aggregated along paths with an explicit Min function. The best path is selected through an implied Max function because the metric is Energy and thus the node with the highest Energy is selected as preferred parent. However, when a DAG Metric Container contains several metrics, then they need to be used in the order of criteria to be achieved. Each Metric criterion will be first met before moving to the next metric when deciding on a rank of a node (preferred parent). Several Metrics/Constraint Objects exist. In this section, the Metrics and Constraint Objects are discussed.

Each of the objects below is a metric that can be considered a criterion in selecting a preferred parent. When chosen, it will be defined in the DAG Metric Container. Only one object of each metric can exist in the DAG Metric Container. Those metrics objects fall into two categories:

1. Node Metric/Constraint Objects (Sect. 5.2.1)
2. Link Metric/Constraint Objects (Sect. 5.2.2).

5.2.1 Node Metric/Constraint Objects

Node Metric/Constraint Objects are metrics or constraints related to nodes such as node processing power, node memory, congestion situation, node energy (e.g. in power mode, estimated remaining lifetime and hop count to reach the node). Several metrics exist to calculate those criterias

1. Node State and Attribute Object (NSA): The NSA object is used to provide information on node characteristics. Those characteristics of node state and attribute are defined by an 8 bit flag. This flag can have the value 'A' flag or '0' flag. 'A' flag means that applications in this node may use aggregation node attribute in their routing decision to minimize the amount of traffic on the network. '0' flag means that node workload may be hard to determine and express in some scalar form. Node workload will then be set based upon CPU overload, lack of memory or any other node-related conditions.
2. Node Energy Object: The Node Energy Object is used as a metric when it is desirable to avoid selecting a node with low energy. Power and energy are clearly critical resources in most LLNs. Node Energy Object is calculated by determining the node Energy Consumption needed for each node [55].

$$EE = \frac{Power_{now}}{Power_{max}} \times 100$$

   where EE is the energy estimation for each node.
3. Hop Count Object (HP): The Hop Count Object (HP) is used to report the number of traversed nodes along the path. The HP object may be used as a constraint or a metric. When used as a constraint, the DAG root indicates the maximum number of hops that a path may traverse. When that number is reached, no other node can join that path. When used as a metric, each visited node simply increments the Hop Count field.

5.2.2 Link Metric/Constraint Objects

Link Metric/Constraint Objects are metrics related to links connecting nodes together such as link quality, link latency, throughput and reliability. Similarly to the Node Metric Objects, only one of each of the objects discussed below can be used at a time in the DAG Metric Container. Several link objects exist to calculate those criteria.

1. Throughput: The throughput is the amount of data moved successfully from one point in the network to another in a given time period. The throughput object is calculated by calculating the estimated actual throughput. This is done when each node reports the range of throughput that their link can handle in addition to the currently available throughput.
2. Latency: The latency is the amount of time a packet takes to travel from one point in the network to another. The latency object is calculated by calculating the estimated actual latency. This is done when each node report the range of latency that they allow in addition to the latency they are suffering based on the power consumption.
3. The Link Quality Level Reliability Metric (LQL) [53]: The Link Quality Level (LQL) object is used to quantify the link reliability using a discrete value, from 0 to 7, where 0 indicates that the link quality level is unknown and 1 reports the

highest link quality level. The LQL can be used either as a metric or a constraint. When used as a metric, the LQL metric can only be recorded. For example, the DAG Metric object may request all traversed nodes to record the LQL of their incoming link into the LQL object. Each node can then use the LQL record to select its parent based on some user defined rules.

4. The ETX Reliability Object: The ETX metric is the number of transmissions a node expects to make to a destination in order to successfully deliver a packet. In contrast with the LQL routing metric, the ETX provides a discrete value (which may not be an integer) computed according to the formula below:

$$ETX = \frac{1}{PRR_{down} \times PRR_{up}}$$

and where PRR is (Packet Reception Rate)

$$PRR = \frac{\text{Number of Received Packets}}{\text{Number of Sent Packets}}$$

and ETX is expected transmission count.

5.3 RPL Objective Functions

An Objective Function defines how a RPL node selects the optimised path within a RPL instance based on the routing metrics and constraints. It provides specific optimisation criteria like minimise hop count, path ETX, Latency etc. RPL forms Directed Acyclic Graph (DAGs) based on the objective function. The OF guides RPL in selection of the preferred parents and candidate parents. It is also used by RPL to compute the ranks of a node. All upward traffic is forwarded via the preferred parent. The ETX metric of a wireless link is the expected number of transmissions required to successfully transmit a packet on the link. Objective Function ETX uses ETX metric while computing the shortest path.

The Objective Function (OF) is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how nodes translate one or more metrics and constraints, which are themselves defined in [55], into a value called Rank, which approximates the node's distance from a DODAG root in term of the number of hops it needs to reach it. An OF also defines how nodes select parents. When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the content of the DIO. OCP is an identifier assigned by the Internet assigned Numbers Authority (IANA). Two OCP values are assigned, one for OF0 given identifier OCP 0 and the other for the Minimum Rank with Hysteresis Objective Function (MRHOF) given the identifier OCP 1. It is worth noting that OF0 and MRHOF are the only two Objective Functions that are fully defined by IETF. ETX is still a draft however it is widely used. Two other draft

Objective Functions that are not used as much and are proven not to be effective are Load Balancing Objective Function (LBOF) and Traffic Aware Objective Function (TAOF).

In this section, the objective functions overview is shown with how each of them format the Destination Advertisement Object (DAO) message with values relevant to the OF and the decision of the preferred parent.

5.3.1 Objective Function Zero

The metrics and constraints objects discussed above in Sect. 5.2 are used, if selected in the DAG Metric Container to select the preferred parent. Each of those individually can be used to determine the path for a node to the root. However when multiple DAG Metric Containers are used, those metrics are grouped together in a Objective Function.

An OF0 implementation first computes a new variable called step of rank (SR). This variable is associated with a given parent from relevant link properties and metrics as explained below.

The SR is used to compute the amount by which to increase the rank along a particular link. It first starts by making sure the node is a candidate preferred parent (received DIO message) by making sure the link is valid in terms of connectivity and suitability. After this, the node makes sure that the candidate node has acceptable node attribute (power, energy, CPU, memory, battery) to be able to act as a preferred parent. If all those criteria are fulfilled, the node selects the candidate as a preferred parent and changes the value of its rank in the RPL DAO message by increasing the rank it received in the DIO of the candidate by 1.

The variable rank increase RI is represented in units expressed by the variable M, which defaults to the fixed constant that is defined in [53] as the default minimum hop rank increase $DRI = 256$.

The SR is then computed for that link by multiplying by the rank factor Rf and then possibly stretched by a term Sr that is less than or equal to the configured stretch of rank. The resulting RI is added to the Rank of preferred parent $R(P)$ to obtain that of this node as below:

$$R(N) = R(P) + RI$$

where

$$RI = (Rf \times SR + Sr) \times M$$

5.3.2 The Minimum Rank with Hysteresis Objective Function (MRHOF)

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a distance vector IPv6 routing protocol designed for LLN networks. RPL is designed for networks which comprise thousands of nodes where the majority of the nodes have very constrained energy and/or channel capacity. To conserve precious resources, a routing protocol must generate control traffic sparingly [56]. However, this is at odds with the need to quickly propagate any new routing information to resolve routing inconsistencies quickly.

RPL organises its topology in a Directed Acyclic Graph (DAG). An RPL DAG must have at least one RPL root and a Destination Oriented DAG (DODAG) is constructed for each root. The root acts as a sink for the topology by storing all routes to all nodes in the DODAG in the routing table. The root may also act as a border router for the DODAG to allow nodes that belong to different DODAGs to communicate [53].

For a DODAG to be constructed, the root will need first to broadcast a DODAG Information Object (DIO) message, discussed in detail in Sect. 5.1.2, to all its neighbours. This DIO message will propagate through the network. Each node that receives a DIO message will consider the sender node a preferred parent to reach the root node until it receives another DIO message with better metrics to reach the root from another node [53].

The DIO message contains the DAG Metric Container option that is used to report metrics along the DODAG. The DAG metric Container may contain one specific metric or various numbers of metrics and constraints as chosen by the implementer [53]. Should multiple metrics and/or constraints be present in the DAG Metric Container, their use to determine the "best" path can be defined by an Objective Function (OF).

The Objective Function (OF) is identified by an Objective Code Point (OCP) within the DIO Configuration option. An OF defines how nodes translate one or more metrics and constraints, which are themselves defined in [55], into a value called Rank, which approximates the node's distance from a DODAG root in term of the number of hops it needs to reach it. An OF also defines how nodes select parents. When a new DIO is received, the OF that corresponds to the Objective Code Point (OCP) in the DIO is triggered with the content of the DIO. For example, OF0 explained in Sect. 5.3.1, is identified by OCP 0 by the Internet assigned Numbers Authority (IANA). The Minimum Rank with Hysteresis Objective Function (MRHOF) explained in Sect. 5.3.2, is the other Objective Function defined by IANA and given the identifier OCP 1.

Several Objective Functions were designed in order to fulfil specific tasks. A Destination Advertisement Object (DAO) message, for each node receiving the DIO message, will be sent to the candidate node (DIO message origin) with values relevant to the OF and the decision of the preferred parent.

This Objective Function describes the Minimum Rank with Hysteresis Objective Function (MRHOF) [57], an Objective Function that selects routes that minimise a

metric, while using hysteresis to reduce lagging in response to small metric changes. First, it finds the minimum cost path, i.e., path with the minimum Rank. Second, it switches to that minimum Rank path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. This second mechanism is called "hysteresis". MRHOF works with additive metrics along a route, and the metrics it uses are determined by the metrics that the RPL Destination Information Object (DIO) messages advertise.

MRHOF uses current minimum path cost for the cost of the path from a node through its preferred parent to the root computed at the last parent selection. It also uses the following parameters

- MAX LINK METRIC: Maximum allowed value for the selected link metric for each link on the path.
- MAX PATH COST: Maximum allowed value for the path metric of a selected path.
- PARENT SWITCH THRESHOLD: The difference between the cost of the path through the preferred parent and the minimum cost path in order to trigger the selection of a new preferred parent.
- PARENT SET SIZE: The number of candidate parents including the preferred parent, in the parent set.
- ALLOW FLOATING ROOT: If set to 1, allows a node to become a floating root. A node MAY declare itself as a Floating root, and hence have no preferred parent, depending on system configuration.

The calculation of the ETX metric is given constant selected metrics based on [58]. The metrics are:

- MAX LINK METRIC: Disallow links with greater than 4 expected transmission counts on the selected path (Set to 512).
- MAX PATH COST: Disallow paths with greater than 256 expected transmission counts (Set to 32,768).
- PARENT SWITCH THRESHOLD: Switch to a new path only if it is expected to require at least 1.5 fewer transmissions than the current path (Set to 192).
- PARENT SET SIZE: If the preferred parent is not available, two candidate parents are still available without triggering a new round of route discovery (Set to 3).
- ALLOW FLOATING ROOT: Do not allow a node to become a floating root (Set to 0). If FR is 0 and no neighbours are discovered, the node does not have a preferred parent and must set the minimum path cost to PS.

5.3.3 ETX

The expected transmission count ETX metric discuss is based on the number of expected transmissions required to successfully transmit and acknowledge a packet on a wireless link. The ETX metric is commonly used in wireless routing to distinguish between paths that require a large number of packet transmissions from those

that require a smaller number of packet transmissions for successful packet delivery and acknowledgement however RPL uses this metric to establish preferred parent based on the value of the ETX metric of the link as defined in [55, 59] and make it available for route selection. This is called ETX Objective Function (ETX).

In ETX, ETX metric allows RPL to find a minimum-ETX path from the nodes to a root in the DAG instance. This is the minimum ETX path between a node and the DAG root is the path (among other paths between the source and the destination) that requires the least number of packet transmissions per packet delivery to the DAG root. Thus, minimum-ETX paths are generally also the most energy-efficient paths in the network.

The ETX uses the ETX metric to find the path to be used to deliver packets in a DAG instance with the minimum number of transmission required by using the ETX link metric to compute an ETX path metric based on the ETX link metric of each hop and choosing paths with smallest path ETX.

At first, the root node set the parameters to identify the smallest ETX path for each node:

- min_path_etx: A variable that determines the ETX path metric of the path from a node through its preferred parent to the root computed at the last parent selection.
- $MIN\_ETX\_PATH\_CONST$: A constant that defines the maximum ETX value that can be considered for a node to be considered for parent selection.

Each other node in the DAG (non root) computes the ETX path metric for a path to the root through each candidate neighbour by using the two parameters explained below:

- $ETX\_Neighbor\_Metric$: A variable that identifies the ETX metric for the link to a candidate neighbour.
- MIN_PATH_ETX: A variable that assigns a value for each neighbour and the minimum ETX path advertised by that neighbour.

A node computes the ETX path metric for the path by comparing all the MIN_PATH_ETX received for each candidate neighbour. If a neighbour ETX metric cannot be computed, it is set to infinity to avoid selecting it and potentially having high ETX paths.

A node SHOULD compute the ETX Path metric for the path through each candidate neighbour reachable through all interfaces. If a node cannot compute the ETX path metric for the path through a candidate neighbour, the node MUST NOT make that candidate neighbor its preferred parent.

If the ETX metric of the link to a neighbour is not available, the ETX Path metric for the path through that neighbour SHOULD be set to INFINITY. This metric value will prevent this path from being considered for path selection, hence avoiding potentially high ETX paths.

The ETX Path metric corresponding to a neighbour MUST be re-computed each time the ETX metric of the link to the candidate neighbour is updated or if the a node receives a new min_path_ETX advertisement from the candidate neighbour.

After computing the ETX path metric for all candidate neighbours reachable for the current DAG instance, a node selects the preferred parent. The selection process is based on the condition that the ETX path metric corresponding to that neighbour is smaller than the ETX path metric of all the other neighbours.

Once the preferred parent is selected, the node sets its min_path_ETX variable to ETX path metric of the preferred parent. The vale of this variable is then carried in the metric container whenever DIO messages are sent.

5.3.4 Load Balancing Objective Function

Load Balancing Objective Function LBOF adds Child Node Count (CNC) as a metric, and uses it to select paths in a way that maintains a balanced number of children per preferred parent in the DODAG [60]. This will balance the traffic between the nodes, resulting in lower power consumption (hence longer network lifetime), a lower possibility of bottlenecks, and better delivery rate. An evaluation for this OF was carried in [61] with a comparison to OF0 and MRHOF, and it shows that LBOF provides longer network lifetime (by 16–40%) and better delivery rate (by 10–15%). However, with larger networks the LBOF seems to consume more energy due to parents churn. For this reason LBOF is considered out of context of this research.

5.3.5 Traffic Aware Objective Function

Traffic Aware Objective Function (TAOF) uses a combination of EXT and Packet Transmission Rate (PTR) as routing metrics, and uses it to select paths with less traffic towards the root and is defined in [62]. Authors in [63] defines TAOF which balances the traffic load that each node processes in order to ensure node lifetime maximization. They alter the DIO message format, introduced a new RPL metric, named Traffic Rate and used a new parent selection algorithm. The results in [63] show that TAOF achieves enhanced performance in terms of Packet Delivery Ratio (PDR) and that it builds more stable networks with fewer parent changes. However, it doesn't cope well with a dynamic network as it will increase the packet delivery ratio if the number of hops to reach the border gateway increases. For this reason TAOF is considered out of context of this research.

6 Security

Security is a major issue in the roadmap as explained in [64] to implementing the Internet of things mainly because it is not possible to directly apply existing Internet-centric security mechanisms due to the intrinsic features of WSN (e.g. the capabilities of the nodes, the bandwidth of the wireless channel).

The purpose of those readings was to understand the standards and protocols that are becoming the driving force for securing a large network of sensors and small

devices that will form the Internet of Things. This security involves securing the key establishment process and the routing discovery and establishment process.

Like any other network, the primary goals of securing the Wireless Sensor Network are the standard security goals such as confidentiality, integrity, authentication and availability.

- Confidentiality: the ability for a message to remain confidential but concealing it from a passive attacker. For WSN, a sensor node should not reveal its data to its neighbours.
- Authentication: the ability to ensure that the message reliable by confirming and identifying the source of this message (origin). Data authentication can be achieved by verifying the identity of source through symmetric or asymmetric mechanisms.
- Integrity: the ability of nodes to ensure that the message was not tampered and modified during transmission.
- Availability: the ability to use the resources and retain them for the whole duration of the communication of messages.

Other security goals such as data freshness, self-organization and secure localization are also of importance. Data freshness is the ability to ensure that the message received is the most recent one and that no newer messages were relayed. Self-organization in a network is when a node is able to self-organize and self-heal itself when it was compromised. Secure localization is the ability to locate accurately a node in a network.

Security challenges for the IoT and its integration within the IoT is studied as the challenges are tightly applicable to other relevant technologies of the IoT such as embedded systems, mobile phones and RFID. Security Threats for IoT based on the goals mentioned above are:

- Confidentiality: threats for confidentiality in IoT involves an attacker eavesdropping and overhearing critical information such as sensing data and routing information. Based on this the adversary may cause severe damage since they can use the sensing data for many illegal purposes [7].
- Authentication: threats for authentication in IoT involves attacks on the network that can alter the packets. It can also inject false packets. Another threat for IoT, is a general threat for wireless networks. The nature of the media and the unattended nature of wireless sensor networks make it extremely challenging to ensure authentication.
- Integrity: a malicious node present in the network can inject false data. Instability of wireless channel can cause damage or loss of data.
- Achieving a self-organizing and self-healing network in IoT is considered challenging since there is no fixed infrastructure to manage the network. This inherent feature brings another challenge as the damage resulting from an attack can be devastating.
- Localization in Wireless sensor network is essential as a compromised node can result for the attacker to manipulate data sending wrong location information by reporting false signal strengths and replaying signal.

Wireless sensor network limitations/weaknesses:

- Limited resources: for wireless sensor networks, the nodes will be limited in terms of memory, energy and processing power. Any of the security functions that will be applied on a WSN will need to take into consideration those issues as most of the available protocols and standards for encryption, decryption, data signatures, and signature verification consume memory, energy and computational power.
- Highly unreliable communication medium is another limitation for the wireless sensor networks as the nature of the communication medium can cause latency, multi-hop routing, network congestion or even conflicts such as collision. Unreliable transfers is another limitation where packets can become corrupted or even discarded which results in packet loss. This will force nodes to allocate more resources to error handling.
- On most wireless sensor networks applications, node will be left unattended and this can cause serious issues and limitation especially when nodes are exposed to physical attacks. The network is distributed thus if the design is not adequate, it can leave a network that is hard to manage, inefficient and fragile.

6.1 Security in RPL

Mayzaud et al. [65] identified three different categories of attacks on RPL that can violate one or more of the security goals defined in the previous section. The first category covers nodes resources such as energy, memory and processing power. The second category includes attacks on the topology of the RPL network and the third category corresponds to attacks against the network traffic. Attacks in the first category can damage the network since all nodes are constrained and this will shorten the lifetime of these nodes. Attacks in the second category will disrupt the normal operation of the network such as how RPL network converge and the third category of attacks will violate the confidentiality and integrity of data in the RPL network.

The main focus on this research is to mitigate attacks against traffic by preventing eavesdropping and passive sniffing.

RPL supports message confidentiality and integrity. It is designed as such that link-layer mechanisms can be used when available and appropriate and yet in their absence, RPL can use its own mechanisms. RPL supports three security modes defined in [53].

They are Unsecured, Pre-installed and Authenticated. Unsecured refers to the security mechanism that is provided in lower layers such as link layer security. Pre-installed and authenticated modes require the use of pre-installed shared keys on all motes prior to deploying the motes. Both modes provide security procedures and mechanisms at the conceptual level and are concerned with authentication, access control, data confidentiality, data integrity and non repudiation. This study focuses on the Pre-installed mode as a method of securing message transmission between motes in an RPL DAG instance. Authentication in the pre-installed mode involves

the mutual authentication of the routing peers prior to exchanging route information (i.e., peer authentication) as well as ensuring that the source of the route data is from the peer (i.e., data origin authentication) [66]. The limitation of the pre-installed mode in its common form, is that it is assumed that a mote wishing to join a secured network is pre-configured with a shared key for communicating with all neighbours and the RPL root. This means that once this shared key is compromised, all network leaves in the RPL DODAG are compromised.

The process of distributing the keys is out of scope for the specification of the RPL request for comment document [53]. The document further assumes that in authenticated mode, the router will dynamically install new keys once they have joined a network as a host however how the router will distribute those keys is out of context for RPL specifications and is not defined.

The RPL control messages incorporated in [53] the secure field in the header contents as shown in Fig. 10. The secure field contains several subfields as shown in Fig. 11 and each of the subfields identify the level of security and the algorithms in use to protect RPL algorithms.

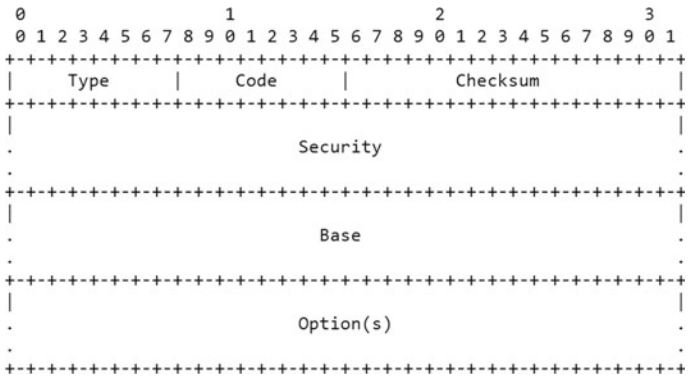


Fig. 10 Secure RPL control message as shown in [53]. The ICMPv6 information message with a type of 155. The code identifies the type of the RPL control messages (DIO, DAO, DIS, etc.), and the checksum computation field that is computed for each security message

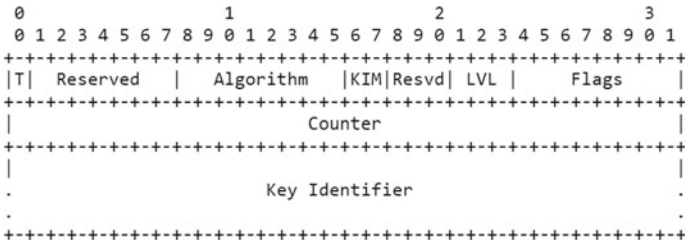


Fig. 11 Security section as shown in [53]. The level of security of the algorithm in use are indicated in the protocol message. The algorithm field specifies the encryption type, the MAC and signature scheme the network uses. The counter is time T that is a timestamp of security

The security variants provide integrity and replay protection as well as optional confidentiality and delay protection. The optional confidentiality variant is not defined in [53] however a security algorithm is proposed to specify the encryption algorithm to be used once keys are distributed.

The main security fields shown in Figs. 10 and 11 are the Message Authentication Codes (MAC) and signatures provide authentication over the entire unsecured ICMPv6 RPL control message, including the Security section with all fields defined but with the ICMPv6 checksum temporarily set to zero. Encryption algorithm provides confidentiality of the secured RPL ICMPv6 message that includes the cryptographic fields (MAC, signature, etc.). In other words, the security transformation itself (e.g., the Signature and/or Algorithm in use) will detail how to incorporate the cryptographic fields into the secured packet. The Security Algorithm field specifies the encryption, MAC and the signature scheme the network uses. The cryptographic mode of operation described in [53] (Algorithm = 0) is based on CCM and the block-cipher AES-128 defined in [67]. This mode of operation is widely supported by existing implementations.

6.2 Cryptography in IoT

The end-to-end principle argues that many functions can be implemented properly only on an end-to-end basis, such as ensuring the reliable delivery of data and the use of cryptography to provide confidentiality and message integrity. Adding a function to improve reliability on a particular link may provide some optimization, but can never ensure reliable delivery end-to-end. Similarly, security objectives that can only be met by protecting the conversation between two end-nodes are therefore best met by performing the cryptography at layer 3 or higher. There may even be security objectives that require protecting the data itself instead of the communication channel. However, this does not mean that all security objectives can be met end-to-end. In particular, achieving robust availability often requires protecting the subnetwork against attackers and more so for wireless networks. Adding a first line of defence at layer 2 may also increase robustness against attacks on confidentiality and integrity.

When combining encryption with authentication, some of the authenticated information may have to be sent in the clear. AES/CCM therefore encrypts a message (m) and authenticates that together with (possibly empty) additional authenticated data a, using a secret key K and a nonce N. A parameter L controls the number of bytes used for counting the AES blocks in the message; m must be shorter than 28L bytes. For IEEE 802.15.4 packets, the smallest value of L = 2 is plenty. Counter with CBC-MAC (Cipher Block Chaining Message Authentication Code) [CCM] is an authenticated encryption algorithm that provides at the same time confidentiality, authentication and integrity protection.

Even with the best link-layer security mechanisms, the data is no longer protected once it leaves the link. This makes the data vulnerable at any point that is responsible for forwarding it at the network layer, or on any link that has lesser security. Even

worse, an attack on the network layer might be able to divert data onto a path that contains additional forwarding nodes controlled by the attacker. End-to-end security that protects the conversation along the entire path between two communicating nodes is therefore an important element of any robust security system, so much so, that this requirement became a banner feature in the development of IPv6 [45].

Security involves two main aspects, the Network access (authorization) and the key management during the device communication. Key management protocols can be classified according to the method the key is delivered (key transport or key agreement) and whether key exchanged are based on symmetric or asymmetric cryptography.

Symmetric techniques demand the communicating parties to possess the same key prior to message exchange. Standard online key exchange protocols involving public parameters or trusted authorities are generally avoided. Instead, as defined in [68] Key Pre-Distribution KPS techniques, involving the following steps are preferred: (i) Preloading of Keys into the sensors prior to deployment; (ii) Key establishment: this phase consists of (a) Shared key discovery: establishing shared key(s) among the nodes and (b) Path key establishment: establishing path via other node(s) between a given pair of nodes that do not share any common key.

All of key management or key agreement schemes follow one of the three general key agreement schemes: trusted-server scheme, self-enforcing scheme, and key pre-distribution scheme. Trusted server scheme is not suitable for wireless sensor network as usually there is no centralized infrastructure in sensor networks such as a centralized entity to manage Kerberos. The self-enforcing scheme depends on symmetric cryptography such as a key agreement using a public key certificate. Limited computation and energy resources of sensor nodes often make it undesirable to use public key algorithms such as Diffie-Hellman key agreement or RAS. Many implementation and evaluation proved this to be an unrealistic scheme for WSN [69]. To use Public Key Infrastructure (PKI) technology. For example, each endpoint must be able to store digital keys, run encryption and decryption algorithms and conduct sophisticated handshakes to establish secure SSL connections, etc. However, many IoT nodes like the passive RFID tags or sensors simply don't have the electrical power, storage, or processing power necessary to tackle even the simplest of PKI tasks.

The time to execute the main cryptographic operation of ECC, the scalar point multiplication has been reduced from 34 s in 2004 to less than 0.5 s in 2009. With ECC, any node can make use of digital signature schemes (ECDSA), key exchange protocols (ECDH), and public key encryption schemes (ECIES). However, PKC is still too expensive to be used by sensor nodes implementing web servers as the overhead of its software implementation (420 ms) is too high. Note that the use of other PKI primitives with extremely efficient encryption and verification is discouraged. However PKI is still too expensive to be used by sensor nodes implementing web servers, as the overhead of its software implementation (420 ms) is still too high [70].

IPsec was considered a serious contender for securing WSN and many methods of research were involved in creating a lightweight version of IPsec to be incorporated into the 6LoWPAN architecture. Authors in [71, 72] suggested compressing the IPsec

and only looked at the authentication header part of the IPsec but suggested to use key pre distribution for the end to end communication. Other research suggested that the IPsec is unsuitable IAS t is designed for one-to-one communication. However, the dominant types of communication in WSNs are Many-to-one and One-to-many. This makes such protocols unsuitable for the usage in WSNs.

Sensors can use the 6LoWPAN protocol to interact with an IPv6 network as they are powerful enough to implement symmetric key cryptography standards such as AES-128 in [73].

Authors in [70] explain that even if assumptions were made that a WSN peer is protected by its own security mechanisms such as using the link layer security of IEEE 802.15.4, the public nature of the internet will require the existence of a secure communication protocol for protecting the communication between two peers. Key establishment is a fundamental security issue in wireless sensor networks (WSN). It is the basis to establish secure communication using cryptographic technologies between sensor nodes. Due to the current resource constraints on sensors, it is infeasible to use traditional key management techniques such as public key cryptography or key distribution centre based protocols. Therefore the key pre-distribution schemes are paid most attention in key management of WSN.

It was very important to understand how those networks utilize the available pre distribution techniques such as the mostly used one, proposed by Eschenauer and Giglor [1] to secure the Distributed Sensor Networks (DSN).

Authors in [69] modified E-G scheme by only increasing the number of keys that two random nodes share from at least 1 to at least q. It increased vulnerability in a large scale node compromise attack. They further extended this idea and developed two key pre-distribution techniques: a q-composite key pre-distribution scheme and a random pairwise keys scheme. The q-composite key pre-distribution also uses a key pool but requires two nodes compute a pairwise key from at least q-pre-distributed keys that they share. The random pairwise keys scheme randomly picks pairs of sensor nodes and assigns each pair a unique random key.

A framework was developed in [74] to be used to improve the performance of any existing key pre-distribution scheme using polynomial pairwise key. This framework does not require any prior knowledge of sensors' expected locations, and thus greatly simplifies the deployment of sensor networks.

It is now accepted to assume that the Key management scheme for distributed sensor networks developed by Eschenauer and Giglor is a standard to use for securing wireless sensor networks. However Eschenauer and Giglor only looked at the key pre-distribution schemes proposed for WSN and ZigBee as the main purpose of their research, our objective is to implement a Key distribution mechanism for the IoT to solve the problem of exchanging key between devices connected to the IoT without compromising the nodes or the validity of the Keys because of a Man in the Middle attack using the same scheme proposed by Eschenauer and Giglor [1]. Algorithm for the key management scheme for distributed sensor networks and how it will be used in the context of the IoT will be shown later on in this report.

6.3 Key Pre-distribution in DSN

In order to provide security between nodes communicating, encryption/decryption keys needs to be used for each and every communication link between devices. The main feature of key pre distribution and how it works is referred in the context of any Ad Hoc network as a challenge. The challenge simply lies in how the keys will be distributed beforehand and how to ensure that nodes communicating in an Ad-Hoc nature share a key and thus can provide secrecy and authentication by encrypting their communication channel.

The management of key is one of the key challenges to secure networks. We list below key pre distribution challenges when used in the context of the distributed sensor networks DSN.

- It is difficult to distribute keys and keying materials such as identifiers prior to deployment.
- Nodes in the networks are not authenticated and therefore obtaining a key does not guarantee that a node is trusted.
- Nodes in the distributed sensor networks are mostly battery operated low power devices, limited memory resources and computation power and the key pre distribution scheme chosen needs to have low overhead to ensure that the nodes can still operate efficiently.
- The nature of the distributed sensor networks and where nodes are located means that it is difficult to know where nodes. This can potentially result in the physical capture of the nodes and they become compromised and all credentials can be exposed.
- Note all nodes are implemented at the same time, for this reason the key pre distribution scheme needs to ensure that existing nodes in the network will work together securely with the newly added nodes.
- If node is compromised.

In addition the challenges to the key pre distribution presented above, the Internet of Things network present on top of those challenges other challenges unique to them. The main challenge related to this research is the nature of how nodes communicate in an IoT network which prevent nodes from creating more than one node and therefore if the key distribution scheme used does not produce enough keys not all nodes will participate in the IoT network.

In sensor networks, key distribution is usually combined with initial communication establishment to bootstrap a secure communication infrastructure from a collection of deployed sensor nodes. In the setting we study in this chapter, nodes have been pre-initialized with some secret information before deployment, but only after network setup will we know the location of nodes. The node location often determines which nodes need to establish a link with which other nodes, so we cannot set up these keys before deployment. In this chapter, we refer to the combined problem of key distribution and secure communications establishment as the security bootstrapping problem, or simply the bootstrapping problem. A bootstrapping protocol must

not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely.

This is a challenging problem due to the many limitations of sensor network hardware and software. In this chapter, we discuss and evaluate several well-known methods of key distribution. Besides these, we present an in-depth study of random key pre distribution, a method that has recently attracted significant research attention and we have also worked on. However, the pairwise key establishment problem is still not solved. For the basic Probabilistic and the q-composite key pre-distribution schemes, as the number of compromised nodes increases, the fraction of affected pairwise keys increases quickly. As a result, a small number of compromised nodes may affect a large fraction of pairwise keys. While the random pairwise keys scheme doesn't suffer from the above security problem and given the memory constraint, the network size is strictly limited by the desired probability that two sensors share a pairwise key and the number of neighbour nodes that a sensor can communicate with.

The interest of this research is to look at the various methods of key distribution between various devices in the context of the IoT proposed and study their feasibility.

Pre-distribution of keys can follow one of three major approaches when used in the context of the IoT as explained in [75]. The probabilistic approach explained in Sect. 6.4, the deterministic approach explained in Sect. 6.5 or the hybrid approach that combines both as proposed in [76–79].

Paterson and Stinson mathematically investigated in [80] the metrics that should be used to assess the suitability of the various probabilistic and deterministic key pre distribution schemes and identified them as the network size, storage requirements, network connectivity and network resilience. When using those key pre-distributions schemes in the context of the IoT other metrics also needs to be evaluated as proposed in [81]. The metrics are scalability to identify if the scheme can support large networks, efficiency to evaluate how much storage and processing power the used scheme will use, storage complexity in term of the amount of memory required to store the security keys for large networks and processing complexity in order to computer the amount of processor cycles required to establish a key and communication complexity as in the number of messages exchanged during the key generation and distribution process. Resilience should also be considered in evaluating how resilient the network will be if a node is captured and keys need to be revoked. Finally the key connectivity metric will need to be evaluated as the number of keys will increase if the probability of two nodes to share a key is low and this will have a high impact on the other metrics.

6.4 Probabilistic Key Pre-distribution

Probabilistic schemes is where the secure link establishment is conditioned by the existence of shared pre-loaded keys and deterministic schemes which ensure total secure connectivity coverage. The idea behind the probabilistic scheme was proposed first by Eschenauer and Giglor [1]. A Random key pre-distribution (RKP) where each

node is pre-loaded with a key ring of m keys randomly selected from a large pool. After the deployment step, each node exchanges with each of its neighbours the key identifiers that it maintains in order to identify the common keys. If two neighbours share at least one key, they establish a secure link and compute their session secret key which is one of the common keys. Otherwise, they should determine secure paths composed by successive secure links.

Traditional key exchange and key distribution protocols based on infrastructure using trusted third parties are impractical for large scale distributed sensor networks. There is no key distribution at the moment implemented on DSN other than key pre distribution. However the key pre distribution offers two inadequate solutions: Single mission key solution is inadequate because if one sensor node was compromised, this would lead to the compromise of all the DSN since selective key revocation is impossible upon sensor capture detection.

The other solution is pair wise private sharing of keys avoids compromise of the whole DSN since it allows selective key revocation. However, it requires pre distribution and storage of n − 1 keys in each sensor. This will mean that each node will require a large amount of memory to store the keys if for example a DSN contains 1000 nodes. In total there will be $n(n-1)/2$ keys per DSN. It will also render the communication between the devices complex and resources draining.

Eschenauer's and Giglor's approach was to propose a single key pre distribution scheme that requires memory storage for only a few tens to a couple of hundred keys, and yet has similar security and superior operational properties when compared to those of the pair wise private key sharing scheme.

Their scheme relies on Probabilistic key sharing among the nodes of a random graph and uses a simple shared key discovery protocol for key distribution, revocation and node re-keying.

This aim of this chapter is to identify how the Probabilistic key pre-distribution scheme can be applied in the context of the Internet of Things networks to allow keys to be distributed among nodes in the network so that only RPL nodes that share a pair-wise key can join the RPL DODAG and how the scheme will perform.

6.5 Deterministic Key Pre-distribution

Deterministic schemes ensure that each node is able to establish a pair-wise key with each of its neighbours. To guarantee determinism, LEAP make use of a common transitory key that is pre-loaded into all nodes prior to deployment. The transitory key is used to generate session keys between neighbouring nodes before being removed.

The scheme suggested by [2] divides the solution into three phases. In the first phase, each node attempts to discover which nodes are within its neighbourhood and to verify their identities. For this, each node will commit to each identity discovered in its neighbourhood and perform the fingerprinted mutual authentication protocol FMAP protocol with each neighbour it is supposed to share a key with. The FMAP protocol assumes that each node that is pre-loaded with the fingerprint of every

other node. Each node that joins the network broadcast a simple HELLO message containing its fingerprint and its key list. Every node that receive this message can verify the fingerprint in order to confirm uniqueness. If a similar fingerprint exists, the node is not allowed to join. At the end of the first phase, each node will have a list of all its neighbours including identity and fingerprint and will have verified the identity with neighbours that it shares key with. At this stage, nodes have not decided whether to accept this identity or not. Each node will overhear all FMAP protocol messages in order to decide whether it accepts its identity or not. In Phase 1, each n_i has now established a path with all direct neighbours that it was able to identify their identity of the form $n_i \rightarrow n_j$.

In the second phase and since a node has already identified direct neighbours that it shares a key with, the next step is to identify if a path can be established further beyond neighbours by using them as hops—That is the neighbours that exist outside of n's neighbourhood in the form of $n_i \rightarrow n_j \rightarrow n_k$. Verifying a node that is not a direct neighbour is more difficult as FMAP protocol cannot be imitated on nodes that are not neighbours (Those nodes cannot respond to HELLO messages from neighbours of neighbours). For this n_i will have to rely on the trust issued by each of its direct neighbours to their corresponding neighbours. However it cannot assume that the process of identifying of its neighbours n_j assumption about the identity is correct. For this it applies a voting process in which if the majority of nodes that are direct neighbours identify n_k as their direct neighbours then it assumes that n_k is an honest node. Since n_k is trusted by the majority, it is now considered as a trusted device by n_i and thus a 2 hop path is established.

In Phase 1, each n_i learns paths of the form $n_i \rightarrow n_j$, and in Phase 2 each n_i learns paths of the form $n_i \rightarrow n_j \rightarrow n_k$. Just as nodes informed their neighbours of the results of Phase 1 so that the information could be utilized to construct 2-hop paths, each node broadcasts the results of Phase 2 so that nodes of their neighbourhood learn which 3-hop paths exist. More specifically, each n_j will broadcast all paths it has discovered of the form $n_j \rightarrow n_k \rightarrow n_l$. This way, in phase 3 each node increases its knowledge of the network by one hop by relying on the nodes that were verified during phase 1 and 3 of the protocol. In phase 3, n_i is not voting for the majority to decide whether to trust n_l and has to trust that n_j already has chosen n_l as it gained majority.

The aim of this chapter is to question how the Deterministic key pre-distribution scheme can be applied in the context of the Internet of Things networks to allow keys to be distributed among nodes in the network so that only RPL nodes that share a pair-wise key can join the RPL DODAG and how it will perform.

6.6 Threats Attacks Trees

Internet of Things networks are subject to several threats as discussed in Sect. 2.2 and identified which threats can be mitigated by using encrypted communication between nodes in the network.

In this section we will look at the different threats that can be carried by malicious actors and the attack surfaces that can be exploited in order to compromise the network. We categorized the threats identified in Sect. 2.2 into two different type of attacks. The first category of attacks explained in Sect. 6.6.1 assumes that the malicious actor is exploiting the link of nodes that are sending data in plain text and on the encryption algorithm used to protect the link. The second category investigated in Sect. 6.6.2 shows how a malicious actor can attempt to exploit the routing formation or the routing table.

6.6.1 Attack Tree on Confidentiality and Integrity of Data

The threat of having an insecure communication between IoT devices is now more tangible than a conventional threat for any other type of networks on the Internet. Plain text communication makes it easier for attackers to tamper with data as well. In another scenario where an attacker tampers with the communication between an IoT device sending regular measurement of a valve in a factory for another machine to switch off for example at a critical level and modifies the temperature data. This can potentially be disastrous for a factory and might even lead to loss of life. We present in Fig. 12 the attack tree that results in violation of confidentiality, integrity or availability of the RPL DODAG.

Man in the Middle (MiTM) attacks will allow a malicious actor to eavesdrop into the communication and sniff the data transmitted between nodes. This will reveal both the data information exchanged between nodes and the control messages between nodes such as routing table formation. Since MiTM is most of the time used to allow further attacks such as session replay where the attacker stores messages exchanged between nodes in order to replay them later on. This will potentially lead to repudiation of data as there will be no method to identify and validate if the data sent is correct and the malicious actor can tamper with the data.

The proposed solution can protect some of the attacks presented in Fig. 12. Traffic analysis can partly be prevented as the traffic is encrypted and the payload (data) is not sent in the clear text. Having the data sent in clear text will violate the confidentiality of the data. This will also lead to violation of the integrity if further attacks are carried out such as Man in the middle attacks that can easily be done if the traffic is sent in clear text. The different cryptanalytic attacks presented depends on the encryption algorithm used.

6.6.2 Attack Tree on the Routing Formation and the Routing Table

In Fig. 13 we present how the RPL routing table or the RPL topology maintenance can be attacked. We note that they all rely on the presence of one or more malicious nodes in the network. A malicious node can disrupt the RPL DODAG formation and results in one of the attacks shown in Fig. 12, however, if the nodes communicate using the proposed solution and form secure links they can prevented.

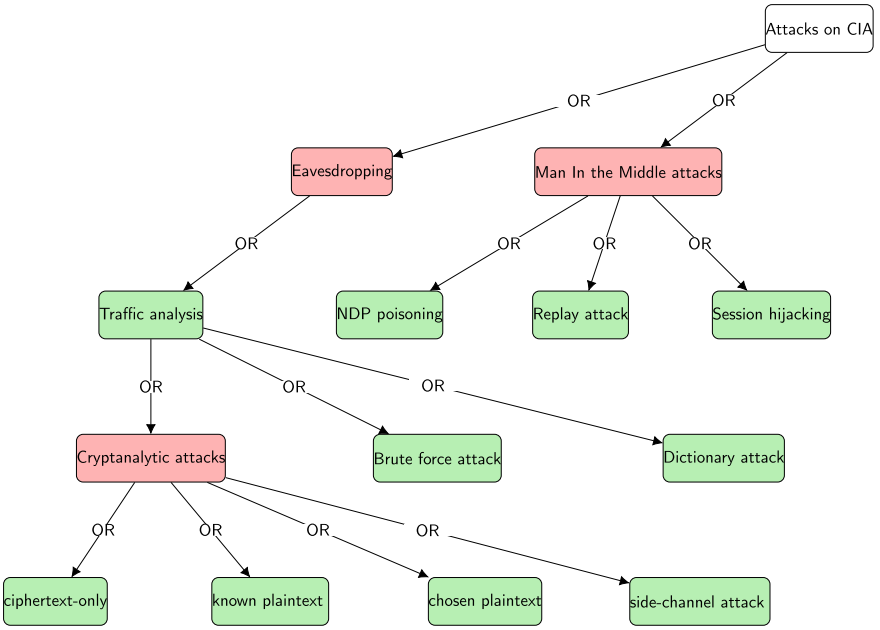


Fig. 12 Attack tree representation of all the attacks on the confidentiality, integrity and accountability that an Internet of Things network is vulnerable when all communications are sent in plain text

7 Summary

In this chapter, we defined the differences between the Wireless Sensor Networks WSN, the Distributed Sensor Networks DSN, and the Internet of Things IoT. The differences are mainly related to the link availability between nodes in the network since nodes between DSN and WSN are between each node and all its neighbours in comparison with the IoT networks where each node form a link only with one preferred neighbour based on certain variables.

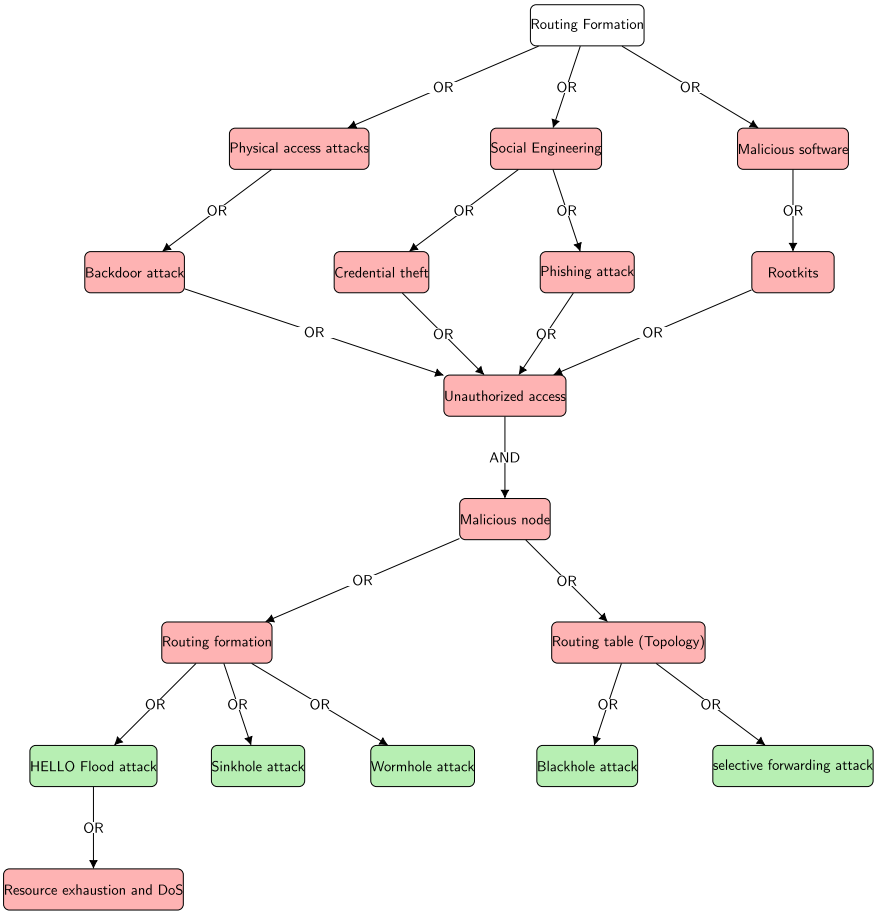


Fig. 13 Attack tree representation of all the attacks on the routing formation when all the routing control messages are sent in plain text

We introduced the IoT 6LoWPAN concept that defines how the Internet Protocol can be used in the context of the Internet of Things and researched the routing power for loss networks RPL and explained how it works and the various objective functions that can be used and the security measures that are incorporated within it.

We finally discussed the threats and vulnerabilities that IoT nodes and networks are vulnerable to and researched different key distribution schemes that are available and how each of them is used in the context of the IoT.

References

1. Eschenauer L, Gligor VD (2002) A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM CCS. ACM, New York, USA, pp 41–47
2. Henry KJ (2015) Secure protocols for key pre-distribution, network discovery, and aggregation in wireless sensor networks
3. Mulligan G (2010) The 6LoWPAN architecture, p 78
4. IEEE Computer Society (2011) 802.15.4 low rate wireless personal area networks (LR-WPANs)
5. Siller M, Carlos-Mancilla M, López-Mellado E (2016) Wireless sensor networks formation: approaches and techniques. J Sens 2016
6. Bellavista P, Cardone G, Corradi A, Foschini L (2013) Convergence of MANET and WSN in IoT urban scenarios. IEEE Sens J 13(10):3558–3567
7. Joby PP, Sengottuvelan P (2015) A survey on threats and security schemes in wireless sensor networks
8. Makhdoom I, Abolhasan M, Lipman J, Liu RP, Ni W (2019) Anatomy of threats to the internet of things. IEEE Commun Surv Tutor 21(2):1636–1675
9. Grammatikis PIR, Sarigiannidis PG, Moscholios ID (2019) Securing the internet of things: challenges, threats and solutions. Internet Things 5:41–70
10. Borgohain T, Kumar U, Sanyal S (2015) Survey of security and privacy issues of internet of things
11. Poudel S (2016) Internet of things: underlying technologies, interoperability, and threats to privacy and security. Berkeley Technol Law J 31(2):997–1022
12. Drăgoi V, Richmond T, Bucerzan D, Legay A (2018) Survey on cryptanalysis of code-based cryptography: from theoretical to physical attacks. In: 2018 7th international conference on computers communications and control (ICCCC), pp 215–223
13. Surendran S, Nassef A, Beheshti BD (2018) A survey of cryptographic algorithms for IoT devices. In: 2018 IEEE long island systems, applications and technology conference (LISAT), pp 1–8
14. Abomhara M, Køien GM (2014) Security and privacy in the internet of things: current status and open issues. In: 2014 international conference on privacy and security in mobile systems (PRISMS), pp 1–8
15. Chen X, Makki K, Yen K, Pissinou N (2009) Sensor network security: a survey. IEEE Commun Surv Tutor 11(2):52–73
16. Bysani LK, Turuk AK (2011) A survey on selective forwarding attack in wireless sensor networks. In: 2011 international conference on devices and communications (ICDeCom), pp 1–5
17. Choudhary S, Kesswani N (2018) Detection and prevention of routing attacks in internet of things. In: 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), pp 1537–1540
18. Raoof A, Matrawy A, Lung C (2019) Secure routing in IoT: Evaluation of RPL's secure mode under attacks. In: 2019 IEEE global communications conference (GLOBECOM), pp 1–6
19. Yang W, Wang Y, Lai Z, Wan Y, Cheng Z (2018) Security vulnerabilities and countermeasures in the RPL-based internet of things. In: 2018 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC), pp 49–495
20. Wallgren L, Raza S, Voigt T (2013) Routing attacks and countermeasures in the RPL-based internet of things. Int J Distrib Sens Netw 9(8):794326
21. Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in RPL-based internet of things. Int J Netw Secur 18(3):459–473
22. Le A, Loo J, Lasebae A, Vinel A, Chen Y, Chai M (2013) The impact of rank attack on network topology of routing protocol for low-power and lossy networks. IEEE Sens J 13(10):3685–3692

23. Rehman A, Khan MM, Lodhi MA, Hussain FB (2016) Rank attack using objective function in RPL for low power and lossy networks. In: 2016 international conference on industrial informatics and computer systems (CIICS), pp 1–5
24. Mosenia A, Jha NK (2017) A comprehensive study of security of internet-of-things. IEEE Trans Emerg Top Comput 5(4):586–602
25. Yang Y, Wu L, Yin G, Li L, Zhao H (2017) A survey on security and privacy issues in internet-of-things. IEEE Internet Things J 4(5):1250–1258
26. Nagrath P, Gupta B (2011) Wormhole attacks in wireless adhoc networks and their counter measurements: a survey. In: 2011 3rd international conference on electronics computer technology, vol 6, pp 245–250
27. Perazzo P, Vallati C, Varano D, Anastasi G, Dini G (2018) Implementation of a wormhole attack against a RPL network: challenges and effects. In: 2018 14th annual conference on wireless on-demand network systems and services (WONS), pp 95–102
28. Granjal J, Monteiro E, Sá Silva J (2015) Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun Surv Tutor 17(3):1294–1312
29. Nguyen KT, Laurent M, Oualha N (2015) Survey on secure communication protocols for the internet of things. Ad Hoc Netw 32:17–31. Internet of things security and privacy: design methods and optimization
30. Ahmed N, Sadiq A, Farooq A, Akram R (2017) Securing the neighbour discovery protocol in IPv6 stateful address auto-configuration. In: 2017 IEEE trustcom/BigDataSE/ICESS, pp 96–103
31. Ahmed ASAMS, Hassan R, Othman NE (2017) IPv6 neighbor discovery protocol specifications, threats and countermeasures: a survey. IEEE Access 5:18187–18210
32. Sudhakar, Aggarwal RK (2017) A survey on comparative analysis of tools for the detection of ARP poisoning. In: 2017 2nd international conference on telecommunication and networks (TEL-NET), pp 1–6
33. Chen B, Ho DWC, Hu G, Yu L (2018) Secure fusion estimation for bandwidth constrained cyber-physical systems under replay attacks. IEEE Trans Cybern 48(6):1862–1876
34. Hoehn A, Zhang P (2016) Detection of replay attacks in cyber-physical systems. In: 2016 American control conference (ACC), pp 290–295
35. Hu Q, Hancke GP (2017) A session hijacking attack on physical layer key generation agreement. In: 2017 IEEE international conference on industrial technology (ICIT), pp 1418–1423
36. Lu Z, Chen F, Cheng G, Li S (2017) The best defense strategy against session hijacking using security game in SDN. In: 2017 IEEE 19th international conference on high performance computing and communications; IEEE 15th international conference on smart city; IEEE 3rd international conference on data science and systems (HPCC/SmartCity/DSS), pp 419–426
37. Celebucki D, Lin MA, Graham S (2018) A security evaluation of popular internet of things protocols for manufacturers. In: 2018 IEEE international conference on consumer electronics (ICCE), pp 1–6
38. John R, Cherian JP, Kizhakkethottam JJ (2015) A survey of techniques to prevent Sybil attacks. In: 2015 international conference on soft-computing and networks security (ICSNS), pp 1–6
39. Zhang K, Liang X, Lu R, Shen X (2014) Sybil attacks and their defenses in the internet of things. IEEE Internet Things J 1(5):372–383
40. Genkin D, Valenta L, Yarom Y (2017) May the fourth be with you: a microarchitectural side channel attack on several real-world applications of curve25519. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, CCS'17. Association for Computing Machinery, New York, NY, USA, pp 845–858
41. Aggarwal R, Lal Das M (2012) RFID security in the context of "internet of things", pp 51–56
42. Special issue on "security and identity architecture for the future internet" (2013) Comput Netw 57(10):2215–2217
43. Ahmadi P, Islam K, Maco T, Katam M (2018) A survey on internet of things security issues and applications. In: 2018 international conference on computational science and computational intelligence (CSCI), pp 925–934

44. Miraz MH, Ali M, Excell PS, Picking R (2015) A review on internet of things (IoT), internet of everything (IoE) and internet of nano things (IoNT). In: 2015 internet technologies and applications (ITA), pp 219–224
45. Shelby Z, Bormann C (2007) 6LoWPAN: the wireless embedded internet, 1st edn. Wiley
46. Honggang Z, Chen S, Leyu Z (2018) Design and implementation of lightweight 6LoWPAN gateway based on contiki. In: 2018 IEEE international conference on signal processing, communications and computing (ICSPCC), pp 1–5
47. Kamma PK, Palla CR, Nelakuditi UR, Yarrabothu RS (2016) Design and implementation of 6LoWPAN border router. In: 2016 thirteenth international conference on wireless and optical communications networks (WOCN), pp 1–5
48. Janicijević N, Lukić M, Mezei I (2011) Routing protocol for low-power and lossy wireless sensor networks. In: 2011 19th telecommunications forum (TELFOR) proceedings of papers, pp 234–237
49. Montenegro G, Kushalnagar N et al (2007) Transmission of IPv6 packets over IEEE 802.15.4 networks. RFC 4944, Sept 2007
50. Conta A, Deering S, Gupta M (2006) Internet control message protocol (ICMPv6) for the internet protocol version 6 (IPv6) specification. RFC 4443
51. Deering SE, Hinden RM (1998) Internet protocol, version 6 (IPv6) specification. RFC 2460, Dec 1998
52. Hui J, Thubert P (2011) Compression format for IPv6 datagrams over IEEE 802.15.4-based networks. RFC 6282, Sept 2011
53. Winter T, Thubert P et al (2012) RPL: IPv6 routing protocol for low-power and lossy networks. RFC 6550, Mar 2012
54. Thubert P (2012) Objective function zero for the routing protocol for low-power and lossy networks (RPL). RFC 6552, Mar 2012
55. Vasseur JP, Kim M et al (2012) Routing metrics used for path calculation in low-power and lossy networks. RFC 6551, Mar 2012
56. Kushalnagar N, Montenegro G, Schumacher C (2007) IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. RFC 4919, Aug 2007
57. Gnawali O, Levis P (2012) The minimum rank with hysteresis objective function. RFC 6719, Sept 2012
58. Hui JW et al (2008) IP is dead, long live IP for wireless sensor networks. In: Proceedings of the 6th ACM conference SenSys. ACM, New York, USA, pp 15–28
59. Gnawali O, Levis P (2010) The ETX objective function for RPL. RFC 6719, May 2010
60. Qasem M, Al-Dubai A, Romdhani I, Ghaleb B, Gharibi W (2017) Load balancing objective function in RPL. Draft IETF
61. Qasem M, Al-Dubai A, Romdhani I, Ghaleb B, Gharibi W (2016) A new efficient objective function for routing in internet of things paradigm. In: 2016 IEEE conference on standards for communications and networking (CSCN), pp 1–6
62. Papadopoulos G, Dujovne D, Montavont N, Koutsiamanis R (2018) Traffic-aware objective function. Draft IETF
63. Ji C, Koutsiamanis R, Montavont N, Chatzimisios P, Dujovne D, Papadopoulos GZ (2018) TAOF: traffic aware objective function for RPL-based networks. In: 2018 global information infrastructure and networking symposium (GIIS), pp 1–5
64. Roman R, Lopez J (2009) Integrating wireless sensor networks and the internet: a security analysis. Internet Res 19:246–259
65. Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in RPL-based internet of things. Int J Netw Secur 18(3):459–473
66. Tsao T, Alexander R, Dohler M, Daza V, Lozano A, Richardson M (2015) A security threat analysis for the routing protocol for low-power and lossy networks (RPLs). RFC 7416, Jan 2015
67. Housley R, Ferguson N, Whiting D (2003) Counter with CBC-MAC (CCM). RFC 3610, Sept 2003

68. Chan H, Perrig A, Song D (2004) Key distribution techniques for sensor networks. Springer US, Boston, MA, pp 277–303
69. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. In: 2003 symposium on security and privacy, 2003, pp 197–213
70. Roman R, Zhou J, Lopez J (2013) On the features and challenges of security and privacy in distributed internet of things. Comput Netw 57(10):2266–2279
71. Raza S, Duquennoy S, Höglund J, Roedig U, Voigt T (2014) Secure communication for the internet of things—a comparison of link-layer security and IPsec for 6LoWPAN. Secur Commun Netw 7(12):2654–2668
72. Varadarajan P, Crosby G (2014) Implementing IPsec in wireless sensor networks. In: 2014 6th international conference on new technologies, mobility and security (NTMS), pp 1–5
73. Healy M, Newe T, Lewis E (2008) Analysis of hardware encryption versus software encryption on wireless sensor network motes. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 3–14
74. Liu D, Ning P, Du W (2008) Group-based key pre-distribution in wireless sensor networks. ACM Trans Sens Netw (TOSN) 4(2):11–20
75. El Mouaatamid O, Lahmer M, Belkasmi M (2021) A review on key pre-distribution schemes based on combinatorial designs for internet of things security. Int J Eng Appl Phys 1(1):1–8
76. Camtepe SA, Yener B (2007) Combinatorial design of key distribution mechanisms for wireless sensor networks. IEEE/ACM Trans Netw 15(2):346–358
77. Huang Q, Cukier J, Kobayashi H, Liu B, Zhang J (2003) Fast authenticated key establishment protocols for self-organizing sensor networks. In: Proceedings of the 2nd ACM international conference on wireless sensor networks and applications, WSNA'03. Association for Computing Machinery, New York, NY, USA, pp 141–150
78. Lee J, Stinson DR (2005) Deterministic key predistribution schemes for distributed sensor networks. In: Handschuh H, Hasan MA (eds) Selected areas in cryptography. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 294–307
79. Liu D, Ning P (2004) Multilevel tesla: broadcast authentication for distributed sensor networks. ACM Trans Embed Comput Syst 3(4):800–836
80. Paterson MB, Stinson DR (2011) A unified approach to combinatorial key predistribution schemes for sensor networks. Cryptology ePrint archive, report 2011/076
81. Yener B, Camtepe SA (2005) Key distribution mechanisms for wireless sensor networks: a survey. Technical report TR-05-07